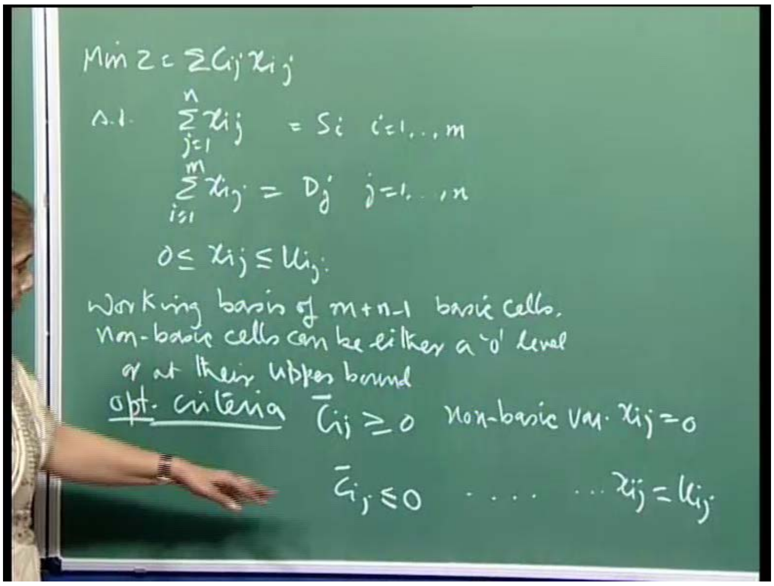
Linear Programming and its Extensions

Prof. Prabha Sharma
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture No. # 28

Bounded Variable Transportation Problem Min-Cost Flow Problem

(Refer Slide Time: 00:29)

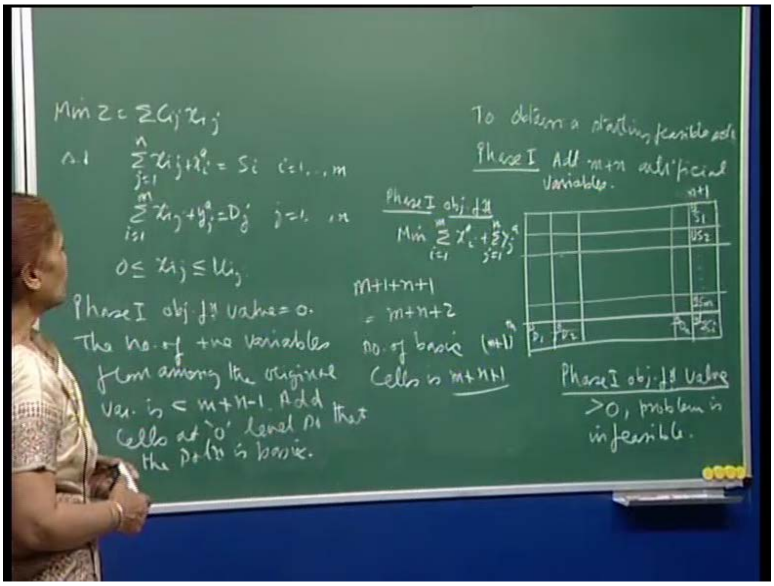


Let us continue our discussion of the bounded variable transportation problem. So, I told you that we can work with the working basis, so we spelled out that working basis of m plus n minus 1 basic cell and your non-basic cells here, can be your non-basic cells can be either at 0 levels or at their upper bound level or at their upper bound.

So, we will always maintain a basic solution of m plus n minus 1 cells; and if there are other non-basic variables, they will be their upper bounds, and I will put a square around non-basic cell which is at its upper bound, it distinguished from the other. Now, the optimality criteria again is same that we discussed for the simplex algorithm for the LPP; so, here the optimality criteria, optimality criteria would be C i j bar greater than or equal to 0 for non-basic variables x i j equal to 0 and it will be C i j bar less than or equal to 0 for non-basic variable x i j equal to u i j.

Because if it was possible for a non-basic variable at its upper bound, then by reducing the value of the variable from u i j to a lower value, I would have reduce the objective function value. So, therefore, the optimality criteria for a non-basic variable at its upper bound will be C i j bar less than or equal to 0, so all that holds as for the simplex algorithm.

(Refer Slide Time: 02:56)



Now, we said that how to get a starting feasible solution, because for the un-capacitated or the unbounded transportation problem, it was very easy to get a starting feasible solution; so, of course, one can use phase 1 to obtain a starting feasible solution; there are different methods; and later on I will give you a method which I consider is very neat, but we will have to wait for it, because we still have to discuss that class of problems; so that will help us to obtain the starting feasible solution. But here, of course, we can take recurs to phase 1; so, the phase 1 here would be simply that, you will add artificial variable here, and artificial variable here, that means, m plus n variables you would have added and so the transportation array, so you add m plus n artificial variables.

So, in phase 1 what will happen? So, your array would look like this, so you would have added…, so here this is…, and finally you will have a here m plus oneth column. So, let me…, so this artificial variables, so this will be your m plus oneth source and here you will have your m plus oneth market column row sorry; this will be n plus oneth row; so, this is a new this thing, and this will be n plus oneth market.

Because the constraints here, you will have added plus x a i - an artificial variable for each supply constraint; and here if you like you can distinguish and you can say that this is y j a equal to this. So, let us see, so for each a destination constraint or market constraint you have added artificial variable and this.

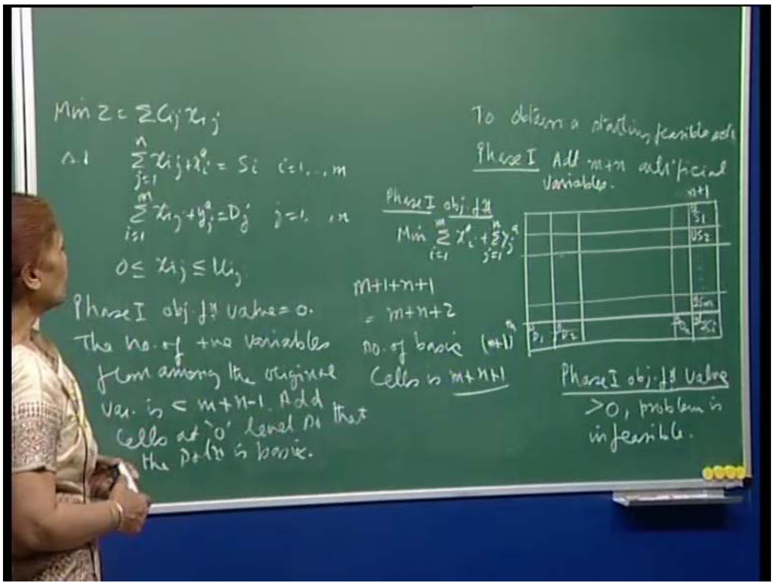
So, you see that this will be this, and here this will be this thing; so, then what would be your phase 1 objective function? So, phase 1 objective function would be phase 1 objective function, this will be minimize summation x a i i varying from 1 to m plus summation y j a j varying from 1 to n, so all other coefficients will be 0s, and these are 1(s); so, we will work out and the starting feasible solution you can immediately read off as x a i is S i and y j a is D j.

So, you will have this thing here; if you write this, so this will become S 1, S 2, and so on, this will be 0, the last one here would be S m, and then this would be D 1, D 2, so even read off your starting feasible solution to the extended problem like this, and here it will be D n.

So, these are the values; and this variable will have a 0 value; and these cells will have cost 0 except for the artificial variables; so, these cells will have cost 1, by the way…, and this will have a 0 cost; so, why would I write 0 here, it will actually the sum of this, so this will be summation S i, because this all adds up to S i, and but the cost will be 0 here; and so, this will also have cost 1, cost 1, and cost 1.

So, one can do the artificial variable thing here; you can apply phase 1, and once you end of the phase 1, that means, you have no non-negative C i j bar in that whole tableau, then you will stop with phase 1; so, if the as usual phase 1 objective function value is greater than 0, objective function I mean optimal, phase 1 we mean by this that at the end of phase 1 whatever the objective function value if it is greater than 0, problem is infeasible, and we will stop there.

(Refer Slide Time: 02:56)



But if the objective function value is 0, then there is a scope for that the problem is feasible; now, it is possible that, you may not have - so I can write out somewhere here - you may not have exactly m plus n minus 1, so if you want to keep the arithmetic. See here in this case, we had how many cells here, you had the number of rows, became m plus 1, number of columns became this, so total number became m plus n plus 2 an as usual; so, therefore, number of basic cells is m plus n plus 1 initially, so that is fine, because this is the extended problem; so, you we will have that many, so here you see can you can see, for example, this is n this is m plus 1, so m plus n plus basic cells you start with.

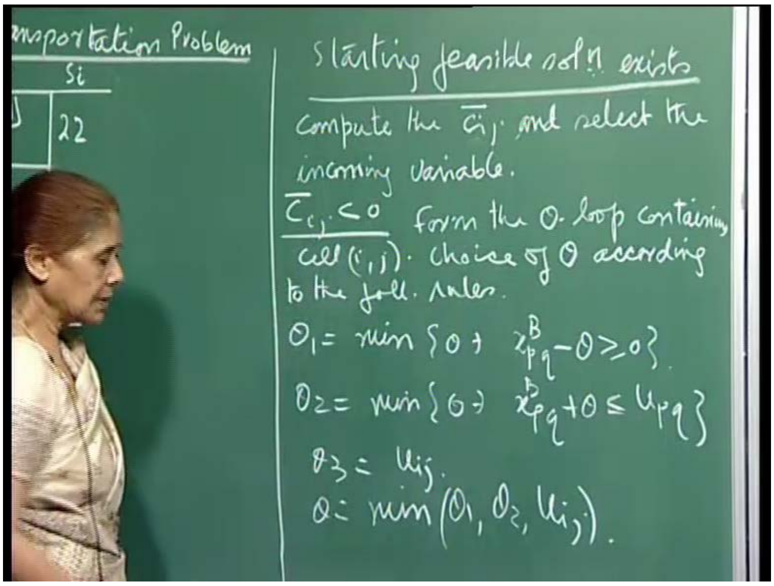
So, if the phase 1, so phase 1 objective function value is 0, then it is possible that the number of positive variables from among the original variables is less than m plus n minus 1; that means, you because there may be some artificial variables in the basis at 0 level, because the value of the objective function is 0, so artificial variables if they are in the basis (( )) 0 level.

So, we will drop those artificial variables, we will drop this row, and this row, and this column, so then add 0 cells or add cells at 0 level, so that the solution is basic; so, just make sure, and you know that the way you will add the 0 cells, you make sure that there is no theta loop formed; once you have that, then you have a starting basic feasible solution for your phase 2 and we have remove the row and the column and the prices you can work back with C i j(s); so, one can have a few simplifications here, for example, you would have to compute the C i j bars all over again, since you will have a new dual solution and this, so that can be done.

For the simplex algorithm, we were able to maintain the other set of C i j(s) also; and therefore, when we are phase 1 ended, we could immediately start with phase 2; but here we will have to do some more book keeping to be able to do that, but fine that is not the big problem, because any way you have so much simplification in the problem.

So, this can be done, but I would later on give you a better method for better in my opinion, its need not be better for everybody; so, I like to give you another method for obtaining starting feasible solution, so that is for the starting feasible solution.

(Refer Slide Time: 12:21)



Now, once you have again simply translation of all the rules that we worked out for the simplex algorithm, for a general LPP, so in this case here if you have a working basis how to proceed, so you have a starting basic feasible solution; suppose, so starting feasible solution exists, so then you will compute your C i j bars relative prices and select; see here there can be two opinions, because what is happening is that, you compute the C i j bar and select the incoming variable.

So, as Danzig says that, you go for the most negative one, if optimality criteria is not satisfied then C i; and of course, in this case, now you will have to look for see either most negative C i j bar or a positive C i j bar for a variable at its upper bound for a non-basic variable.

So, we will have to do this kind of sign or you can or you can just select the first variable which will improve the (( )) objective function, that means, you starts scanning, and the first variable that you first cell that you hit for which the corresponding variable at 0 level and the C i j bar is less than 0, you can enter it.

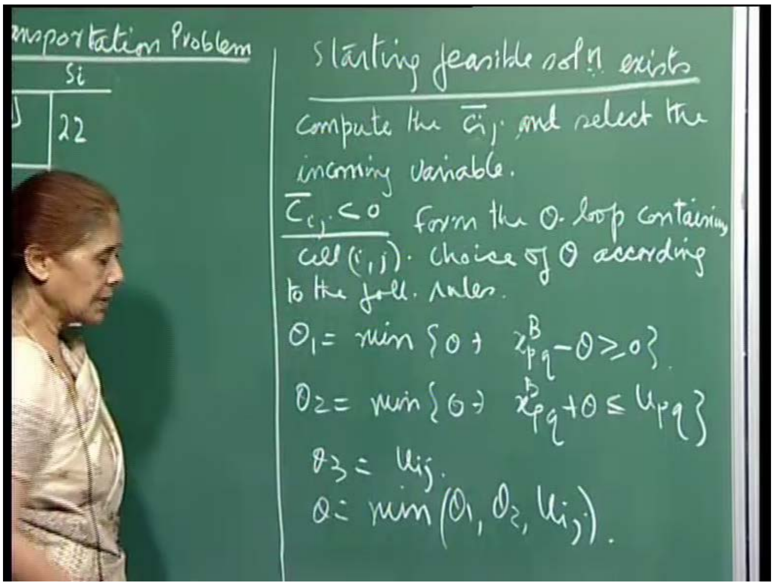
Now, it has been shown empirically again that, Danzig rule certainly gives you fewer iterations, because when you choose the most negative C i j bar or most positive C i j bar for a non-basic variable at its upper bound, the decrease of the objective function value is quite large, and so the number of iterations are fewer as compare to…, you are selecting the first cell which can improve the objective function value.

So, empirically it has been shown, but then the thing is that you see here, when you have a very large problems to select the most negative or the most positive, you really have to scan all the cells and find out which one is the most negative or most positive; so, compare to the amount of work that you have to do for iteration and the number of iterations; so, one can find out if the problems are very large, then may be the first negative thing would be or positive one.

Anyhow, first improving cell would be a better choice, because scanning will take lot of time, and the number of iterations amount of work require for iteration may be fewer or may be lesser; so, all those things considerations are there, and they all come by practice and so on, so we will select these, and then compute the C i j bar, and select the incoming variable.

So, here we have two cases; if your C i j bar is less than 0, and again we will have three considerations, so your theta loop form the theta loop containing cell i j; so, choice of theta according to the following rules, so theta 1 you choose as minimum of…, see in the theta loop, remember we will start with plus theta here, then alternate with minus plus in the theta loop I will be explain, so this will be minimum of all theta, such that, x i maybe I should write something else here, x p q B minus theta is greater than or equal to 0.

(Refer Slide Time: 12:21)



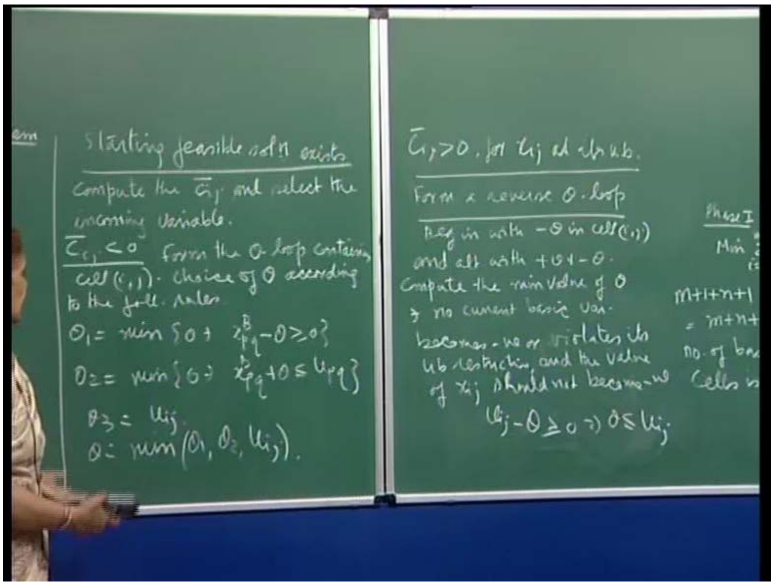
So, wherever the basic cell the theta is with the minus sign, then this will change by x p q minus theta, see you do not want to do this to become negative; so, for all theta, such that, this remains, so choose the smallest, so that none of the current basis basic variables becomes negative, that become 0; then theta 2 same I mean it is a nothing new, it just that I am repeating, so that you understand the concepts better, so theta 2 would be minimum of theta, such that, x p q B plus theta is less than u p q.

So, you understand for the basic cells; so, for the basic cells some of them will have plus theta, some will have minus theta in the theta loop, so the with the plus ones this value should not exceed their upper bound value, so theta accordingly, that means, here theta would be u p q minus x p q B, and here theta would be less than the current x p q value; and finally, theta 3 would be see, since i j is going to enter the basis, and it is going to become a positive number, because currently its value is 0, C i j bar less than 0, so theta 3 should be your u i j, that means, the incoming variable should not exceed its own upper bound.

So, all these considerations, we have already gone through well discussing the bounded variable simplex algorithm; so, then you choose your theta to be minimum of theta 1, theta 2, and u i j, and again I without spending time on writing down this thing; now, you know very well that, if the minimum occurs for if the minimum is theta 1, then that means, this particular variable for which the minimum occurs will become non-basic at 0 level, and your i j cell will become a basic cell; if this happens, theta 2, if theta 2 is the minimum of the three numbers, then the corresponding basic variable here will become non-basic at its upper bound.

So, we will put a square, we will remove the circle, put a square around it, and the incoming variable will be a basic variable; so, again you have a change in the basis; if the minimum is u i j, then that means, the status of the i j cell will not change, it will remain non-basic, but it will be at its upper bound, and the original basis remains, it just that you again now put a square around the i j cell to indicate that, it is a non-basic variable at its upper bound.

(Refer Slide Time: 19:59)

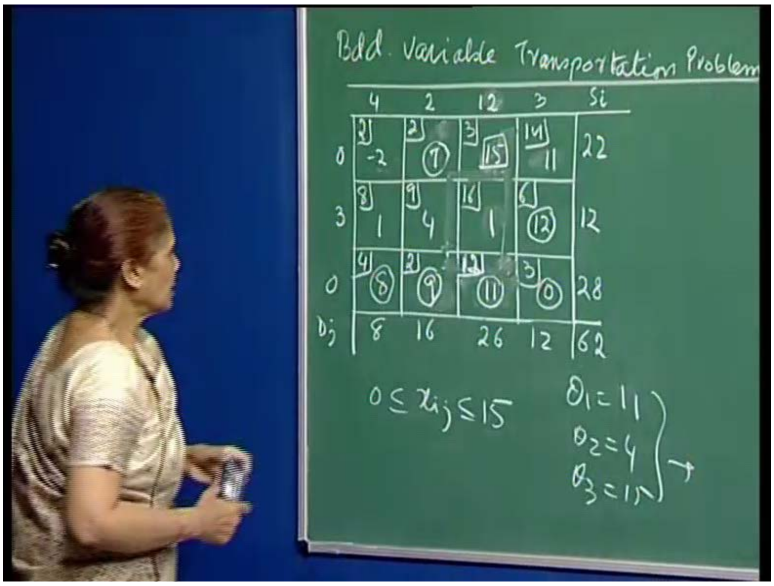


Similarly, you can argue out for C i j bar greater than 0; so, here we will say that, when C i j bar is greater than 0 for x i j at its upper bound; so, C i j bar is positive, so there is a scope, because when I remove where I reduce the value of x i j from its u i j value; so, this is positive, they will be decrease in the objective function value, and so there will be an improvement. Now, here what we will do is, you form a reverse theta loop, which means that, you begin with minus theta in a cell i j and alternate with plus theta n minus theta in that loop.

So, you form a reverse theta loop, that is all you have taken care of; and then again it is a same exactly so, and then compute the minimum value of theta such that no current basic variable becomes negative or violates its upper bound restriction, and the incoming variable and the outgoing variable; see for example, not I should not say outgoing, and the value of x i j should not become negative should not become negative, so the same values will be computed, they will be u i j, because this x i j value you are reducing, you are writing x i j minus theta, because you have beginning the loop with minus theta.

So, this should not become negative, that means, theta should be greater than sorry this should be this is u i j. Now, your currently your variable is at its upper bound, so this is u i j minus; therefore, theta should be should not become negative say this will remain this, so which implies that, theta is less than or equal to u i j; so, here the same considerations, we will have theta 1, theta 2, and theta 3, so how do you will choose the best one; and accordingly, you will decide whichever the minima occurs, you will decide whether the incoming variable is the incoming variable is going to be at 0 level or it is going to become non-basic or whatever.

(Refer Slide Time: 23:44)



Well, in this case, yes, so you can decide all that, fine, I think that makes the algorithm quite clear; we have already gone through the whole thing in the general case and it is just the simplifications which I thought I will point out. Now, let us go through this example, I thought I will show you the steps; so, here for example, you have bounded variable transportation problem each x i j has to be less than or equal to 15, these are the supplies, these are the demands, and I show you an initial starting feasible solution.

So, here for example, this is non-basic variable at its upper bound, otherwise you have a working basis, so 1, 2, 3, 4, 5, 6, your m plus n minus 1 is 6, so I have a basic feasible solution and one is thing. Now, we quickly compute the dual solution, so I will see put this as 0, so I can immediately get this value 4, 2, and this is 3, this is the non-basic variable. And now, you can come here, so this will be 0 again, because 2 plus 0 is 2, then this will be 3, and what else once you have this, then this should be 3, so this is your starting dual solution and lets compute this.

So, this is minus 2, 2 minus 4, this minus 2, so as I said one could have just started from here, but let us since this is small problem we will see, we can you follow that 6 rule, so this would be 3 and 11, no problem, 7, this is 1, then 5, this is 4, and here it is 6, so 10, and for this one remember this is the non-basic variable; so, here this is 12 minus 3, which is 9; so, you see it will improve, it will help to…, so we will choose this one as I said I could have selected this as an incoming variable or I can go for this one.

So, here you form the theta loop, and you see that, you immediately have this is as this, this, and this, but here it is going to be minus theta, plus theta, minus theta, plus theta; so, if you compute the three values, see here for example, theta 1 is 11, that is the only one; then these two values here, this should be…, so theta 2 is 4, because otherwise this variable will exceed its upper bound; and here your theta 3 is 15, because this should not become negative.

So, I will choose this one, theta 2 equal to 4, and so what will happen is that, this will become basic, but this will now become non-basic; so, let us quickly do it, and this number will become 7, so 4 this becomes 9; and now, this becomes a basic variable at level 11; and let me get rid of this loop here, yes these values also move away, this is 12, and this one becomes now a non-basic at its a upper bound; so, this is your new basic feasible solution, yes so we quickly compute; now, in this case, your really have it easy, because 4, 2, and this becomes 12, that is 3, so I simply have to compute, so this will remain the same, and yes and here this one will be your 3.

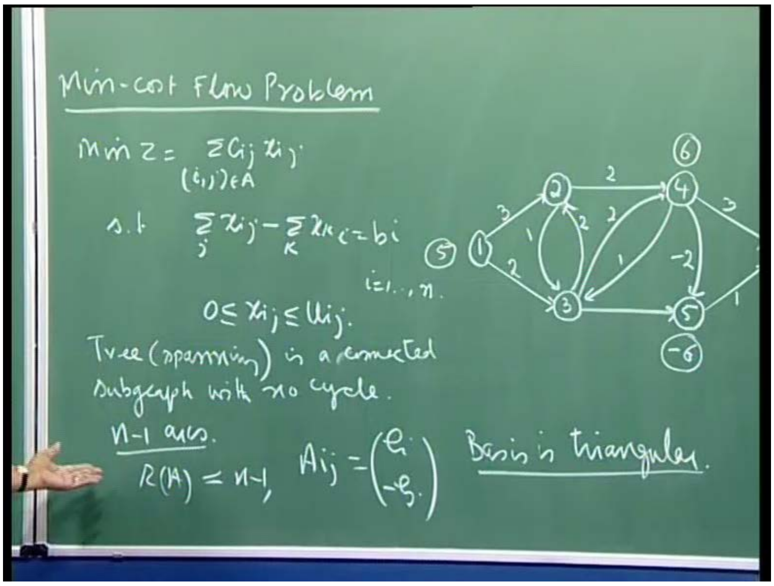
So, these numbers will not change; see it is only this one which will change, because these numbers and these numbers are the same, so I do not have to compute them; and here what will happen? If it is 15, this is still 1, when what will be this, this will number will also not change, because this is 11, yes I have to compute for this one, so this will be 3 minus 12, which is minus 9, so that is fine, and we are done yes.

So, just two iterations, because I choose a small problem would have like to show you another iteration, but does not matter, so it turn out to be a small problem and optimality conditions are satisfied, so this is your bounded transportation problem; so, once we have done this, I would like to now go to your general min-cost flow problem, which I had stated for you in the beginning; and then I said that the transportation problem with a special case, because the underlying graph becomes by part I for the transportation problem, so things become simpler; and now, let us look at…, I already showed you some what were the considerations.

So, there is not must change except that, see we will go through it, so that you become familiar with - you know - the graph terminology, because for the transportation problem we could translate everything to the array, so the graph was or not at all you know we were not referring back to the graph, those you should sometimes just draw the bipartite graph, and trace out the basic feasible solution on the bipartite graph we can do that.

So, why did not do it, because we want to cover some few more topics. Now, for the for the general min-cost flow problem where the underlying graph need not be bipartite, you will see that the, of course, the algorithm is the same, the simplex algorithm the simplifications are there; but then now we will be working with the graphs as the basic strictures instead of this array.

(Refer Slide Time: 30:12)



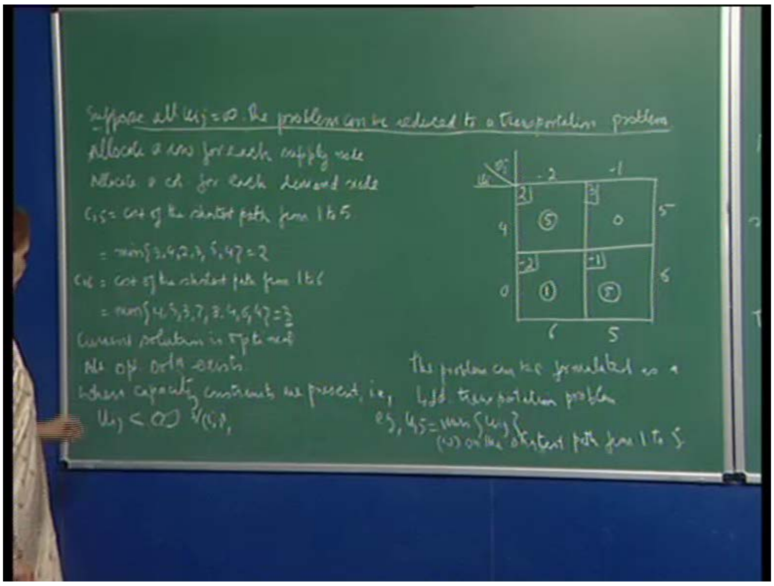
So, min-cost flow problem, we had defined it as minimize Z equal to summation C i j x i j where are arc i j belongs to A subject to summation x i j summation j minus summation x k i summation over k is equal to b i i varying from 1 to n; and of the general problem this will have x i j less than or equal to u i j, where again u i j may be finite or infinite, that means, the infinite case means the arc has no upper bound restriction.

So, I had talked to you that the basic solution, here would be a tree, I defined to you the tree is a…, or now sometimes I will use that tree or a spanning tree is the same thing, in the same thing I will be using them synonymously, but of course spanning tree is a connected…, I am just repeating this definition connected sub graph with no cycle.

So, we said that should have n minus 1 arcs; so, the spanning tree will have n minus 1 arcs, there will be no cycle present, I had defined all these things in the beginning when we were talking about it; so, here again, that means, you want to find, we can see that the simplex algorithm gets simplified here, because the structure and yes n minus arcs, and all this I had because the rank of the coefficient matrix is n minus 1; and we showed you.., again it is the same structure, because the transportation problem…, you are columns for e i e j, because I could get rid of the minus sign, there is here, if an arc i j appears here with the plus sign; if you see for the jth node, it will appear with a minus sign.

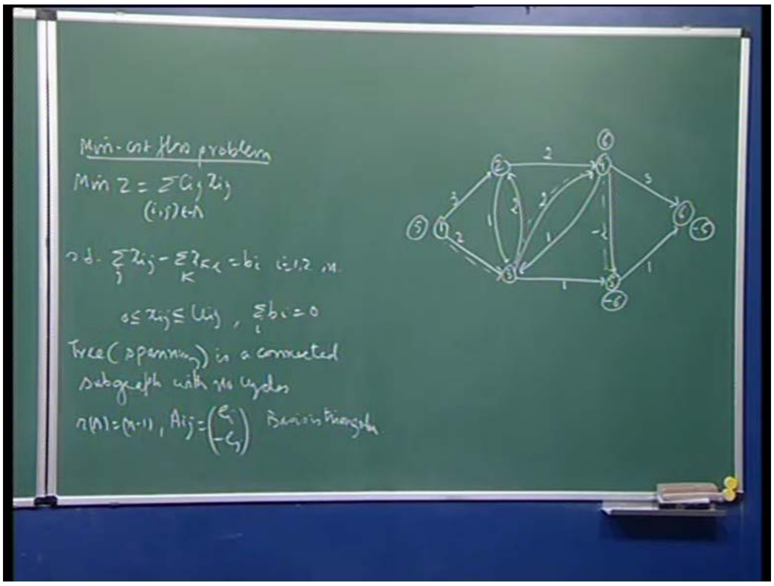
So, the column, because so you here you are A i j would be e i minus e j, so triangular basis and everything all those things hold here, I had shown to you all these things in detail in the for the transportation problem. So, we will say that the basis is triangular, all that things and that thing (( )), because you could do it, e i e j, you could show that the rank is n minus 1; so, here also you can immediately show that the rank is other matrix is n minus 1, and of course, I should have put the condition here that summation b i is 0.

(Refer Slide Time: 33:25)



So, this is important; we are always working with this condition. Now, you see for the un-capacitive case; suppose, all u i j's are infinity, that means, there is no restriction on the…, you can send as much amount as you wish, the problem can be reduced to a transportation problem; see they all related, and that is why…, and so you can take the advantage of the structure and so how will you do that? Yes.

(Refer Slide Time: 34:19)

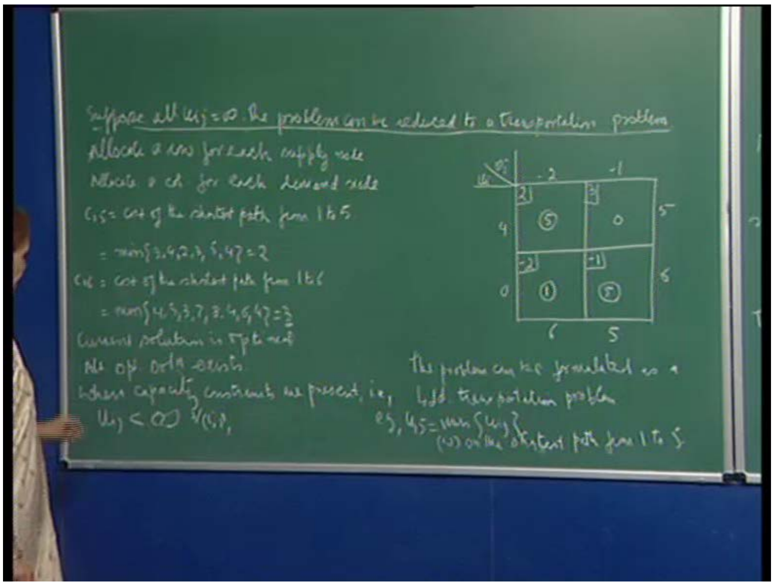


See what you can do is, in the just think here, it is possible that some b i(s) may be 0, and see it is a very general case, so some arcs, some nodes may not have any thing any supply or any demand. So, they would be…, and we call them transshipment points, also this term comes along, so for example, I have drawn this graph for you, so here there is amount available 5 units available at node 1 and at node 4, so there will be treated as our supply points. And here at node 5, at node 6, amounts are demanded, but 2 and 3 have no amount, no this thing available, or no amount to be demanded.

So, we will call these as transshipment points; so, I am now considering the case, when all u i j(s) are infinity, and I want to show you that this problem can be reduced to a transportation problem; in fact why I am saying reduce because the size of the transportation problem would be smaller, and what do we do here, we say that you allocate a row for each supply node, and allocate a column for each demand node.

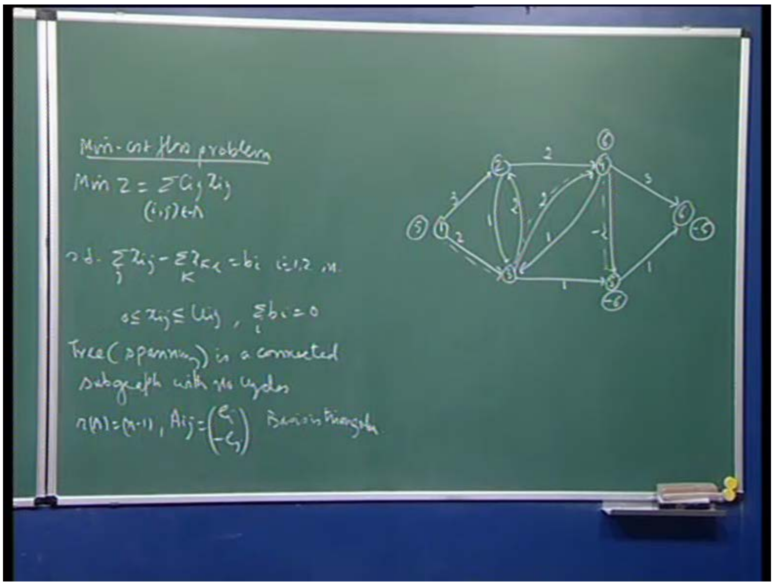
So, for example, you have a node 1 and 4, as we supply nodes, therefore 1 row for node 1, another row for node 4, and then nodes 5 and 6 are the demand nodes. So, therefore, column this column for this node 5 and this is the column for node 6.

(Refer Slide Time: 35:51)



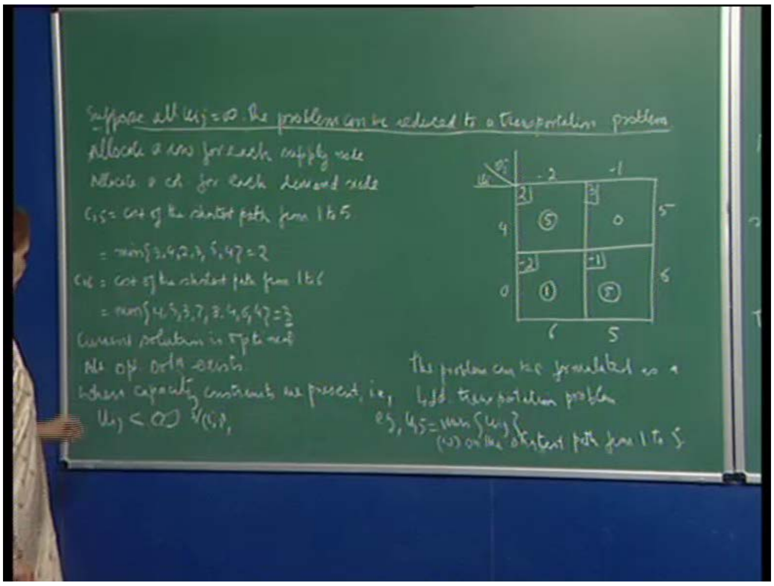
Now, how do we determine the costs; for example, I need to know the cost c 1 5, and so what we are saying is that, you will choose C 1 5 as the cost of the shortest path from 1 2 5; and this is again my idea is to show you how the size can really explode; now, for the small problem, you see the possible paths from 1 to 5 or 2 to 4 6 in number, and please for yourself sit down with that network and see which are the path which have these costs, so I think they should be no error here, and them I will choose the minimum one here.

(Refer Slide Time: 36:27)

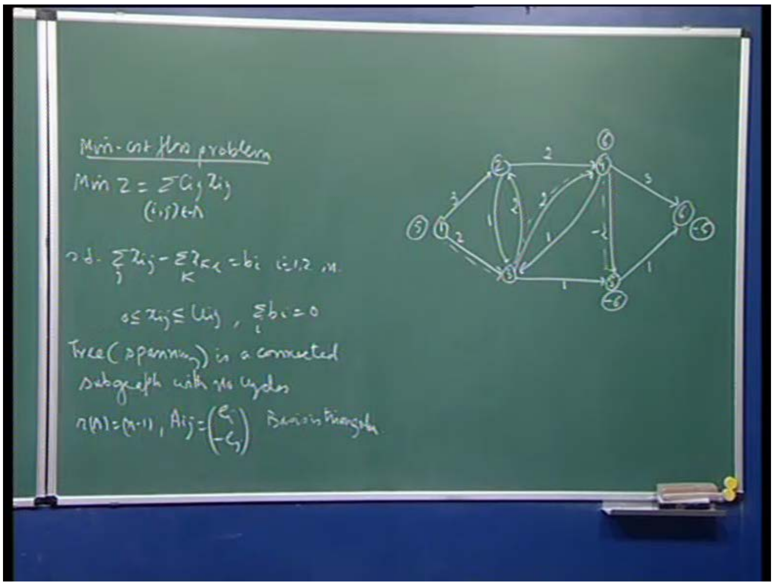


So, this is 2, and you will lead to of course somewhere write down which is that path, because later on you will have to…; so, for example, here I did not make a note, but you can quickly see that the path from 1 to 5 would include, so if you come here, this way 2 2 4 minus 2, so this 2, 1 to 3, 3 to 4, 4 to 5, so this 2 and minus 2 will cancel, and therefore the cost will be 2; so, this is the smallest path; so, from here you are going this way indicate the path here.

(Refer Slide Time: 36:57)



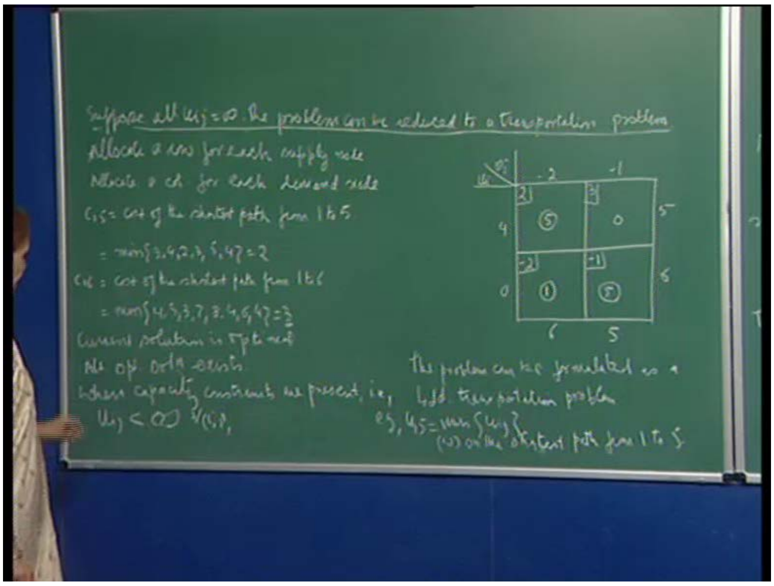
(Refer Slide Time: 37:14)



So, this will be the shortest path, and the cost there 4 will be 2, so I have put it down here. Then similarly, C 1 6 cost of the shortest path from 1 2 6, and here the number of paths is 2 4 6 8, so completes figure out reach are all these paths, I have written down the cost, so minimum cost is 3; and here also you need to figure out which is the smallest cost here, say for example, 3 to 5 and then minus 2.

So, this will be 3 and 4 no no sorry minus 2 to 3 3 that becomes 4, but my cost is 3, so there is the path will have to be short, may be if you do it this way 3 see I need to use that minus 2. So, if you come this way, then 3 plus 4, 4 plus 2, 6, again it is coming out to be minus 2, and then this will be 1, so that will be 5, so we will have to figure out which is there is path really, which gives you…, fine this will be 3 1 4 4 and 2 6, again this will be 6 minus 3 4 2 6, 6 minus 2 would be 4, 4 1 5, so that does not help you, 2 1 3.

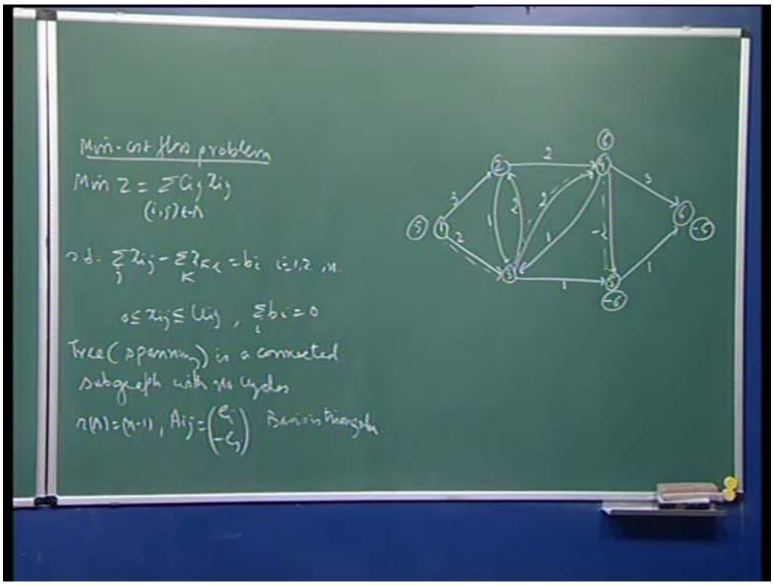
(Refer Slide Time: 38:31)



Now, fine, so then please figure out which is the shortest path, 3 1, because this will become 6, so I have somehow computed, maybe there is an error here, but just make sure that; in case, this is the minimum path, the cost is 3, or if it is not then you write the path, but I will show you calculations with this one; so, the current solution is optimal, I should have written now the path here; and now, you see once, you have these costs, let me use the north west corner rule and make the start with the basic feasible solution.

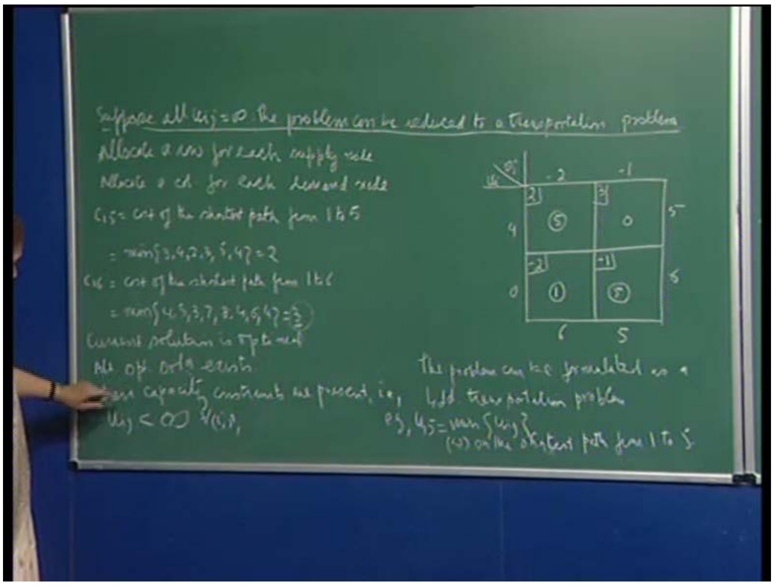
So, here I can allocate 5, because the minimum of 5 and 6 is 5, then I will come down here, allocate one more unit here, and then you go this way, so then this becomes 5 and 5, so you have this 5 here, so this is your cost.

(Refer Slide Time: 39:26)



And then you find out the dual solution, so obviously I will make this as 0, u 2 is 0, and then that gives me minus 2 minus 1 is that is obvious, because when you want to know the thing from 4 to 6, 4 to 5 is minus 2, and 4 to 6 would be 4 5 and 6, this will be the path.

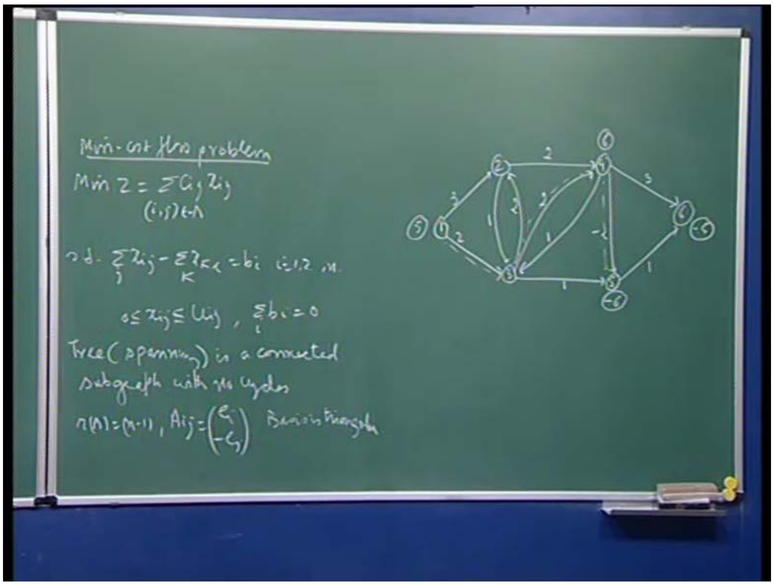
(Refer Slide Time: 39:40)



So, that will be minus 2 plus 1 minus 1; so, these are the shortest paths from the supply point 4 to the 2 demand points 5 and 6, so these are the 2 costs; therefore, by reduce this thing is there, reduce transportation problem is there; and now I can apply the transportation algorithm; so, as I said we will find out the northwest corner will apply the north west corner rule to get a starting solution, but it turns out that when you write down the dual variable, so this is 0, and therefore that will make these 2 as minus 2 minus 1, and then from here, you see that this will be 4, because 4 minus 2 would be 2, so 2 minus 2 would be 0.

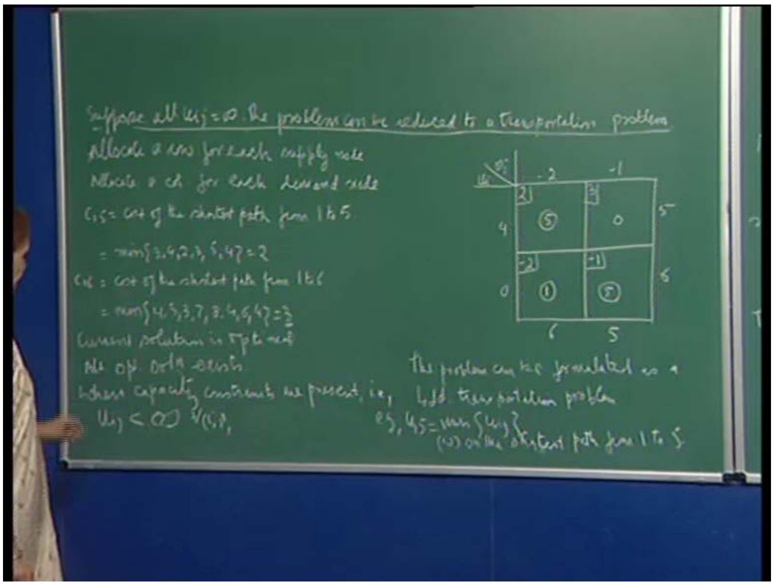
So, this is your dual solution, and then when you compute the cost of this cell, it is going to be 4 minus 1 is 3, so 3 minus 3 0. So, you have an alternate optimal solution; so, this is when my this cost is 3 if it is different, then you please make the calculations correct, so any way correctly there is an alternate optimal solution.

(Refer Slide Time: 40:59)



But before I go to the alternate optimum solution we can quickly interpret this solution now, so that is why you need to record the shortest paths somewhere, because if you have a larger network, then you simply show that 1 to 5; so, along the shortest path 5 units go this, this, and this as I told you 5 units, and then this is form 4 to 6, 4 to 6 the path was this one, so one unit along this path, and then you have 4 to 5, 4 to 5 you have 5 units along this.

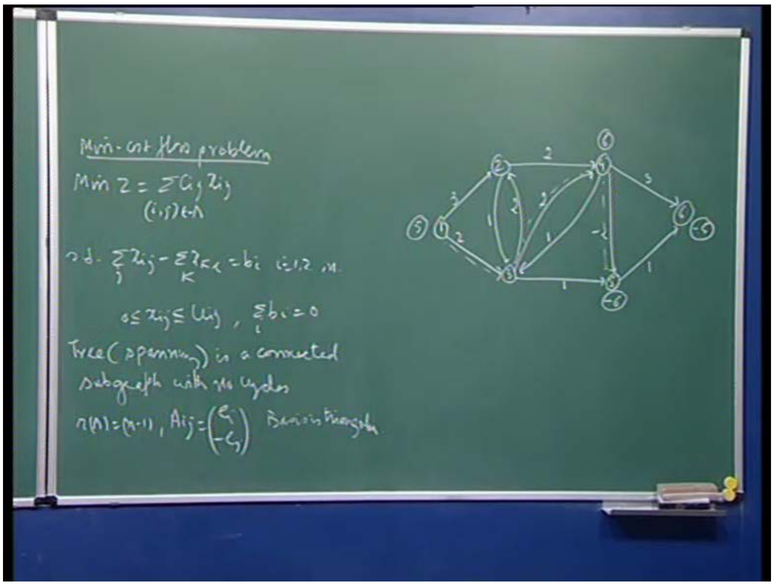
(Refer Slide Time: 41:21)



So, this are gets used the maximum number of times; anyway, so once you have the optimal solution to the transportation problem, you can go back and distribute the flow along the arc to get the actual flow on the network; and then the alternate optimal solution, you will have to form a loop like this, and then you can see that this is theta, this is minus theta, plus theta, minus theta, anywhere theta comes out be 5; and therefore, this will be 5, and this will be 6, and you will have a degenerate basic feasible solution, you can treat one of them as a basic variable, the other one you can drop out.

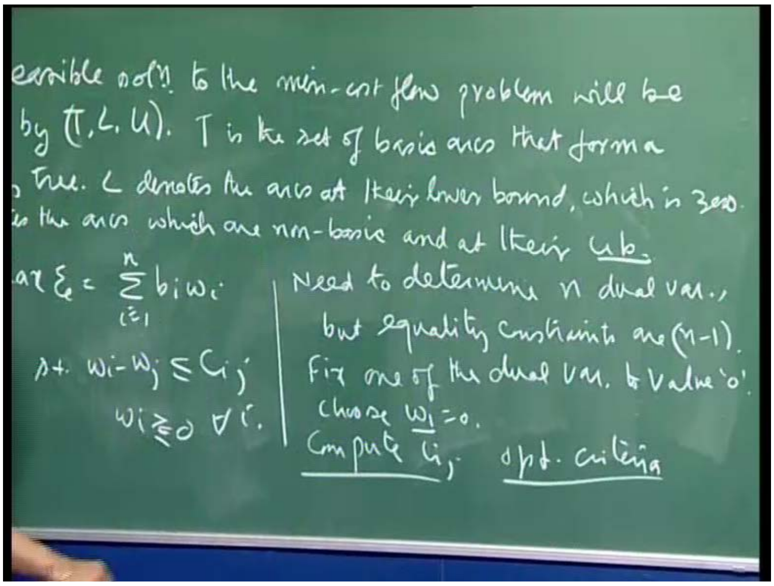
So, this was this was the idea behind; and if you have a capacities, if the u i j(s) were finite as I said here, if the u i j(s) are finite, then you the problem can be formulated as a bounded transportation problem and what will we do, for example, to compute u 15, you will do it as the u 15 will be the minimum of the…. So, the path you had from 1 to 5, whatever the path you will take the minimum capacity of all the arcs in the path and that will be your u 15, see you can immediately compute the capacities of the arc in the reduced transportation network.

(Refer Slide Time: 42:35)



And so, you will then handle the problem as a bounded transportation problem. Obtaining a starting feasible solution we will require some consideration, so that is only difficult point that gets added, otherwise the implementation of the simplex algorithm as you see is really quite simplified here. So, let us get back to the general solution here; so, how do I…, I will just tell out the steps of the algorithm, and then we will try we will work out a numerical example to again show you this steps of the algorithm.

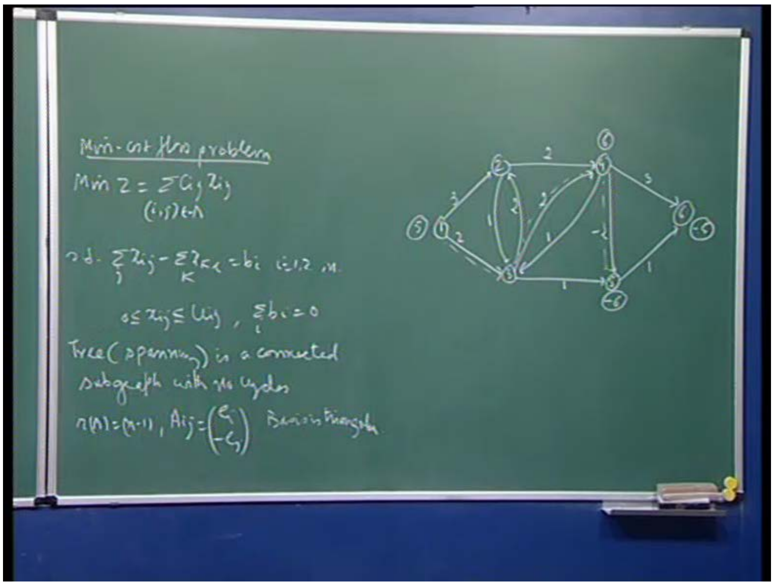
(Refer Slide Time: 43:12)



So, here so basic feasible solution to the min-cost flow problem will be denoted by T L u. And let me explain, so T is the set of basic arcs, the set of basic arcs that from a spanning tree that from a spanning tree; then L denotes the arcs at their lower bound, which is 0 here; see we could have the general situation where even your lower bounds are not necessarily 0s, but we will first discuss this case; and then once you understand this, than you can handle with a little modification, you can handle the case when the lower bounds are also different from the 0s, but which is 0, and u denotes the arcs which are non-basic and at their upper bounds.

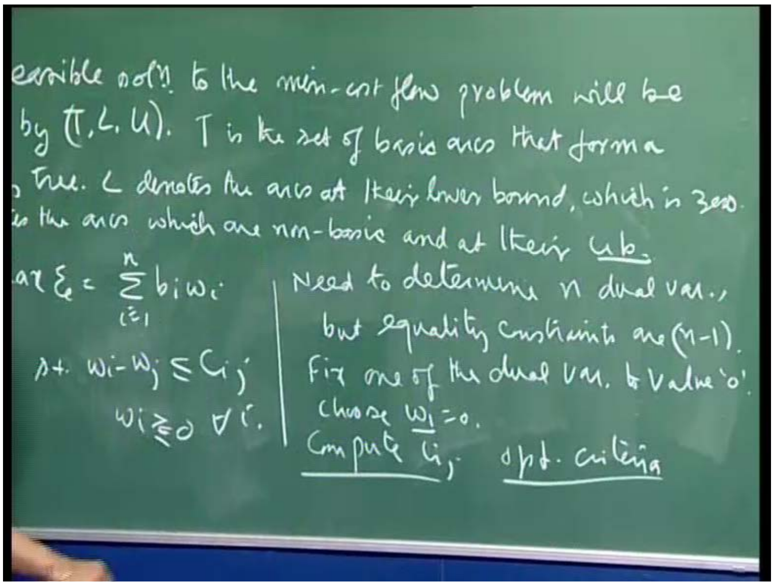
So, this is the kind of thing we will always maintain; and when you modify our current solution, then some things will change here, your tree will change, your arc will go out or come in or non-basic variable here becomes reaches its upper bound (( )), so all those things will happen; we have already seen this in the terms of transportation and bounded transportation problem and so on.

(Refer Slide Time: 46:10)



So, let me just give you now a feeling as to what they happens, see thing is that, and you will compute the fine, so let me write that dual, and then again not much different from what we have been doing so far, so here the dual same thing, because a column has e i and minus e j.

(Refer Slide Time: 46:30)

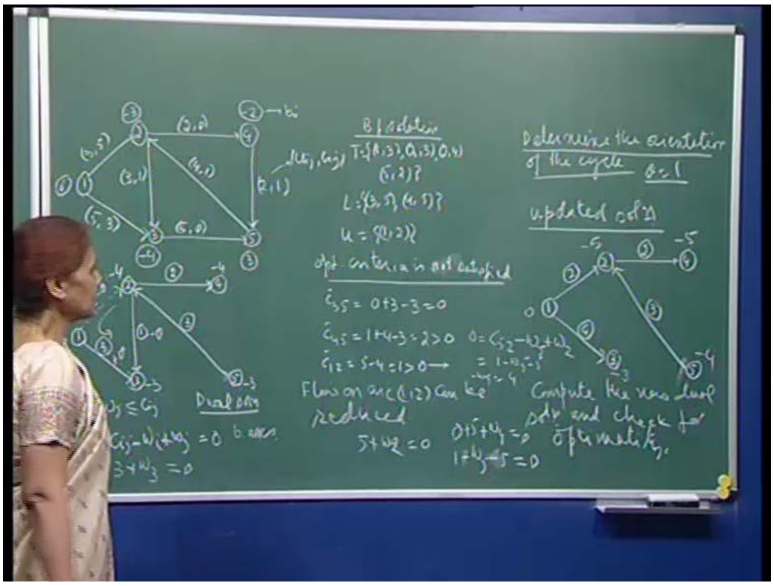


So, there will be u i and v j let us say; or in fact, here we are not having two different sets of dual variables, because we have node set is now divided into two subsets, so it will be the same set of this thing; so, let me call here a dual problem, we will define as maximize psi equal to summation this is b i w i i varying from 1 to n and nodes; so, you have this subject to…, so it will be w i minus w j less than or equal to C i j, and yes I am not explicitly considering these constraints, so then your w i greater less than 0 for all i for all nodes, so this is your dual problem; and same thing you need to determine n dual variables, but equality constraints are n minus 1, because corresponding to each basic variable the dual constraint must be satisfied as equality. So, we will have n minus 1; so fix one of the dual variables to value 0, and mostly so we choose w 1 to be 0.

So we just keep that convention; so, w 1 will always be 0, and again because of the triangular nature I had already spelled it out for you in the simplex algorithm for the transportation problem that you know; it does not really matter what value I gave to w 1 as long as I give it some constant value the relative prices will not…, because all the w i(s) will change by the same amount whatever the value of w 1, because of the triangular nature; you can see that or (( )) I showed the computations to you for the transportation problem.

So, therefore, the relative prices will not change, no matter what; so, for ease of for convenience we choose w 1 to be 0, so you will do that, and then will compute the…, and then so compute C i j bars; and here the same considerations, if C i j bar the optimality criteria would be compute this, and you know the optimality criteria C i j bar should be non-negative for arcs, which are at 0 level, and C i j bar should be less than 0 for arcs which are at their upper bounds non-basic cells.

(Refer Slide Time: 49:40)



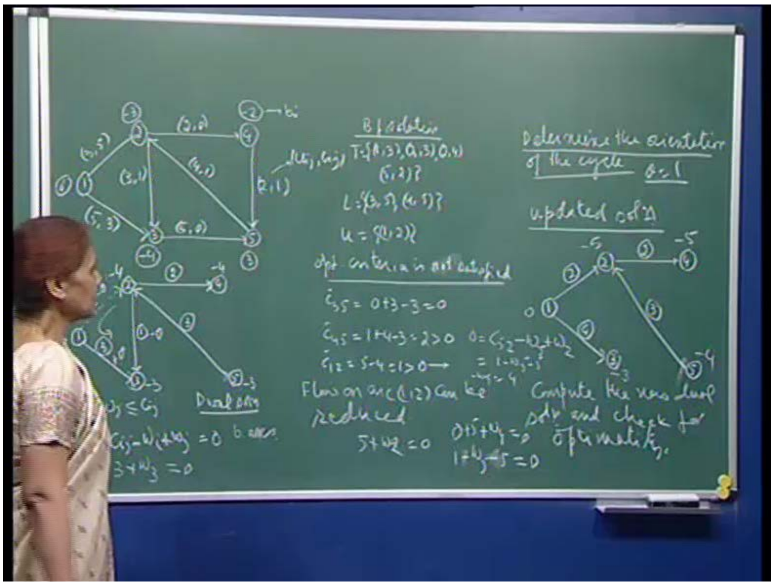
For the basic cells you have equality. So, optimality criteria also you can spell out; so, therefore, how do I now obtain a new basic feasible solution, so going through this example for a min-cost flow problem, here currently these numbers are the b i(s), so they must again add up to 0, so this is 6 and 3, 9, and this is minus 4, minus 3, minus 2, so sigma b i is 0; these numbers first number is the capacity of the arc, and the second number is the cost of the arc; and I have a basic feasible solution shown in this network, and accordingly you see your tree the tree of basic arcs consists of 1, 3 which has a flow of 3 units 2, 3, the arc 2, 3 which has a flow of 1 unit, then 2 4 has a flow of 2 units, and the arc this is 5 2 not 2 5.

So, 5 2 has a flow of 3 units; you can see that all the demands are met, and all the supply is used up at the supply nodes; you can check from here, then so the notation that we are using is our basic feasible solution and is denoted by T, L, u. So, the arcs in L are the non-basic arcs at 0 level, and the arcs in u are the non-basic arcs at their upper bounds; currently, this broken arc indicates the arc, which is at its upper bound, but it is a non-basic arc, so the flow is indicated by the square around the number to indicate that it is a non-basic arc at its upper bound; then we need to compute the dual solution, and you can see that for the dual solution for the for the basic arc this will be 0. For basic arcs, so we start with this w 1 as 0, then you can see from here, because this number has to be 0; so, for example, for 1 3 the cost is 3, so this will be 3, and w 1 is 0, so then plus w 3 is 0 which gives you that w 3 is minus 3.

So, this is how the number is indicated. Similarly, you can compute w 2, w 5, and w 4, because from here around this arc the cost is 1, and in along arc 2 4 the cost is 0; so, accordingly, you can compute the remaining dual variables, so the dual solution is indicated here. Now, you need to check whether the current solution is optimal or not, so we will verify the optimal I mean we will check optimality criteria.

So, for example, for arc 3 5 which is a non-basic arc at its lower bound, the C bar 3 5 would be C 3 5 minus w 1 plus w 5, so which will be accordingly 3 5, the cost is 0, so 0 minus of w 3 which is 3, and then 5 w 5 is minus 3, so this is 0; and therefore, it this indicates the alternate optimal solution; c bar 4 5 you will be c bar 4 5 will be 1, the cost is 1, then plus this was a 4 5, so this is minus 4, so becomes plus 4, and this is minus 3.

(Refer Slide Time: 49:40)



So, 5 minus 3 2 greater than 0 again optimality criteria is satisfied; and for C bar 1 2 the cost would be 5 minus 4, C bar 1 2, because this is 0, the cost is 5, so 5 minus 4 is 1, which is greater than 0, so when you reduce the flow on arc 1 2 cost will come down.

So, therefore, the current optimality criteria is not satisfied, and so flow on arc 1 2 can be reduced; so, then we will determine the orientation of the cycle, because you want to reduce the flow on arc 1 2, therefore orientation is this; and if I want to do the update the flow by theta units, then on this arc it will be minus theta 1 3 has a same orientation as the cycle; therefore, the flow will go up arc on 3 by amount theta, again arc 2 3 is against the orientation of the cycle, so our flow will reduce on arc 2 3 by theta, so therefore minus theta; and now, since you want the new solution also to be basic feasible, therefore theta can be at most equal to 1.

So, the value of theta comes out to be 1. And now, you need to update the flow on the tree, and you see that arc 2 3 becomes now a non-basic arc at its lower bound, because the flow has reduced to 0; arc 1 2 is now becomes a basic arc, because the flow 1 it is not equal to its upper bound, the flow comes down to 2 units, and I have indicated the new updated solution here, which is so arc 1 2 flow is 2 units, these two flows did not get disturbed flow on arc 1 3 has gone up by 1 unit, so that becomes 4 units.

This is just to translating whatever I mean the network the simplex network algorithm that we discussed. So, I am just trying to translate it through this example; and now, of course, lets compute the new dual solution, so once you take this 0, this will remain minus w 3, w 3 is minus 3; and now, you need to do it for this 2, and that 2 came out to be…, see here again the same thing, you will do it in the same way, because this thing is the cost is 5, so 5 and plus w 2 is 0, so w 2 will be minus 5; so, in this case, this will be minus 5, and then for arc 2 4 what is then what are the numbers, the cost is 0.

So, 0 plus 5 and plus w 4 is 0, so w 4 is also minus 5, which makes sense, because the cost c 2 4 is 0; and here also it was minus 4, minus 4 is now this is a basic arc, this is what will happen; and then for arc 5 2 also it should not change, for arc 5 2 the cost is 1, so 1 5 2; therefore, this will be minus w 5 plus w 2, which is minus 5 is 0; so, w 5 will be 4, hold down what do I want to write down here, see I want to say that 0 is equal to C 5 2 and then this becomes minus w 5 plus w 2, so w 2 is equal to minus 5; C 5 2 is equal to how much? C 5 2 is 1, so 1 minus w 5 minus 5 w 2, so w 5 is going to be 4, that is minus 4, so this is that shows you minus w 5 is equal to 4.

So, w 5 is minus 4. You see the positive number should did worry me, because see this is minus 4, you take it to this side become 4 for minus w 5 is minus is 4, therefore w 5 is minus 4. So, you must sort of build your own checks while you are doing the computation; and you can now verify the optimality criteria. I mean, do it check for optimality; and if you see that it is still not satisfied then you would determine the incoming arc, and you will try to then determine the orientation of the cycle, determine the value by which you can update the flow and then continue till you satisfy the optimality criteria.

So, this is the idea took a small example to show you that how by exploiting the structure and you can really simplify the networks simplex algorithm, and get a reasonably efficient algorithm to solve these network flow min-cost flow problems.