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

Module No. # 01
Lecture No. # 26
Sensitivity Analysis

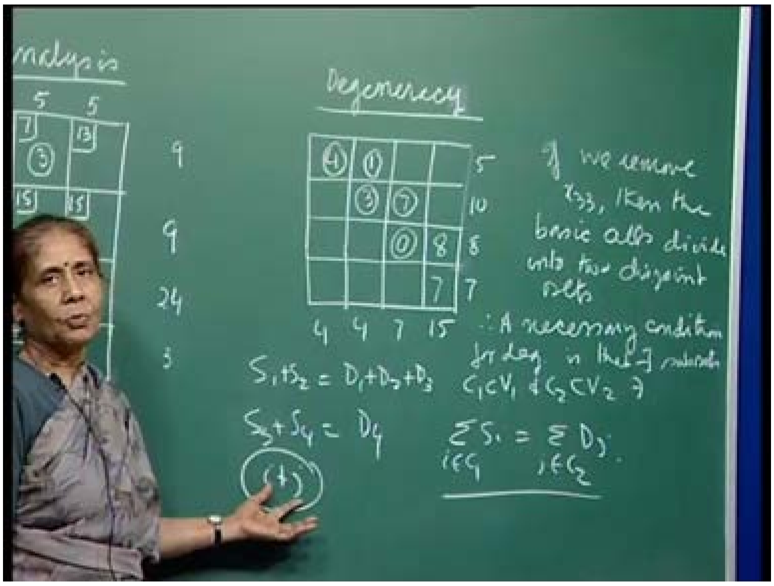
In the last lecture, I hope you got a fair idea as to how the simplex method gets simplified. The calculation part gets simplified, when you applied to the transportation problem. I had worked out a problem for you, not whole of it, but whatever it is. So, now, let me look at the show of degeneracy today, because this also has a role to play, and here for example, if you look at this transportation tableau, I have not given the cost, as it is not important.

So, this adds up to this; 8 17 30 and 15 and 15 30, but what is happening is when you apply the north-west corner rule, we start from here, and then this and this, but then at this point what happens, well essentially, here this is, we have made allocation of 15 and 8 plus 7 15. Both the row constraint and the column constraint get satisfied.

I will refer to this as row constrain which is actually a supplied constraint and these are your demand constraints. So, the column constraint and the supply constraint both get satisfied simultaneously. You can either move here or there, to make a positive allocation.

Therefore, you have to allocate 0 here or there; you could do it either way and then continue with this 8 and then 7. This is where degeneracy will occur and this is a necessary condition. So, what we are saying is that, this is the degenerate term the value of the basic variable here is 0. If you remove;

(Refer Slide Time: 02:00)



That means, if we remove x33 then the basic cells divide into 2 disjoint sets. The first set would contain 1 1, 1 2, 2 2 and 2 3 and the other set will contain the cell 3 4 and 4 4. The thing breaks into two disjoint sets. That means, now we say that, therefore the necessary condition for degeneracy is that there exist subsets c 1 of v 1 and c 2 of v 2 said that summation.

Essentially, before I put down the conditions, what we are saying here is that a summation or let me just write it with the sigma sign say that s 1 plus s 2 is equal to d 1 plus d 2 plus d 3 or also you can say from here that summation or no need to write this; s 3 plus s 4 is equal to d 4.

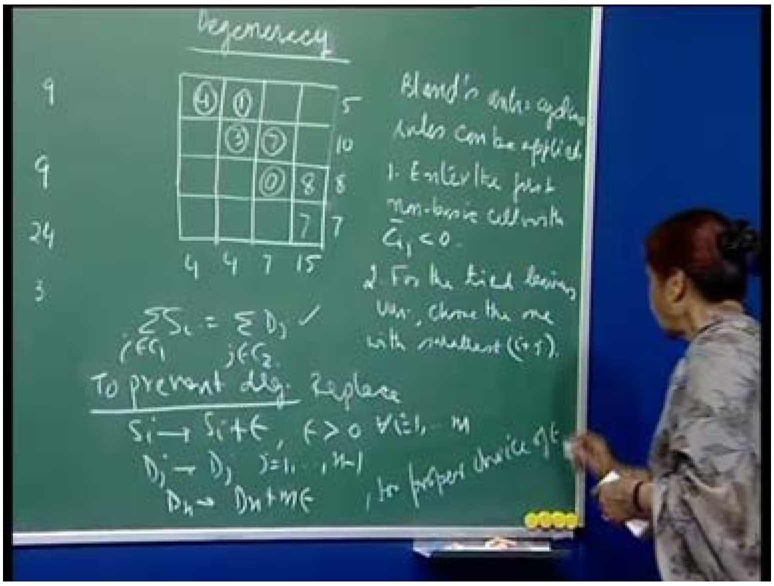
So, either way when it breaks up into 2 disjoint sets; obviously, because all the s i's have to add up to d j's. So, once you have this then you have also this. So, therefore, what we are saying is the sigma s i belonging to c 1 is equal to summation d j, j belonging to c 2. So, you will always be able to find. This is the necessary condition because what will happen is, to obtain any basic feasible solution; we always satisfy either a row constraint or a column constraint.

So, once this happens, the row and the column constraints will get satisfied simultaneously and therefore, you will have 1 less allocation. You will have to go for an o allocation; logic is simple. So, whenever this condition is satisfied and of course, 1 can also prove that this is a sufficient condition.

This is the necessary condition; also it can be shown that this is a sufficient condition for degeneracy. So, this is what it is. So, therefore, 1 needs to come back. So, again the question one can ask or you should ask is that why not the simple Bland's rule? Yes, Bland's rule can be applied here and the rule would be that you enter the first negative c i j first cell with negative c i j (( )) c i j bar is negative, the first one then you encounter and then for the outgoing variable. For the outgoing variable remember what we were doing, because when the ratio was tied, the minimum ratio was tied, we were choosing this smallest index.

Now, here the index are in i j so; obviously, you will go for this sum. So, whenever you are doing a minimum ratio and you have a tie, either the variable can leave the basis there, two of them, then you will select the one for which the i plus j is the smallest. So, that could be the Bland's rule. So, you can apply. Let me just write it down, we can do it, but then I will give you another practical way of handling degeneracy, which has been proposed, but again it is your choice. So, Bland's rule can be applied, Bland's anti-cycling rules can be applied.

(Refer Slide Time: 06:18)



Well why because, the moment there is degeneracy, you know that cycling can occur and I will give you an example also where the cycling does occur. So, Bland's anti-cycling rules can be applied and the rules would be what, but before I. So, the rules enter the first non-basic, with c i j bar plus less than o and second for the tied leaving variables or outgoing variables leaving variables, I choose the one with smallest i plus j because here the index is actually i plus j in our sense. So, this is it, but before let me now show you.

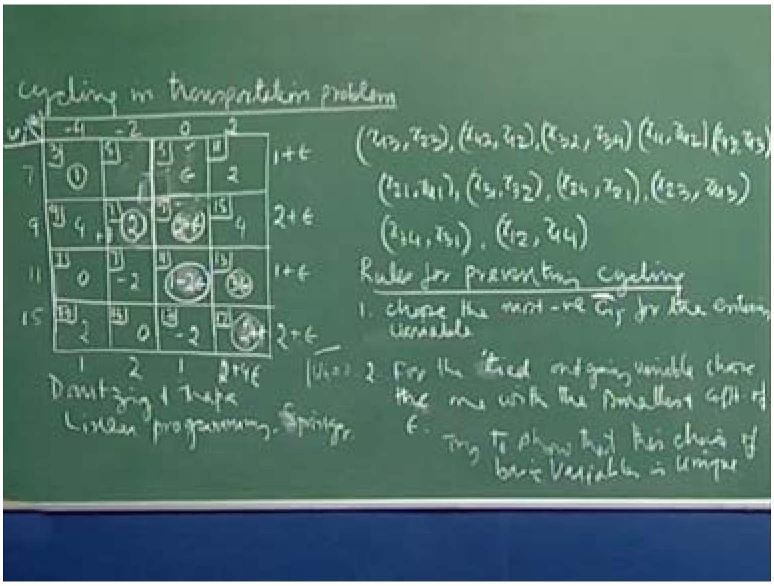
So, since the necessary condition tells you that i belonging to c 1 is summation d j, j belonging to c 2. So, what was proposed can be replaced. So, replace the s i's to prevent degeneracy and this is where you would ask the question, why is this enough, but to prevent degeneracy replace s i by s i plus epsilon where epsilon is greater than 0; some proper choice you have to make for all i varying from 1 to m then replace d the d j goes to d j for j varying from 1 to n minus 1 and your d n goes to d n plus m epsilon; the problem has to remain balanced.

So, therefore, if you added m epsilons here, the demand numbers should also be increased by m epsilon. So, I just increase the last one by n epsilon and we can see why this would satisfy this; the choice of this change in the s i's and d j values will never satisfy this condition. So, the changed s i's and d j's will not satisfy this necessary condition for degeneracy which has also been shown to be sufficient. So, therefore, this claim is good enough that the degeneracy would be prevented, if you replace the s i by s i perceptual and of course, this is a proper choice of epsilon.

So, you have to say somewhere here, for proper choice of epsilon and this of course, epsilon is small enough; also it does make the problem solving by hand little cumbersome, but again mostly large problems get solved by the computer. So, therefore, adding this epsilon should not create any problems. So, maybe we can see it here, now let me give you an example .This is cycling in transportation problem; or quite some time it was believed because empirically no problem was ever encountered in transportation problem which was cycling because of degeneracy.

So, people may decline that may be it is not necessary to worry about cycling in transportation problem, but then came this example which shows that cycling can occur and therefore, people had to address this problem. So, let us see this is a transportation tableau and of course, I have modified the problem so, essentially the problem comes from Dantzig and Tthapa.

(Refer Slide Time: 11:10)



This is a linear programming and I think editors or springer will give you more details later on, but it is a recent book; may be 3 to 4 years old. In this book, they have an example and I have modified it a little bit that does not matter. So, now, what they say is that if you have this s i's and these d j's and you see that condition is satisfied, 1 1 then 3 3 then you have 4 4. So, at every step this condition is satisfied and therefore, you have 3 zeroes in the basic cells. So, 3 degenerate values.

This is the basic feasible solution and starting with this, they said that if you start with this, you will just use the normal simplex algorithm, and then this is the series of incoming and outgoing variables. For example, in the first iteration x 1 3 come in and x 2 3 goes out, then x 4 2 comes in, x 1 2 goes out and this way. So, you see that in the last 4 iterations what happens is that 2 3 is coming back again. So, 2 3 is coming back into the basis then 3 4 has come into the basis and 1 2 has come into the basis.

Once these 3 have come back, you will have the same basis; you will have the same set of basic cells. So, therefore, you will start cycling again because you started from here with these series of iterations pivot elements; then you come back to the same thing, now when you start again you will keep cycling. So, this will create problems and therefore, idea was that we can start with this 1.

Let me show you that if you do this and therefore, I should give you the rules also, so that means, we would multiply. Increase these by epsilon and this 1 will get increased by 4 epsilons. So, what are the rules for preventing cycling? Rules for preventing cycling: the first one of course, choose the most negative c i j bar for the entering variable and for the tied outgoing variables. Choose the 1 with the smallest coefficient of epsilon.

Now,1 would say that how can you be sure that this choice is unique well that can also be proved and these are the things which you can sit down and try to show because remember we are adding and subtracting the feasible solutions because your basis matrices triangular and the determinant of the triangular matrix basis matrix is 1.

You saw that the right hand side values; all these they keep different additions and subtractions of these values will get and so, this is not a very difficult thing to show, this smallest coefficient of epsilon. So, this is a question for you to try to show that this choice of basic variable is unique.

This, I should underline it, it will be good if you can sit down, this will give you good understanding of the algorithm. So, for the outgoing variable you will choose the 1 with the smallest coefficient of epsilon. I will show you 1 or 2 iterations on this, to demonstrate that actually this sequence of pivots will get changed, because of these rules for preventing cycling; and will learn to go beyond that because it will take time otherwise to also to prove at these this rules work as they have been shown to work.

So, for example, here let us quickly do it and another note I want to make is that, we have been saying that, always choose v n o; now this is not necessary because since you are always working with the basic feasible solution, all the constraints satisfied is in equality. Any dual variable can be chosen as o, because you need to choose 1 of them as you need to fix 1 of them and then the other remaining m plus n minus 1, dual variables get fixed uniquely.

So, it is not necessary and in fact, a very good thumb rule, to make your calculations fast is to choose that v i v j equal to o for which the column has the maximum basic cells, because then immediately, a number of basic cells is large or a certain column I choose, that particular v j to be o and then u i's get fixed immediately and then the remaining u i's can be determine and the v j's.

It is not necessary that we should always choose v n equal to o but anyway in this case for example, for v n there is only 1 entry. So, there is no point choosing. So, I can choose any 1 of them to be 0 so; that means, say that this is 0 or I could have chosen this e 1 to be o

Anyone of the dual variables you can fix and then the remaining 1s get fixed uniquely. So, this is the idea.

Therefore, let me choose this is as 0. So, what I am saying is that this will be 9 and this will be 1. So, these 2 get fixed sorry 11 and now let us quickly do it. Once you have this 7, this should be minus 2 because 9 minus 2 should be equal to 7; once this is minus 2 this will be 7 because 7 minus 2 is 5 this is this, then this will be minus 4, and then from here this is 13. So, this should be 2, once that is 2 this should be 15.

Let us now compute the relative prices which is minus 2 here 5 minus 7, then 11 minus 9 which is 2 when here this is 9 minus 5 which is 4 and this is 15 minus 11 which is also 4, this is 7 minus 7 o then 9, so, this is minus 2; the last row you have to compute here, this difference is 11. So, 13 minus eleven is 2 and then this is also 13.

So, this is 0 and this is 15. So, this is minus 2 obviously. So, solution is not optimal because you have minus entries here for the c bars as c bar i j's and therefore, you can improve; now then this is 15 or 2 7teen and so, this is o. Sorry, this has to be 0 because this is the basic cell. So, the computations are ok; this was the basic cell I did not show. So, now, I want to give you this rules for preventing cycling another new method and this is your perturbation method that is called of course, for the general linear programming problem simplex algorithm I had given you rules Bland's anti-cycling rules, but I want to show you that because the structure is special structure here, you can get another simplification here and this is you see, choose the most negative c i j bar for the entering variable c i j bar. So, the first rule is a same as that for the simplex algorithm and then for the tied outgoing variable when because you see the tie occurs in the minimum ratio and so, you have to decide which 1 is the outgoing variable; that choice may not be unique in the simplex algorithm.

Here we say that you choose the 1 with the smallest coefficient of epsilon. So, now, I will explain. So, therefore, the idea that you perturb the right hand side 1 plus epsilon 2 plus epsilon 1 plus epsilon and 2 plus epsilon and here this is 4 epsilons. Because the row sum and the s i's and the d j's must add up to the same number. So, the perturbation has to maintain that balance. So, this is again a balance transportation problem and we said that the choice of epsilon has to be appropriate and I will show you in the process, how the rules have to be applied.

If you want to now update your thing then, here I can make this epsilon and let me also point out that is not the only way of perturbing the s i's and d j's; you can do it differently also let us add 1 plus epsilon 2 plus epsilon here and then add 4 epsilon here. So, you can do it either way. So, this becomes epsilon and then since this is 2. So, you have to have this 1; you have to update this 1 and so, this becomes 2 minus epsilon and since you have updated this, and this here is 2 plus epsilon. So, you need to change this, which becomes 2 epsilons.

Let us quickly do it now; this is again intact this is 1. So, I have to reduce this by 1 minus 2 epsilon and then this will be 3 epsilon because 1 plus epsilon and then here this will become plus epsilon. Now, all the numbers match, the row sums and the column sums. Everything is ok and here you immediately see your epsilon cannot be bigger than half because you do not want this number to be negative. So, this is what we mean by appropriate. So, essentially you say that epsilon is small enough. So that, at no iteration you get an infeasible solution.

So, now this is what it is and you have to choose the most negative incoming variable. So, since all these c bars are minus 2, I can easily take this to be the incoming variable;1 could choose anyone of these and so, let me choose this 1 and then you will form a theta loop and this will be this theta loop can be easily. So, as I said that for small problem when you doing by hand you can use any adopt method to locate the theta loop.

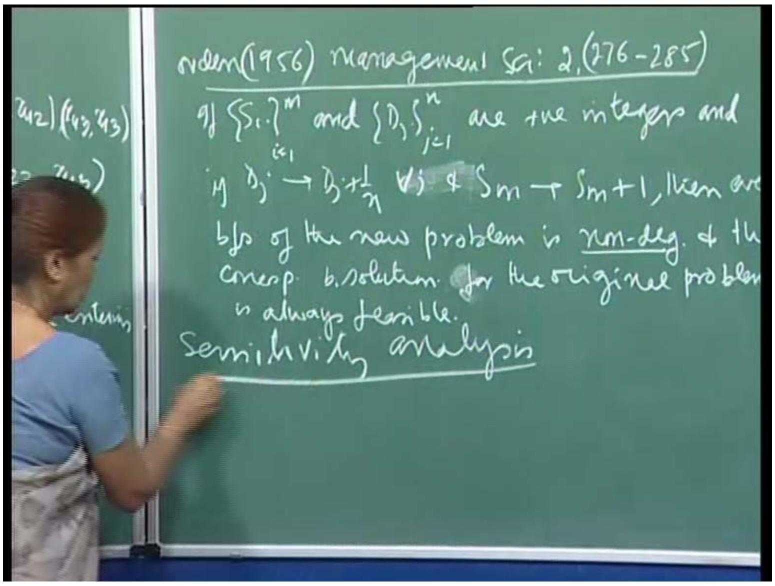
Here this goes by theta, this goes by minus theta plus theta minus theta and here we will be governed by. So, of course, there is no tie as such, but here you see to maintain feasibility, you will choose theta to be equal to epsilon because here this is 2 epsilons. So, this will remain positive. So now, your new solution; that means- I updated. So, this is the incoming variable and here this will become. So, you have to remove this. So, this 0 is gone out and this 1 now becomes epsilon, this 1 you will add epsilon because the row columns sum has to be intact. So, this is 2 and then here this is minus theta. So, this is epsilon.

This is your new basic feasible solution and some of these u i's and v j's will change. So, you continue with the simplex algorithm, because of lack of time and so, I will not go through, but it just happens that now when you recomputed the c bars the next iteration you will land up with the optimal solution.

All the c i c j bars I think 1 more iteration is needed and after that if you follow these rules then you will get optimal solution and this is the point we want to make here, is that Bland's anti-cycling rules, have been known to work very slowly. So, they are not at all suitable for combating degeneracy or cycling in the transportation problem and taking the advantage. So, the epsilon perturbation methods have proved to be very reasonable and they get you I mean of course, what you can do is finish this with the epsilon perturbation method and then try to apply Bland's anti-cycling rules to this problem and just see.

And of course, 1 problem is not a criteria for judging which is a better method, but you will see that certainly Bland's rule will take you for more iterations than what you had to go through for the perturbation method. Now interesting result by Orden and let me just give you, I want to point out

(Refer Slide Time: 26:o2)



Orden in 1956 and this is the general's name is management signs this is number 2 and this is 276 to 285 pages. In general, he adds this article where he has further improved on the perturbation method and he says that. So, here of course, here thought they are saying is that if. So, say that if s I will just code the theorem by Orden which says i varying from 1 to n and d j's are the demands 1 to n are integers of course, positive integers are and if d j are replaced by d j plus 1 by n.

Here he is perturbing each of the demand point demand and then you will do. So, if d j goes to d j plus 1 and s i, sorry and this is, I should say for all j and s m goes to s m plus 1 right because n of these. So, then the total thing would be n into 1 by n which is 1.

So, you perturb the last supply point supply as s m plus 1; then every basic feasible solution of the new problem is non-degenerate and the corresponding basic solution for the original problem is always feasible, before I comment on this.

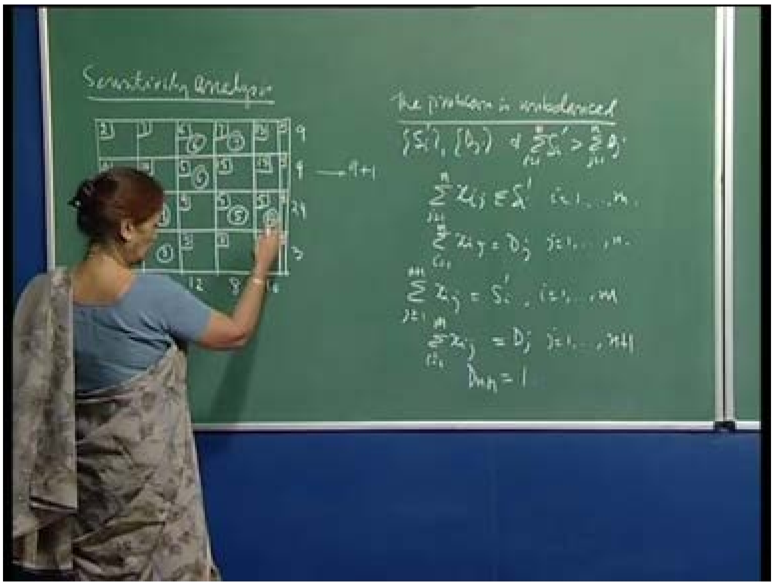
Now, we know that once we have got an optimal solution with the perturb problem then I will get the optimal solution for the original problem; I will just put epsilon equal to 0 and because I have maintain feasibility, my choice. So, that was the point that the choice of epsilon would have been appropriate.

So, that you never got an infeasible solution when you put. So, therefore, what you get when you put epsilon equal to 0 would be again a feasible solution and an optimal solution for the original problem. Now, Orden’s says that you do not have to worry about the appropriateness of epsilon, you just perturb each of the d j's by 1 by n and then s m by s m plus 1 and in the case that what we did we actually perturb each s i by. So, we if you follow Orden’s rule then we would perturb each s i by 1 by m and then the last d n, d n would be perturb by 1. So, that would be the only difference.

So, then he is saying that you just go on with this perturb values of d j's and s i's and then solve using the transportation algorithm and finally, at each iteration you will be getting a non-degenerate solution; that means, none of your basic variables will become 0. So, this is important and therefore, once if you do not have any if you can maintain non-degeneracy all the time then there is no question of cycling and therefore, it takes care of. So, it prevents cycling and that the original problem will always be feasible. So, this is what Orden's results says and you can try it out; may be the same problem you can use this perturbation and then see that you will always get degenerate solution at each iteration and they will be (( )) .

The next thing that we were talking about after taking care of degeneracy is sensitivity analysis. Sensitivity analysis, we will continue with and here I want to say that the idea here is that suppose I change 1 of these s i's or the d j's, how do I, the same question that we posed for the linear programming problem in general was that can I either try to find out whether the current solution is optimal for the changes that I have made in the s i or d j, 1 at a time of course, we will consider that or can I get a new basic feasible solution which is optimal. So, this is the whole idea.

(Refer Slide Time: 31:55)



And now here for example, the issues are different in the sense that suppose I change this 1 to 9 plus 1 and see we always do marginal analysis and since I am dealing with integers you see I can think of changing the s i’s or d j by at least just 1 in number 1 because the solution will since the solutions are at each iteration integer valued.

The idea is that this 9 plus 1, I have more amount available here and then the question asked is, will the cost change? that means, can the cost go down, that is the whole concern and why because you see, it is possible that from this particular source, the cost of sending the material to different markets may be cheaper or the roots some of the roots that is your using in your optimal solution the cost may be cheaper and so, you would like to send more from here and then show the surplus elsewhere; that means, you may not use the whole amount from another source from where the cost of sending the material to different markets is more expensive.

This is the consideration and these are kind of things which you can see here and therefore, for the transportation problem this kind of sensitivity acquires lot of meaning. So, here let me just consider this case of that particular s 2 has gone up to 10 units.

My problem has now become unbalanced; the problem is unbalanced, let me just quickly say that; that means, your s i's primes are there and you have the same d j's, I have not changed them and here also only 1 of the s i's has changed and summation s i prime i varying from 1 to m is quickly greater than summation d j; j varying from 1 to n so; that means, that your constraints. So, for example, now since you do not know, the question asked is where this surplus will show.

So, therefore, I do not know which constraint would be satisfied as equality and which supply constraints satisfied as inequality. So, x i j, j varying from 1 to n would be less than or equal to s i prime, I am saying because I do not know which 1 right and then summation x i j summation over i varying from 1 to m is equal to d j.

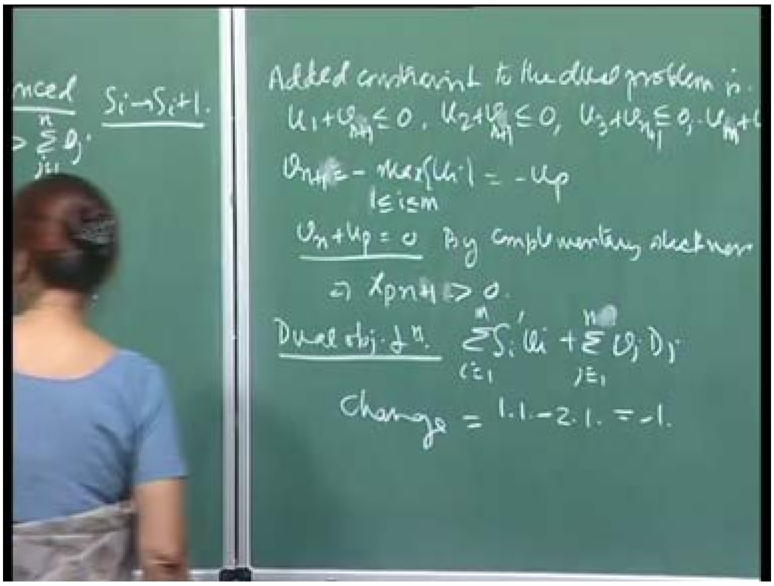
I have to meet the demand constraints j varying from 1 to n. So, then as usual because we have a less than or equal to kind constraint we will add slack variables and what it means here is that you simply add a market, see when I add, because I want I will be adding constraints here to the supply things. So, therefore, for each row there is a variable and since it is playing a role of a o variable; that means, I am adding a virtual market or an artificial market where the surplus will show, but actually there is nothing like that. So, wherever I am not using the complete supply, that is where the surplus will show; that means, the numbers will not add up to. So, somewhere these numbers will not add up to that particular number wherever the supply the surplus has to show.

So, your adding and therefore, now your constraints become x i j, j varying from 1 to n plus 1; I am not writing x s x i s and so, on. It is n plus 1. So, then this becomes s i prime; i varying from 1 to m and summation x i j i varying from 1 to m this equal to d j and here j varies from 1 to n plus 1 and you know that your d n plus 1 is equal to 1, the difference between sigma s i prime minus sigma d j.

Because I have just increased the 1 supply by more than 1 n and now you can see that immediately, because this market, see this column is the new column.

So, in the dual constraint you will simply have constraints added. So, that the added

(Refer Slide Time: 36:22)



constraint to the dual problem is, so, therefore, just since 1 column has added, you can see from here also, but I will quickly do it this way. So, you have a new thing here and you will have a v n you will compute the dual variables also I will quickly do it. What you will have now here is u 1 plus v n less than or equal to o because the corresponding c i j's are o's. So, then this will be u 2 plus v n less than or equal to o u 3 plus v n less than or equal to 0 and u 4 plus v n is less than or equal to 0.

Let us first show you the calculations of the u i's and v j's which I have already done here. So, this comes out to be minus 1 2 4 5 and 5 and then this is 2 1 0 minus 1. So, you can do this by putting this through o and then you can because again I am using the rule that you have the maximum number of basic cells here. So, put this o and then you can make the computations.

Essentially in the dual, remember dual feasibility implies primal optimality. So, here what do you see here? We and I should write, I am sorry if I am doing for the general thing then I should have said this is up to dot dot and this is u n dot. Just please take it that way fine. So, then what we are saying is that your v n is less than. So, if I choose my v n to be equal to minus max u i, i varying from or 1 less than or equal to i less than or equal to m then you see the choice of v n, then all these constraints will be satisfied once you have v n chosen like this when you see I am, say suppose this is equal to minus u p suppose.

Then you have v n plus u p equal to o and so, by a complimentary slackness condition, this implies that your x p n should be equal to now again sorry for the this thing this should be n plus 1 have I write to v n here that is was a mistake this is v n plus 1 and for this particular problem 1 2 3 4 5. So, that will be 6, sorry. So, glad I got the mistake right now.

This will be plus 1 plus 1 right. So, x p n plus 1 will be positive. So, if you have a feasible solution for the primal, you have a feasible solution for the dual and they satisfy complimentary slackness conditions then you know that that solution is optimal.

Here, that means, for the expanded problem when you have you know transform the unbalanced problem to a balanced one and you have this extra dual constraints are these and so, by choice of v n plus 1, here again v n plus 1 equal to this, you see that your new solution will satisfy. So, I do not have to change the other u i's or the other v j's simply choose v n plus 1 to be equal to this and then you see that the constraint is satisfied. This particular is satisfied is equality and so, your x p n plus 1 can be positive.

That means the surplus will show in the pth row or the pth source so, that means, if you had changed, see here I did not say. So, for example, here we were saying that if s i have gone up to s i plus 1 this was our mended, then the surplus will show at the pth source and p corresponds to this number here. So, in our particular case let us just look at it this space this as gone up to. So, your i is 2 s, 2 has gone up to 10. The maximum of this is 2 so; that means, your u i is max of u, u r 1 less than or equal to r less than or equal to 4. So, u i's this and therefore, u 1 this is u 1 which is equal to 2.

So, that means, what it say is that you use all the supply that is available here and the surplus should show here so; that means, how do I transform to my new solution so; that means, this will go up by 1 and because the column some has to remain intact. This will be minus 1. So, that means, here in this route I will only be sending 8 units from the source. So, 1 extra unit will be left here and the extra unit that was added here got used up because you see the difference in the cost.

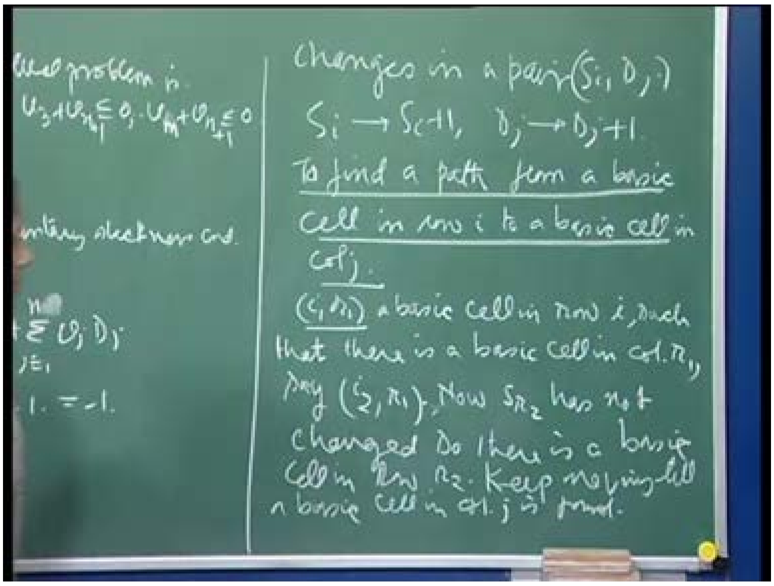
This particular route you charges 5 units 5 rupees and this will charges 6 rupees. So, therefore, that is the different and you can you know from here whichever way you want to do it this is you can see the compute. The difference in the cost and for example, how will you do it? So, you can because this particular route the cost has gone up and this therefore, difference in the cost you will compute; difference in cost is 5 minus 6, this way.

And if you want to look at the dual then what is happening; see your dual objective function, dual objective function, you have summation s i prime u i, i varying from 1 to m plus summation v j d j, j varying from 1 to n plus 1.

So, nothing is changing here because, now, I do not have to take it 2 n plus 1, what would it be? Yeah, we do not have to take it to n plus 1 because I am referring to the original problem. So, the original problem, I am trying to compute the difference in the cost. So, what is happening is that essentially when we do this, my s 2 went up. So, that means, a change in the cost will be that is. So, the change in the dual objective functions. So, this was your u 2 which is 1, 1 into 1 because s 2 went up by 1 and this d 1 by 1. So, the difference is, minus 2 your u 2 into y because s 1 came down by 1 and the cost when the corresponding u 1 is 2. So, I subtract this and this becomes minus 1. So, the 2 cost match and therefore, now, I have just shown you how to tackle the change in the s i, if you change 1 of the d j's then you see; that means, you will have extra demand and that means, here you will have to so; that means, in this case we are only considering the changes in the s i's because the movement you have increased in the d j and you do not change the s i's, then your problem is not feasible because your alloying or either you should then have the possibility that if some particular demand is going down, then some other demand should go down. If some other some particular demand is going up then some other demand should go down, so that you can handle the problem or the other case which I will look at, with you now is that, you simultaneously change a particular supplied point and a particular demand point because remember, for feasibility you have to have either a balance problem and if you want to satisfy the demand constraints as the equality, then if you raise 1 of the demands then some other demand must go down or you allow a demand constraints to be satisfied as inequality and in that case you will have to then add surplus variables and that will require, but surely with the knowledge that you have gained here about how you transform the simplex algorithm to the transportation problem.

You should be able to handle it, but that will require separate treatment which I have not considered here.

(Refer Slide Time: 45:56)



So, the next topic that I want to look at is when you have changes in a pair- so, now, see that means, I want to look at changes in a pair s i d j so, that means, I am saying that a particular s i goes to s i plus 1 and d j goes to d j plus 1. So, now, the problem remains balanced and just a question of adopting the current solution to the changed supply and demand. So, here the idea is that to find a path from a basic cell in row i to a basic cell in column j. So, if I can find this then I can.

Accommodate the changes in s i and d j this is the whole idea. So, see what I will do, so that means, for example, if I have, what we are saying is that, say consider the cell i r 1, a basic cell in row i; now see because only s i has changed and d j has changed. So, the demand for column r 1 has not changed. So, when I choose the cell because I have to raise s i as gone to s i plus 1. So, I can use this particular value x i r i, will go up by 1.

Therefore, when I choose this cell, a basic cell in row i such that there is a basic cell in column r 1, remember the basic cells are connected and I have to get back to this 1 particular property which I will write now. So, what is saying is that you should be able to move to column along the r 1 because the demand for row column 1 column r 1 has not changed. So, there is a basic cell in column r 1 say; what do you want to say here something like i 2 r 1, now here again.

If i 2 is not equal to i 2 is not equal to n; I do not have to worry about that. So, say this thing. Now s r 2 has not changed. So, there is a basic column, basic cell in row r 2. So, the idea is that you just keep on moving. So, to find a path, you have to find a path from a basic cell in row i to a basic cell in column j, this is the whole idea. So, cell in and then go on moving. So, keep moving till a basic cell in column j is found. So, we said that. So, you find a path like this and therefore, and then we will alternatively say for example, let me keep this as this and suppose I increase this to 1. So, the idea here is, I have to look for a cell in this row. So, there are 2 of them, but since there is nothing here. So, therefore, if I cannot increase this, then I will not be able to balance it with any decrease in a basic cell. So, I simply do this one here and in this case we are just lucky because this is 6 plus 1 and this is also. So, the whole total goes up or forget about this.

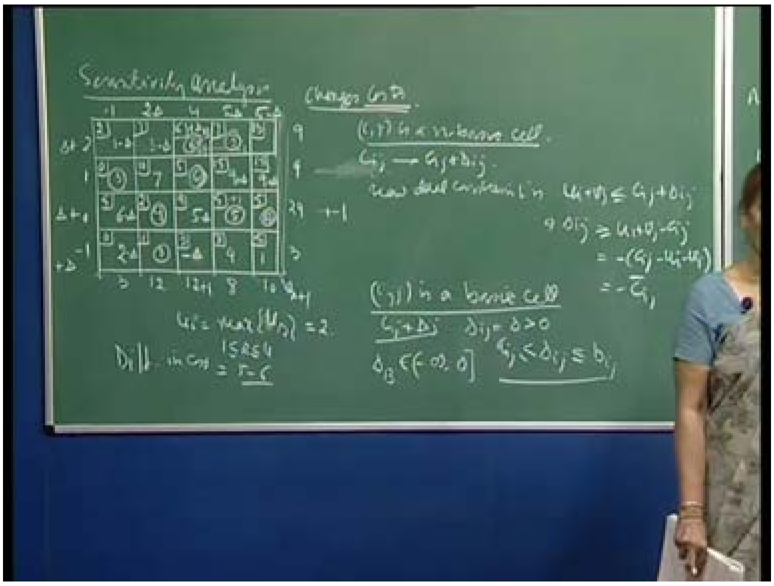
So, here, this is 13 and those are 10, but suppose you had changed this 2 plus 1. So, suppose this went up to plus 1 and this again was this, then what I am suggesting is that I will have to look for a basic cell here. So, the basic cell for example, again I cannot choose this or I can choose this because there is a basic cell here so, that means, if I increase this by 1, I will decrease this by 1 and then I will come here because this row is intact. So, this will become plus 1 and then you see from here I can then increase. So, that is it.

Now, we are talking about this has 24 it will become 25 and this is 13. So, then you see this is now. So, therefore, I found a path with the cell starting in row 3 and then I went up here and this. So, this row sum is intact and this column is become. So, the whole idea, that you find a basic cell in the row when, the supply has gone up and then connected to, with a path and you will always be able to find a path. So, this is a comment I wanted to make is that I define for you a set of basic cells and then I said that there will be.

What we want to say is that a basic cell will always be at least 1 basic cell, will be present in a row or a column for the whole array which we can see by connectivity because if I want to take any non-basic cell in a column and if there is no basic cell present in column or in that row, then I cannot find a path from that basic non-basic cell connected to the basic cells.

Because I need that, for when you have a basis you will always be able to find linear combination of the basic cell basic columns. So, to express that non-basic column and so, the same applies to a non-basic cell and therefore, every basic cell, a basic cell will always be present in a in every row and column and so, the connectivity is there and the technique that we are telling; you will be able to find a path from the row. So, which the supply has gone up to demand point for which the demand has gone up

(Refer Slide Time: 53:13)



Now, the other thing is cost. So, changes in cost; I want to talk about changes in cost and here if the treatment is straight forward because again you have the calculations, you can do directly on the tableau here on the array and what I am saying is that, we can just get rid of this, come back to the thing I have computed the c bars here and what we are saying is that if i j is a non-basic cell and your c i j goes up to c i j plus delta i j then it is very straight forward because the new dual constraint or the optimality criteria dual constraint is u i plus v j. So, c plus c i j and so, here it is a delta i j.

What you say is that your u i plus v j should be less than or equal to c i j plus delta i j. So, this implies that your delta i j should be greater than or equal to delta i j u i plus v j minus c i j which is equal to minus of c i j minus u i minus v j which is minus c i j bar.

That is it; if your delta i j, if the increase in the value of the c i j is greater than or equal to this then your current solution remains optimal. So, you can just read up the numbers from here; for example, this cannot. So, the increase if, is more than minus 1 here the increase is more than minus 3 then your current solution remains optimal and you can read it for the others.

Now, for the non-basic variables it becomes a little complicated, for the basic variables. So, is this is a non-basic cell; now suppose i j is a basic cell, basic cell if it is a basic cell and your c i j plus delta the increase is by delta i j here then you see your u y's and v j's will also change and. So, then you will get an interval for delta i j

For example, here if this number goes up to delta. So, this delta 1 3 and you can quickly compute, I will just show you the numbers here because otherwise it will take time. So, this is 2 minus delta then 5 minus delta and this is also 5 minus delta then here it will become delta plus 2, I am writing delta just to save time because otherwise it should be delta 1 3. So, this what we are writing has should be delta 1 3 and then this becomes delta here plus delta and this is plus delta here

This is your new values and you see what happens is, you can compute it for others it turns out that this comes out to be minus delta. So, right now I am taking delta to be if I am taking delta to be a positive number so, that means, here I am saying that delta i j which I am taking to be delta is positive that you see this will not be feasible, this will not be optimal; the current solution will not be optimal. So, if you are asking a question what happens when my particular c 1 3 goes up to 6 plus delta 1 3 then is there a limit for delta 3. So, that my current solution remains optimal.

You see I have computed, once you change c i j for a basic cell then 6 plus delta this becomes and because you u i's and v j's depend on the basic cell. So, they see the calculations change all the c bars almost all the c bars except for these 2 changes. So, 1 minus delta 3 minus delta and so on.

So, that means, here you would essentially. So, I am talking in general that you would have a interval for delta and if your delta is in that interval then the current solution will remain optimal, but in this case you see what is happening is that this particular number is coming out to be minus delta so, that means, if for delta, I am using delta 1 3 is in the interval minus infinity to o, see then the current solution will remain optimal; that means, if delta if I take it to be negative see here I said 1 is positive then I mean this will be negative. So, this will remain positive if your delta 1 3 is interval minus infinity to o.

And the moment your delta becomes positive, then this is a negative number and then this will be a candidate for coming into the basis because you see that from here otherwise all other values remain non-negative for like delta less than or equal to 1 delta less than or equal to 2 and so on.

This will be the first variable to come into the basis and you can enter this into the basis and continue with your optimal solution; otherwise when you get it interval say for example, if you get an interval for i can use the number let us say some b i j and a i j. So, you will get an interval like this and for delta i j greater than b i j or less than a i j, the corresponding cell will be a candidate for entering the basis and then you can continue with your simplex algorithm to get a new optimal solution

You can handle the changes in the costs here and I have also shown how to handle the changes in the s i's and also the changes in s i's and d j's. Now the changes in d j's require a special treatment which hopefully, I will mention sometime later on.