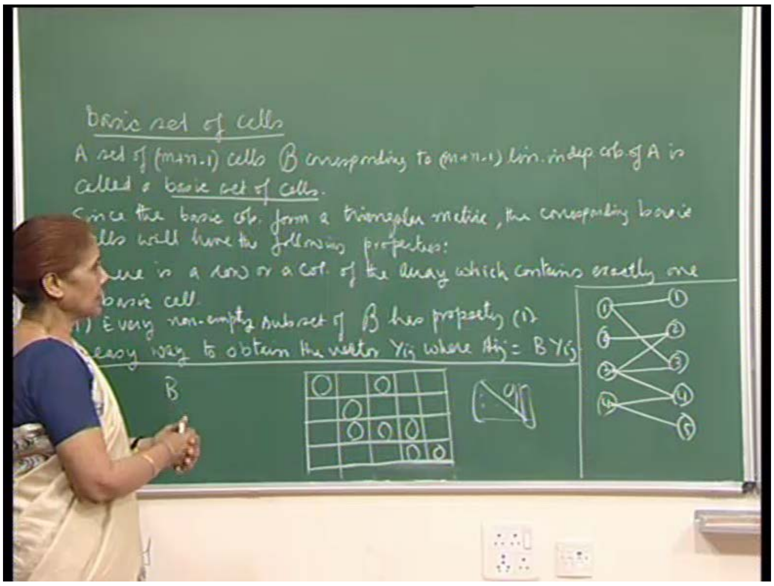
Linear Programming and its Extensions

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

Lecture No. # 25
Transportation Problem Degeneracy Cycling

(Refer Slide Time: 00:31)



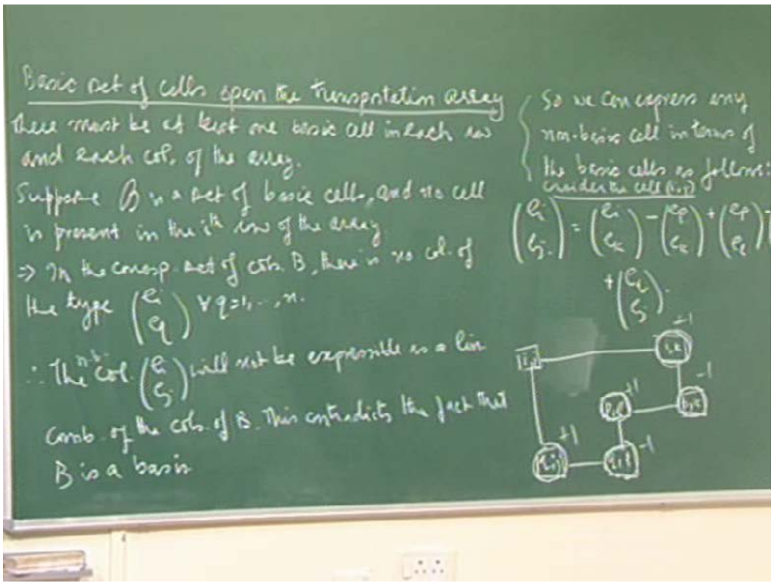
So, I will continue with the translating the concepts of a simplex algorithm to the transportation array. So, let me just begin with, the first thing that we are going to talk about is the, basic set of cells I am defining. Now, see we have m plus n minus 1 linearly independent columns of a, which form a basis for the columns space of a. And so, then, the corresponding cells as I told you, that the 1 1 here means it is the a 1 1 column, a 1 2, a 1 3, and this is how we had said that the transportation array is represented.

So, you take the m plus n minus 1 linearly independent columns of a, and the corresponding cells that you will take in the array, they will form basic set of cells we will call them, right. And later on, I will show you that, it is not necessary that every set of basic cells will give you a feasible solution, just as in linear programming problem, we saw that every basis will not necessarily give you basic feasible solution; it will give you a basic solution. So, we will look at that later on.

Now, what we are saying is that, since the columns I showed you, transforming in fact just interchanging the rows and the spaces of columns or rows, one kind of transformation you can do. And then, you can show that the set of columns of a basis can be reduced to a triangular matrix. And so, when you translate that property to these basic cells basic set of cells, then what will happen is, that this following properties you can see, that there is row or a column of the array which contains exactly one basic cell.

Now, this is equivalent to the fact saying that, in a triangular matrix, you remember, if you have a this is all 0s and you have entries here, then the triangular array, you see for example, if you look at this column, then there is only one entry. And then, when you delete this column and this row, then again they will be only one entry. So, this is the kind of property that you have in a triangular matrix; we are translating it to the set of basic cells, and so we are saying that, if there will be at least one row or column of the array, which contains exactly one basic cell. And then, if you take any non-empty subset B, then it will also have so here I am calling this script B.

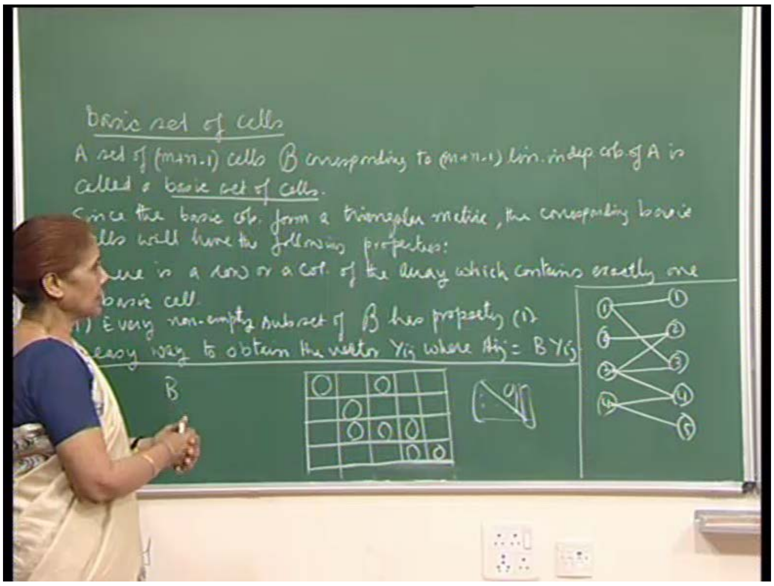
(Refer Slide Time: 03:19)



I am denoting by script B, the set of basic cells corresponding to the basis B of linearly independent columns which form a basis. So, script B, you take any non-empty subset, that will also have property one. So, this is the translation of the triangular property of the basis matrix to the set of basic cells. Then, another thing - another property - that we want to say here is that, the basic set of cells will span the transportation array. Now, what does that mean? And I will again, may be just give you a very ad hoc treatment of the whole thing. And then, later on, when we do some develop some more graph theory and for networks, then the things will fall into place, but here what I am saying is that, they must be, so what do we mean by basic set of cells spanning the transportation array?

So, they must be at least one basic cell in each row and an each column of the array. So, they cannot be any row or any column of the transportation array, which does not have a basic cell present in it. And how can I support? I can support it is by the fact saying that, because you see if the corresponding set of columns B, there is no because if write. Take the basic set of columns, and what we are saying is that, if no cell is present in the ith row of the array, suppose B is a set of basic cells and no cell is present in the ith row of the array, what does that mean? That means, in the corresponding set of columns B, which is the basis for the matrix A, there is no column of the type e i e q, because if there is no cell present; for example, in the first row here, that means, I do not have columns of the kind, e 1 e 1, e 1 e 2, e 1 e 3, e 1 e 4, e 1 e 5. So, similarly here, for the when they are number of columns is n, I do not have columns of the kind e i e q, for q varying from 1 to n; what is that mean?

(Refer Slide Time: 05:42)



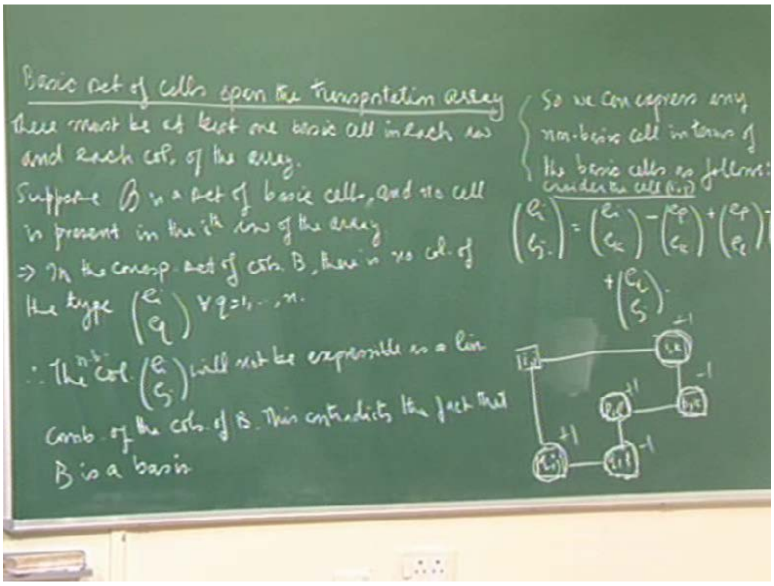
That means, if I take a non-basic column e i e j, then I cannot express it in terms of the basis columns of B, because since B does not have a column of this kind, you see how will I express this, because I need a one in the ith place to express this column in terms of the basis columns. But if the basis columns do not have any column which has one in the ith place, I cannot express this as a linear combination of the columns of B. So, this will contradict the fact that B is a basis.

And therefore, this is not possible; so, for each, that means, a basic set of cells, there will must be at least one cell in each row and in each column. And now, let me try to demonstrate to you, what we mean by that; so, for example, if I take this 4 by 5 transportation array and I have this collection of a basic cells, then I am trying to show, first of all, I am trying to show you the triangular property of the basis; and therefore, that means, that you see there is at least one column in which there is only one basic cell.

So, if you now delete this, then in the remaining subset of basic cells; you now have this cell, which is the only cell in the row. And then, if you now delete this, then you can again find that; for example, this one is the only cell in its column, you delete this; then, this is the only cell, which is present in this row, I delete this; then, this will be the only cell present in this column and so on.

So, once you do this you delete this, then you will have this one as the only one; then, I will delete this row and so on. This will be the only row, I mean, this will be the only cell present in this one; so, you delete this, then they it will come. So, you can anyway you try out, they will be always be a subset of the basic set of cells, such that only one of them is present in either a row or a column.

(Refer Slide Time: 07:08)

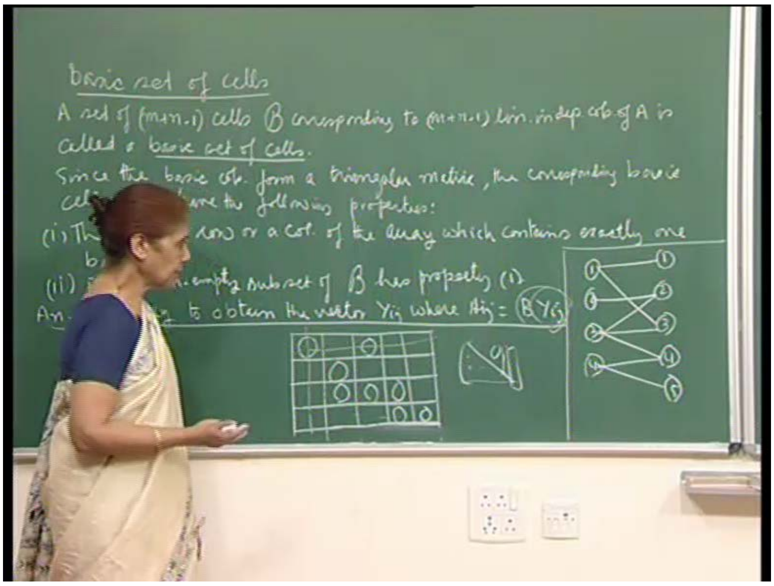


So, this is the triangular property. The second property as I am saying that, since this has to be present in every row and every column; now, again in a very ad hoc way, I am trying to say is that, we can express any non-basic cell in terms of the basic cells as follows. Now, of course, this property is anyway already there, because since the basic cells form basis, therefore, correspond to the m plus n minus 1 linearly independent columns. Therefore, any non-basic cell will also be expressible always in terms of the basic columns, and so, any non-basic cell will be expressible in terms of the basic cells. And remember, how we were doing it; we said that, if you want the column e i e j to be expressible in terms of the basic columns, then you see you first look for a column which is in the same row; so, I go e i e k which I am saying.

So, this the these actually these are the circles; so, these indicates the basic cells, this may non-basic cell. So, I will go across, and remember because there is a basic cell present in the ith row, so I can go from here to here; this is the idea. And suppose this cell is present in some column k, so I will have something like e i e k; this is my basic cell. Then, since there is there has to be at least one, because remember I have to express this in terms of the basic cells. So, I should have, because there is a plus 1 here, which is not here, then I must have a column which has a minus 1 here. So, the 2 cancel out this one; that means, I am able to traverse along the column - they kth column.

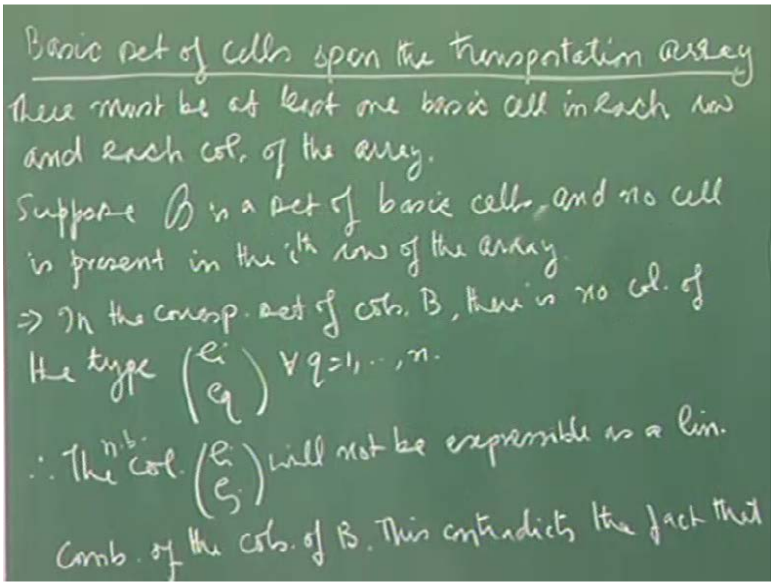
So, I will come down and may be this there is something like p k; there is a basic cell here in the pth row. So, I will have a column e p e k; then, again because I want to counter this minus 1 by plus 1, so I must have a column like e p e l; and therefore, they must be a basic cell of the kind p comma l. So, therefore I will travel along the row. And then, similarly, again you will travel along the lth column, because you want to counter in this plus 1 by a minus 1; so, I come here, and then I will again travel along the qth row, and this I will come to… so, of course, I will just demonstrated one arbitrary way of connecting i j with the basic cells.

(Refer Slide Time: 10:01)



So, essentially what we are saying is, that you will be able to form up a cycle and this is the only arc which is the non-basic cell, and the renaming basic cells they form a connected path like this and then you have this. Therefore, now you immediately know, that i j can be the column a i j, can be written as a linear combination of these basic columns. So, I am just trying to make you conversant with the array and so you can translate of the concepts that we learnt, for expressing a non-basic column in terms of the, because remember our aim is to finally be able to obtain the column Y ij, because to progress with this simplex algorithm, I need the components Y ij. And which are, as we know are plus 1 minus 1 or 0s, but so I need to know which are the plus 1s or able to continue with the simplex algorithm; and therefore, this is all is being done, so that I can just by having the transportation array; write here, I can compute my the necessary components of Y ij and continue with the simplex algorithm, this is the whole idea. and I am just so the connectivity

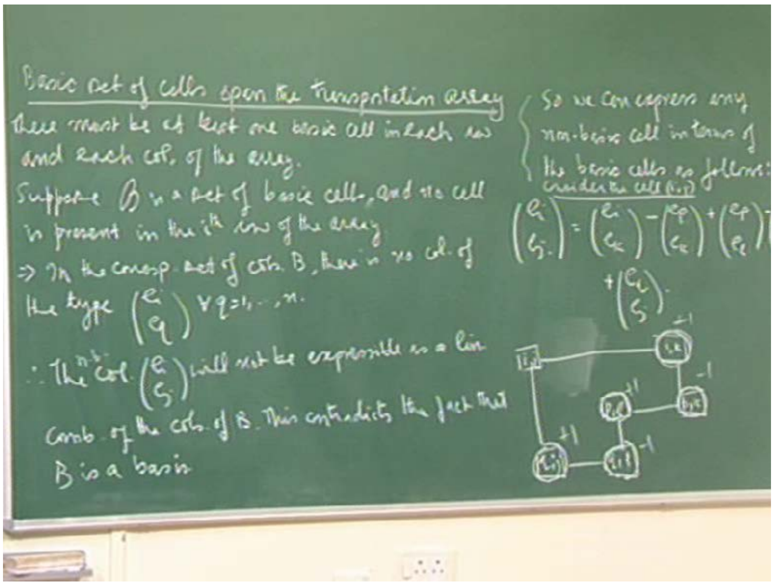
(Refer Slide Time: 10:47)



So, when we say that the basic cells span the transportation array, the whole idea is that, the basic cells are in such a way, that I can always take a non-basic cell. And then, connected to the basic cells, so that I form a cycle, and then immediately I get how to express my column corresponding to the cell ij in terms of the basic cells; for example, this would be plus 1, this will be minus 1, plus 1, minus 1, plus 1, and all other Y ij components will be 0s.

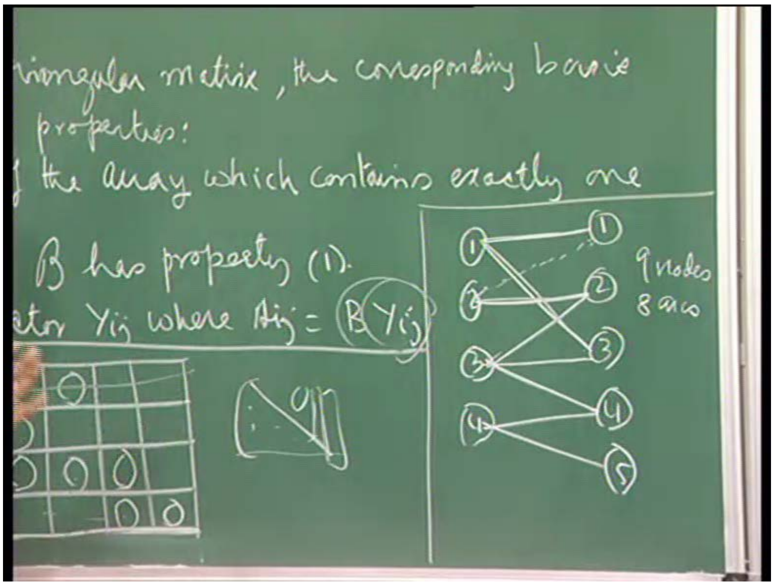
And here, again these concepts we will later on also spend some more time on it, when I do network flows and so on. So, I am trying to say is that, C corresponding to this transportation array 4 by 5, this is my bipartite graph. And you can see that, these basic cells the correspond to this basic arcs, which are my, right; so, this is the thing and if you want to have a particular, see here, if you have the arc 2 1 which is a non-basic arc, then you immediately you see you can find out the cycle that you will form here; so, with this one, see you will go this way, this, this, this, this, this and this. So, here also, essentially the word that I have I am not using it right now, but see this is a graph and here the m plus n with m plus n.

(Refer Slide Time: 12:32)



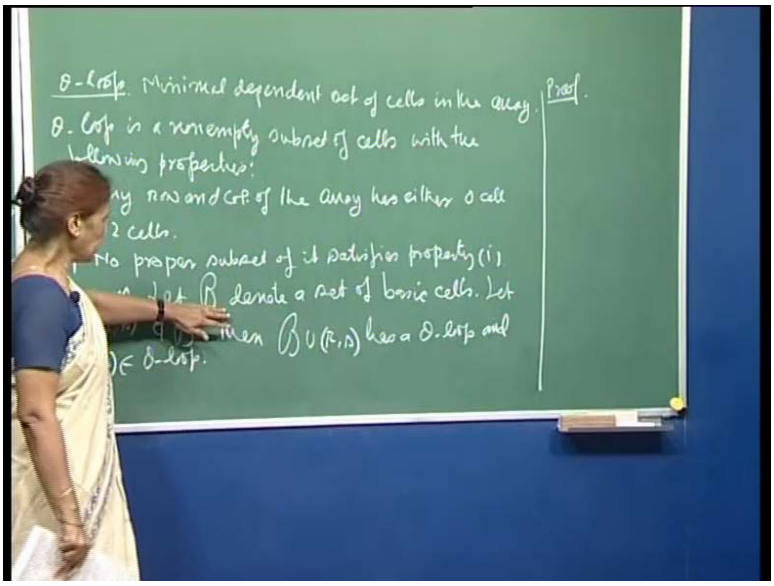
So, here for example for it is 9 nodes; so, I am saying that you take eight arcs; and they correspond to your basis and they will form a spanning tree; and the spanning tree means, and that is why I use the word span there, so spanning tree now you take any non-basic arc, and then, it will form a cycle with the subset of the basic arcs.

(Refer Slide Time: 12:37)



And then, you will immediately know how to express this columns or this arc in terms of the basic arcs. So, this is the whole idea. So, all these concepts are the same, I am just trying to make you familiar, so that once we work out the simplex algorithm on the transportation array, you can immediately see what all is happening. Therefore, now the next step would be, to find out, how we write down the components of Y ij; and for this, I will need the concept of a minimal dependent set, and so will formalize this computation of the components of Y ij.

(Refer Slide Time: 13:20)

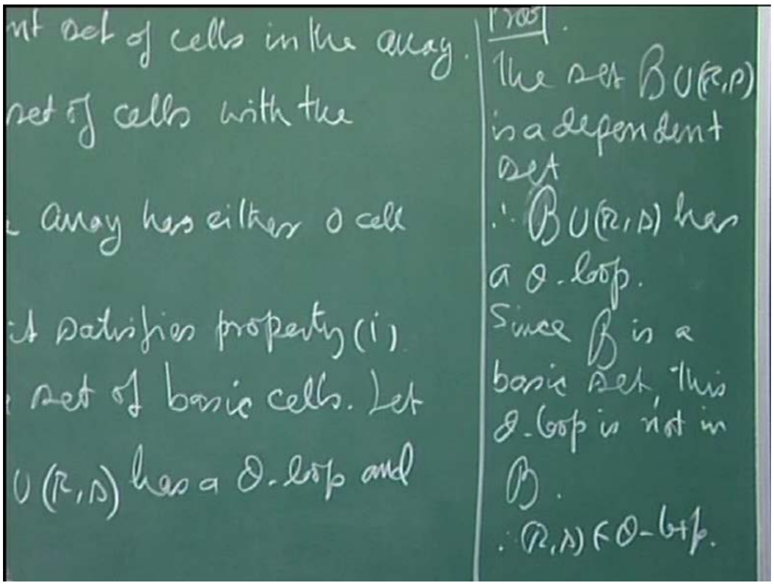


So, the concept of theta loop; I want to introduce the concept of a theta loop and this is a minimal dependent set of cells in the array. So, concept of a dependent set; I have already told you what we mean by an independent set of cells. Now, we talk of a dependent set; so, theta loop is a non-empty subset of cells with the following properties. The first property says that, every row and column of the array has either 0 cell or 2 cells from this theta loop. So, I am trying to say that, it is a subset of cells, and when does it qualify to be theta loop. It is that, every row and column of the array has either a 0 cell from this set or 2 cells. And no proper subset of it satisfies property 1.

So, you see the moment, I say that, no proper subset of it satisfies property 1; that means, no proper subset of this theta loop of the set of cells will have a two cells from it in a row or a column. And remember, by now, you should have this clear in your mind, that if you have a collection of columns, such that each of them has two non-zero entries, which is 1 1. Then, we know that set of columns is linearly dependent and the same concept is being translated to the cells of the array. So, the array has either 0 or 2 cells; so, no proper subset of it satisfies property 1. So, therefore, you can see that it is a minimal dependent set. So, this tells you this and let me know give you this theorem. Theorem says that, let beta denote a set of basic cells, which means that, there are m plus n minus 1 cells here, which correspond to m plus n minus 1 linearly independent columns.

And let cell r s not belong to B; so, it is a non-basic cell, r s is a non-basic cell, all the elements of B a script B are basic cells. Then, union r s has a theta loop and the cell r s belongs to the theta loop; so, the theorem is great forward and I can write down the proof also. What are we saying here, that you start with the set of basic cells, which are m plus n minus 1 in number; and then, you add non-basic cell to it. Therefore, the number becomes m plus n, and then, we are saying that this must contain a theta loop, and the cell r comma s must belong to this theta loop.

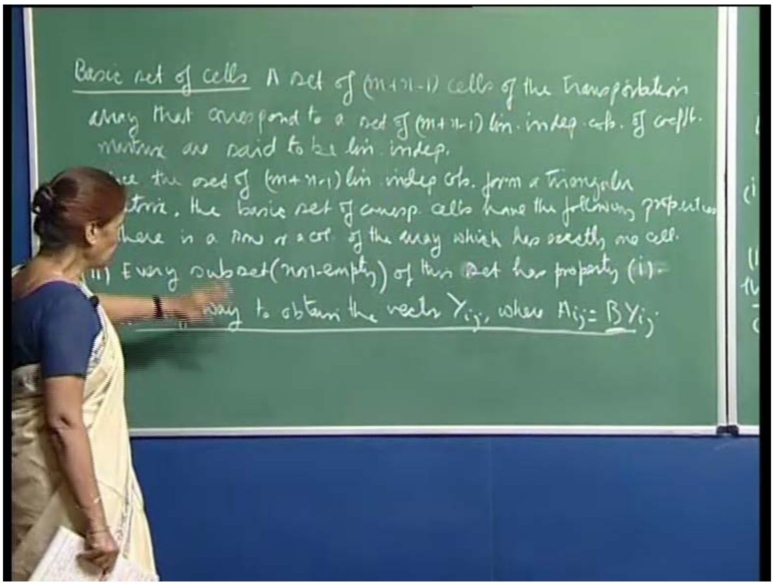
(Refer Slide Time: 18:54)



So, now, since B union r comma s has m plus n cells, that means the corresponding columns are m plus n in number, and we know that they cannot be linearly independent, because the rank of the matrix has been shown to be m plus n minus 1. Therefore, the set union r s is a dependent set, and script B was not a dependent set. So, therefore, your script B union r comma s has a theta loop, may be, I will agree that I am not giving you a very regress exposition here, but the idea is that, you just understand how we are translating the concepts of the simplex tableau of the calculations to the transportation array; so, but has a theta loop.

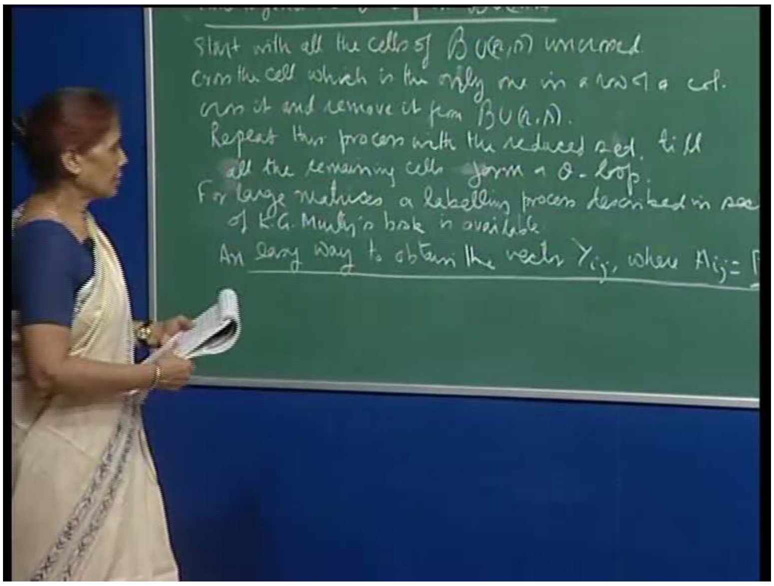
And because one should talk of a uniqueness of a theta loop, which also I will mention and try to give you some rational for it, but this has a theta loop. And since script B is an independent set is a basic set, this theta loop is not in script B; therefore, the cell r s belongs to the theta loop. You might say that, that is simplification of the arguments and so on, but any way this tells you that, because this is a dependent set, so it must have a theta loop. And since script B is a basic set, it cannot have a theta loop; so, the theta loop has come because you have added the cell r comma s; so, the theta loop must contain the cell r comma s.

(Refer Slide Time: 20:49)



Now, this and as I told you I want to this all this is leading to trying to obtain the vector capital Y ij, through they which has the components to how express the column A ij in terms of your basic columns; so, this what you want to do.

(Refer Slide Time: 21:03)



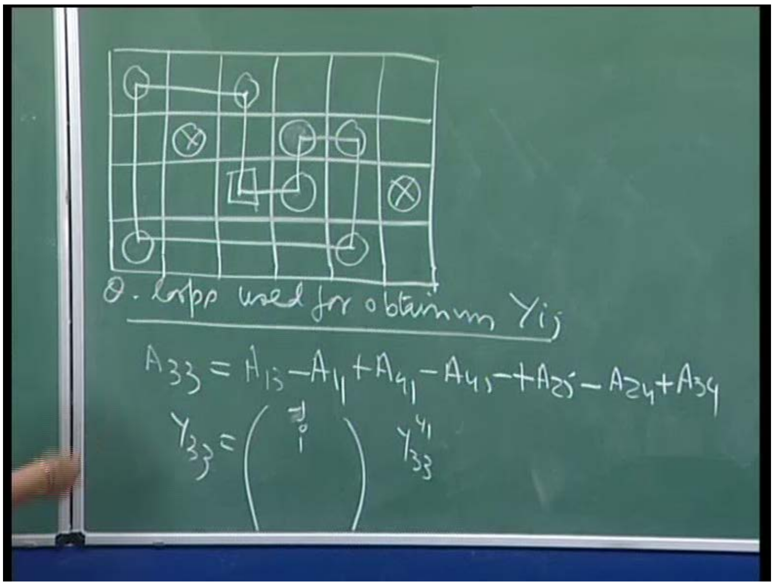
First of all, how to find the theta loop in… so, this is the next part. And here, again I will give you method, so what it says is that, how to find the theta loop in this. So, start with all the cells of uncrossed, so you begin with all the cells of script beta the script B union r comma s uncrossed. Now, cross the cell which is the only one in a row or a column; so, out of this, you will set certainly find one cell at least, which is the only one in a row or a column, and cross it and remove it from union r comma s.

And then, you repeat the same thing with the reduce cell repeat; this process with the reduce set, till you do this, still all the… what I want to say is that, you will repeat this, continue with this process, still you are able to find one cell, which is only cell in a row or a column. And when you cannot find cell from this, which is the only one in a row or a column, we will stop, which means that, the remaining cells that you left with have the property that every row on a column has two of them. So, this what I want to say; so, repeat this process with the reduce set, till all the remaining, maybe I suggest remaining till all the remaining cells form yes I wanted to rephrase the sentence

So, that means, all the remaining cells now will form a theta loop; that means, they will have this property at every row or a column has either two of them or has 0 of them. So, therefore this is the idea, but now the thing is that, you may for when you doing small problems by hand, you may actually just use an ad hoc method to locate theta loop, but otherwise this is the way to do it. And for very large problems, when you have 100's of rows and 100's of columns, then regular labeling algorithm is used for locating a theta loop, because that itself will require lot of effort, and you want to streamline it, so that you do not spend too much time.

So, for large matrices, so one can just use and that is what is will happen. When I am solving a problem, I will just use an ad hoc method and may be not always use this method. By the labeling algorithm of course has to be programmed, and then it has to be part of a software for solving transformation problems. So, for large matrices, a labeling process described in section 11.4 of K G Murthi's book; I have already given references K G Murthi's book is available simple labeling procedure, but it certain streamlines the search for a theta loop, when you have a large array and you want to locate a theta loop; therefore, this can be done. Now, let me get to this, how do we try to get these numbers the vector Y ij, and so, let me just first show you, give you an example here quickly.

(Refer Slide Time: 26:59)

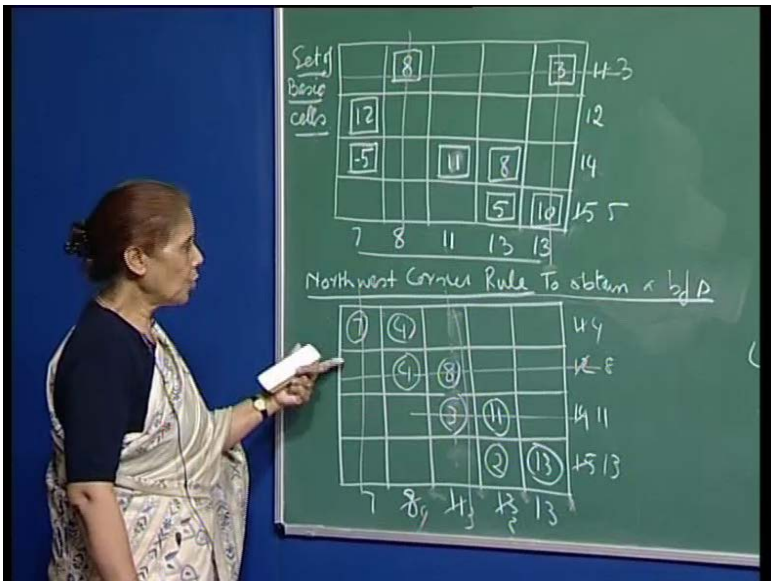


See, for example, if you have a 6 by 4 array, and I will just show you the basic cells here; this is a basic cell,, this is a basic cell and then you have a basic cell here; do not take the numbers, just the collection of a cells. And then, so 6 4 10; so you should have 9 basic cells 2 4 5 6 7 8 9. So, these are the collection of your basic cells and suppose I have this non-basic cell here, then it becomes, if this set of basic cells which is 10 in number, must contain a theta loop; and therefore, see I will to locate the theta loop, we said that we look for a cell which is the only one in a row or a column; so, I will cross this. Once this is gone, now you look for another cell, which is the only one in a row or a column, and this is the 1. And now, after this you see, every row and column in which the cells are there; so, these are 2 and so you have a theta loop. And if you able to connect the things, so here you see, you can do this way, this, this, so this is the thing; so, you have a theta loop. Now, I want to quickly show you how to locate.

So, now theta loops use for obtaining Y ij, as I told you earlier; so the thing is that, now I will probably just give you an example here, instead of and then you can always formalize it; see for example, now this is the non-basic cell and you would want to express it in terms of the non-basic cells. So, this is for example, this is 3, 3; so you having A 33, and how can you write it now? You see again as I have explain this procedure to you earlier, you are going up here, therefore this will be A 13; then, you are coming down this way, alternatively plus and minus. So, you come in down to minus A 11. Then, when you come down here, this will be plus A 41. And I will let you do the arithmetic A 41; then, 1 2 3 4 5; so, minus A 45. And then, you are going up; so, this will be plus; this is what 1 2 3 4 5. So, A 25 and minus A 24 plus 1 2 3 4, A 3 4. Please check out the arithmetic that this will be this is the representation and you see through the theta loop, you can get the, that means, your component of when you write the vector Y 33, you see this will be, because you have n basic cells, therefore it will be an n dimensional vector; and so here you start see if you have the convention that, this is the first component, second, third, and so on. So, accordingly, you will have something like 0 0 1, then you will have A 11. So, A 1 1 is minus 1 and so on.

So, in any case, this is the expression that you can immediately just by looking at the array having the theta loop, in which your cell is present, a non-basic cell, then you can have this expression and you can find out the components. And this we will later on also use, when we are doing the parametric analysis; and see actually what I should do is, if I want to look for a particular component here, see for example, this one, then I should have refer to this as you know 41 or something like this, but later on when I am doing the parametric analysis, I really do not distinguish between; the idea is that you are taking a particular component from this vector Y 33, and as you see, as we have also already mention because the triangular property that the components of Y 33 will be either 0 plus 1 or minus 1, which is evident from here. So, the components of this will be this. And now, you see that, once you have these numbers, then you can immediately obtain which I will explain to you later on also.

(Refer Slide Time: 32:34)



That is, if suppose this is a cell which is to become basic cell, then you will have plus theta here, minus theta, plus theta, minus theta, plus theta, minus theta, plus theta and minus theta. And so, from here, a putting the constraint of optimality feasibility, because you want this to come at a level theta, which means that, you must make sure that the none of the basic variables becomes negative. And so, from there, you will be able to get a value for theta and so you can obtain a new basic feasible solution.

Now, I am telling you that, if you have a set of basic cells, one can run this labeling algorithm to find out if there is a theta loop or not, or in the process, when we are locating, when we are giving out the values to these basic cells, we will discover whether there is a loop or not.

And you will see how; so for example, here where is a basic set of cells, and remember there will be at least one row or a column, which has only one basic cell, because all rows and all columns cannot have to; otherwise, it will become a dependent cell. So, I see here, that in this columns, this is the only entry. So, we will make will give the value 8 to satisfy. So, once this is satisfy. this is out; this constraint is satisfied. Now, because I have made allocation here, so I move along the row and I will find entry or you can up keep applying the same rule; that is, I first look for a row or a column which has only one basic cell in it. So, I make the allocation according to which our value is the minima of the 2 - the row and the column number. So, this become this; then, you see this becomes 3, because I have already made 8 units of allocation in the first row. So, that means, the first market is from the first supply point, I have already of use 8 units.

So, then, I look for this row and I look for a row which as only a single entry. And this now has single entry, row or a column whichever; so, this as this. And since for this one 3 is a left over now, and 13 is this, so minimum is 3 and I cannot make a location of 4 and 3.

So, then this row is complete; that means, this supply point is completely used up. So, then, if you come down you see, again you look for a row or a column which has a single cell, this is the 1. So, here I can make allocation of 10, because this is 30. So, this will be gone and this will get updated to 5; then, you see that, here this row has only single entry. So, 5 and 13; I will make 5, then accordingly 8 here and this is 6. This is not helping me, because so what is going wrong; why did I see if I make this here, then this has to be 8; and so, that means, of course, if you look at, so do I have no so, this is a set of basic cells; so, what is happening is that if I make 6, yes fine [FL]

So this is not yes I will tell you what is the problem. See here this is 11, and this if I make a this thing 8 here, then this will be 6. So, there will be a problem, if I make 6 here, then this remains; so that means, I can only do. So, this will not be satisfied; therefore, what I need to do is, I have to make 11 here, because this is the only entry in this column, so I have to make 11 here. And since this is becoming now how much 19, I will have to make entry of minus 5 here.

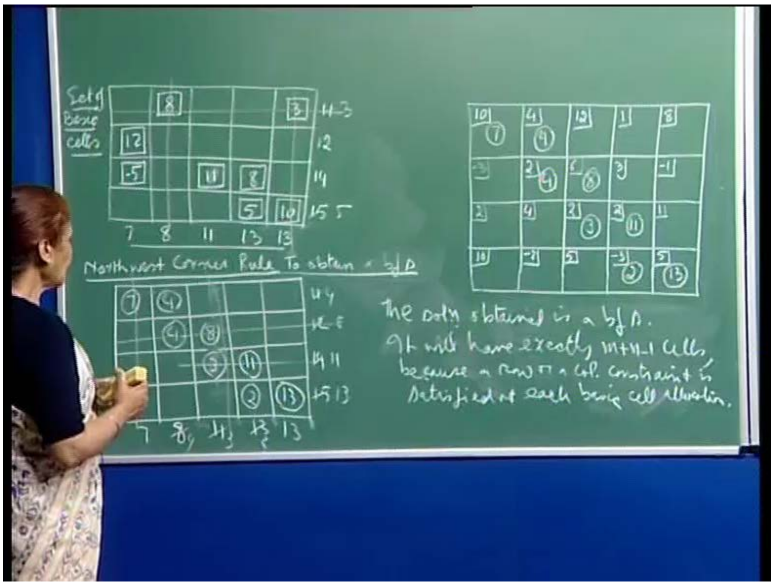
So, I do not use and we have to understand what we mean by this side. So, minus 5, if I do it, then you see this is 12; this is the only entry and then this is 7, because my requirements is the same as the number of amount of supplies. If you add up these numbers, they all add up to 23 and 14 is how much? 23 and 14, 37, and 15 52. So, both the numbers add up to 52. So, this apply is equal to it is a balance transportation problem, no problem, but see so this is what I wanted to show you through this example, that this is a set of basic cells. It satisfies the conditions for being a set of basic cells, but it will not lead you to a basic feasible solution; which means, so as you saw in, for example, when we were looking at the system a x equal to b x greater than or equal to 0, we had a basis which would need not lead us to a basic feasible solution; it will give us a solution to this system, because I have a basis here. So, it will give me a unique solution here, but the components of the solution may not be all non-negative. So, it will not be a feasible solution. And this is what I have shown you that, every set of basic cells need not lead you to a basic feasible solutions; so, this is not a basic feasible solution.

And now, I want to give you a rule for obtaining basic feasible solution very easily and that is known as the North West for in a rule. And later on, we will look at many more methods for obtaining a starting basic feasible solution; so, here, the idea is very simple; north west corner rule to maybe I should complete the thing to obtain a basic feasible solution; this is b i.

So, the next question is, how do we go about obtaining a starting basic feasible solution? And we do not need phase 1 or anything here, because of the structure we are exploiting the structure and you see how simple it is to obtain a basic feasible solution. And we already said that, once it is a balance transportation problem, it will always have a basic feasible solution. So, let us see, what you do is, you start from the North West corner rule and that is why the name. So, start from here, look at the 2 numbers and you can allocate 7 here; once you have done this, this column is consider out, whichever either the row or the column number will be satisfied.

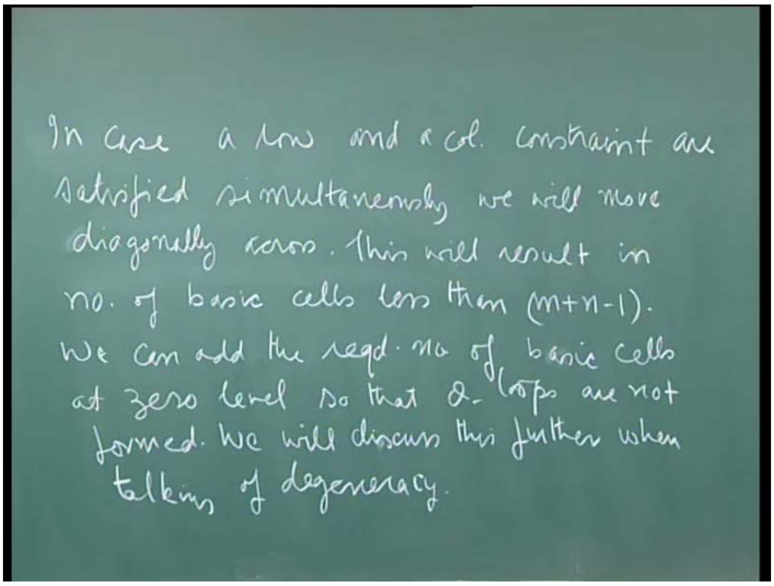
So, you cancel out the corresponding thing, then you have because this is still left over, there is an element 4 units still have to be use from the source, we will move along the row and I will make in a location of 4, because this 4 and 8, this is the minima. So, always check; now, this is 4 here. So, I have to move along the column; move along the column, then 4 and 12; so, I cannot make an allocation of more than 4. So, this way and then this is 8, and so move along the row. When you come here 8 and 11, this is 8 and this becomes 3; this is satisfied. So, I move along this, only this row gets canceled out; then I move down the column, because this is still to be satisfied.

(Refer Slide Time: 40:35)



So, therefore, this will be 3, because 3 and 14, the minimum is 3; so, this is this; so, this gets now canceled out. And I move along the row, this is 11 and 13; so, this is 11. So, this gets satisfied; then, this is 2; to come down the column, this will be 2. And now, you see this is 13 and 13, because the last constraint will always be satisfied automatically. And that is why we drop it from the constraint system and now you have a feasible solution. Why is it a basic feasible solution? Because I am satisfying row or a column constraint at a time; therefore, first of all, this is yes, the solution obtained is a basic feasible solution, why? Because it will have exactly m plus n minus 1 cells.

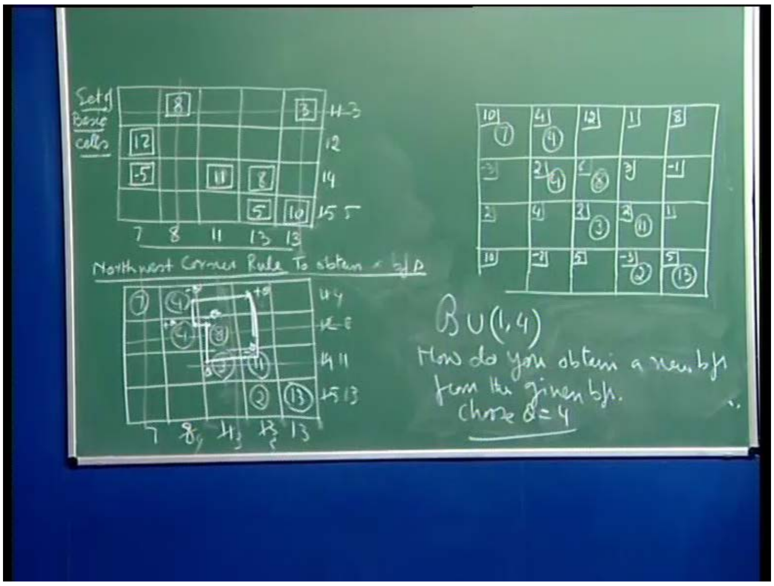
(Refer Slide Time: 42:11)



So, you can do it for the general case; I have shown it to you for a 4 by 5, but this will happen it will have exactly because why? Because a row or a column constraint is satisfied at each basic cell allocation, because the last one will get satisfied automatically; therefore, exactly m plus n minus 1 cells will be allocated. And why will it not have any theta loops? Because I will never go back; see the moment, the way the moment is, and node no theta loops present; you can argued out yourself, note that, so this is the basic feasible solution. Just see how is the life is, in case a row and a column constraint are satisfied simultaneously, this is while discussing the North West corner rule to obtain a starting basic feasible solution.

See if a row and a column constraint are satisfied simultaneously, then we will move diagonally, because I cannot move horizontally or vertically, since both the constraints are satisfied. So, this will result in number of basic cells less than m plus n minus 1, but then we can add the required number of basic cells at 0 level, So that theta loops are not found. So, this can be done, but in the North West corner rule, actually what we do is, we still move either horizontally or vertically and add up basic cell at 0 level, because both the row and the column constraints are satisfied. So, we will just move across horizontally or vertically and put a 0 on a basic cell, so that the number of basic cells remains m plus n minus 1.

(Refer Slide Time: 43:26)



We will discuss this further, when we are talking of degeneracy. So, this gives you a basic feasible solution, and then we said that, I want to show you that, for example, if you add just add this cell, that means, if you want to look at the current, so that means, now look at the union 1 4, look at this 1 4 is the new cell. So, this is going to be a dependent cell.

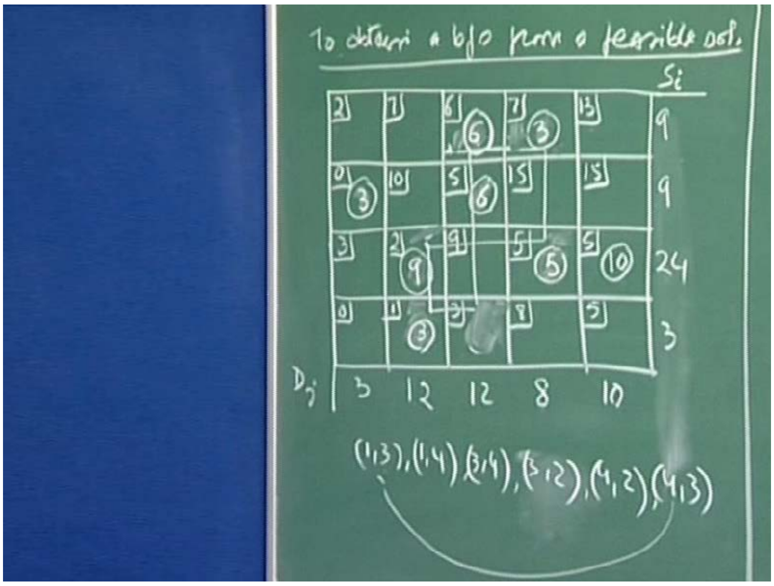
So, how do I discover? How do I locate the theta loop? So, what I will do is, as I told you, the procedure is you start from here; then, you either look at along the row or the column. So, I look along the row; this is the basic cell I hit. So, we will put plus theta here; this will be minus theta. Then, once I come here, I move down the column; so, second column and I come here this is my basic cell. So, this will be plus theta; let me write plus theta here. Then, I move along the row, I come here and this will be minus theta. Then, I moved down the column, this will be this is minus theta, this is plus theta; then, you move here and this will be minus theta. And you see I have in the same column, so I get connected.

So, this is the path of the theta loop, and plus theta minus theta, you can see that this is a set, this is a subset which is present here and here; 1 4 is part of the theta loop, and these two columns, these two rows have to 2 theta cells - theta loop cells; then, this column has it, this row again, this column, this row and then this column.

So, this is how you can locate at theta loop in this. And you see that, once you locate the theta loop, how do you obtain mu basic feasible solution from the given basic feasible solution? So, you can immediately see that, if I want to bring this at a level theta, then this will go down by theta, this will go up, this will go down, go up, go down.

So, therefore, the ones which have going down I have to make sure that they remain non-negative. So, here, theta cannot be more than 4. And similarly, here 8 and 11, so I will choose the minimum of them; that means, if I choose theta equal to 4, this will make what? This will make this cell 0, so that becomes non-basic; this will become a basic cell. And so, you have a new set of basic cells which do not have a theta loop in then, and they give you a new basic feasible solution. So, one has to, just now see, so that means, you have to so many options so which non-basic cell to choose to enter the basis, we will look at the relative prices, and I show you that how easy this to compute the relative prices and then continue with the simplex algorithm for the transportation.

(Refer Slide Time: 47:06)



So, we just continue, and let me show you the how to again locate a theta loop and to obtain; that means, you are given a feasible solution, how to obtain a basic feasible solution from it? So, obviously feasible solution, if it is not a basic feasible solution, it means it has a theta loop. So, we locate a theta loop, remove it and get a basic feasible solution. So, here look at this example; so, these are the supplies, normally I should write here this is S I, and this is your D j's; these numbers give you the cost the C ij's - C ij's for the cells - and the circle numbers give you the values of the variable. So, here is a feasible solution, because 4 5 9 3 6 9, then 11 21 3 24 1 plus 2 is 3. Similarly, all the demand constraints are satisfied; 3 12 12 8 and then 10. So, this is a feasible solution, but it is not a basic feasible solution, because how many positive variables 2 4 7 and 2 9, whereas a basic feasible solution should have 8 basic cells.

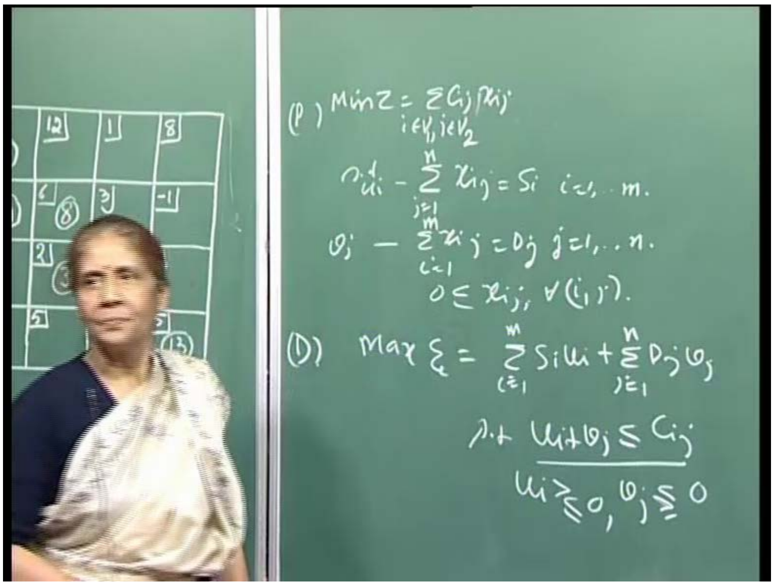
So, therefore theta loop is present. And again, we just start from somewhere that, say I start marking this, so this becomes plus theta; and then, I will go to yes. The thing is that, if I go here, I should because the connectivity is there; so, I should be able to locate a theta loop, and therefore, because I see that here, yes, and that is why the labeling method is needed, when you have large set. So, I will go here; so, this will be minus theta, then I will come here; remember, so this will be plus theta. And then, I can go, because if I go here, then there is nothing else to move up and down; therefore, I need to move here, this will be minus theta. Now, here, again I have to come down, because there is no basic cell upstairs; so this is the 1.

So, this becomes plus theta; then, I can shift here, this. And then, you see, you have this; so, the path that you have, the theta loop that you have is you know consisting of the cells 1 3, 1 4; and then, here this 1 is 1 2 3 4. So, 3 4 and then 4 2. no this is 3 2. See from here, so this one is 3 4 and then so 3 2; then, you come down, 4 2, 4 3, and then, you see your back, because you started with the 1 3, so you are back to 4 3 and then this. So, this will form a theta loop, when you combined here. So, this is the theta loop and plus 5.

So, now we want to get a basic feasible solution from this one. So, this is plus theta, this is minus theta. So, look at the minus theta's, because so here for example, from here, theta can be 5 at most, then you come down here, this is plus, this is minus, theta can be 11. Then, this is plus, this is minus, so theta can be 2; therefore, the smallest value permissible of theta of 2. And so, I will put theta equal to 2, and this basic solution is gone. So, remember here, plus you have to make it 3, because the row and the column constraints have to be satisfied; this is minus, so this becomes 9, this becomes 5, plus then this is minus, so this becomes 3 and that is it. I think we have taken care of so will just make sure that the row and the column constraints are satisfied. This one, this one become 6; so, a check is needed, because you should make sure that you have the 6 plus

3 is 9, 6 6 is 12; this of course was not disturbed, then here this is 14 10, 24, this is 12 and this is 8 and that is it. So, this is now a basic feasible solution that you have obtained.

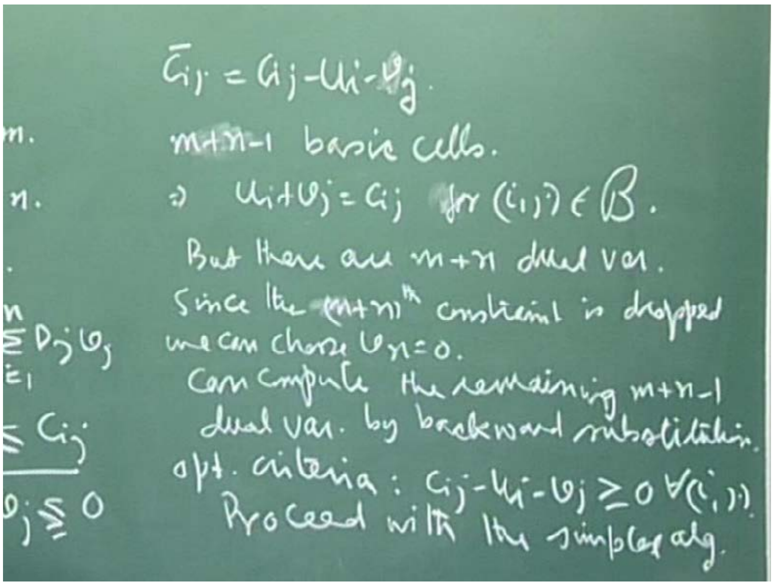
(Refer Slide Time: 52:00)



So, let us now proceed with the method for computing the relative prices. So, we look at the dual problem. Now, I will rewrite your transportation problem primal; primal is minimize Z equal to summation C ij x ij, where i belongs to your V 1 and j belongs to V 2, subject to summation x ij, j varying from 1 to n is equal to S i, i varying from 1 to m; these are your supply constraints, and summation x ij, summation over i varying from 1 to m is D j, j varying from 1 to n. Right now, we are looking at simply this, for all i j; this is your primal problem, and as we said, a every column has only two entries plus 1 or 1 in this case, we are writing plus 1, plus 1.

So, what will be the dual? This will be maximize. So, we will associate with this u i and with this v j, just to distinguish. So, u i is refer to the supply constraints and reaches refer to the demand constraints. So, maximize psi equal to summation S i u i, i varying from 1 to n plus summation d j v j, j varying from 1 to n. And since yours primal constraints are all equality constraints, there will be no restriction on the sign of the dual variables, but it is a minimization problem so subject to yes. Now, you see each column here becomes a row column has only two entries, so they will be 1; and P corresponding to u i and the other corresponding to v j, because look at the column a i j, a i j has entry e i and e j. So, subject to u i plus v j less than or equal to C ij; so, simple structure; no restriction on the signs of so you can see that u i greater less than 0 v j 0, for all i and j.

(Refer Slide Time: 54:41)



So, therefore you see the calculation, remember the dual feasibility implies primal optimality. So, here, the relative prices are simply, so I will not spend time here, because you should be able to figure it out to yourself, that therefore the relative price, that means, your C ij bar is nothing but C ij minus u i minus theta minus v j. So, the relative price for i j is this and so you can simply check.

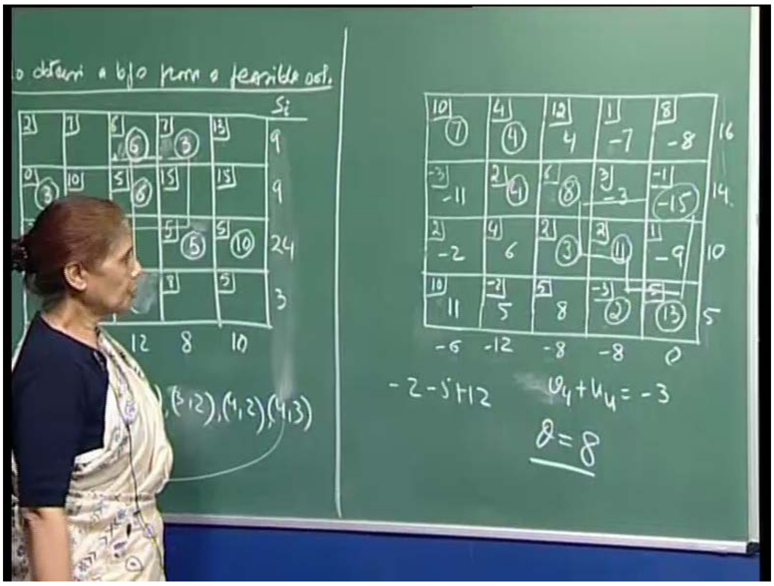
So, but anyway, so first we need to find out the u i's and v j's. So, if I can calculate the u i's and v j's fast, so then no problem; I can then just find out the difference of the right hand side from the left hand side and that gives you the relative price for each non-basic variable. Now, let see, how many equations, because since m plus n minus 1 basic cells, this implies u i plus v j is equal to C ij, for i j belonging to the basic set. So, I will refer to the script B as a basic cell the set of basic cell.

But I have u and m plus n, but there are m plus n dual variables. Now, since the last m plus nth constraint is dropped, we can choose v n to be 0. So, once I choose v n to be 0, because the last constraint is dropped; therefore, if v n is 0, then the product will always be 0. And at any case, it is a balance just seeing if we are working with feasible solution of the time, then the last constraint is anywhere satisfied.

So, actually, it will not really important, that we choose v n a 0, you can choose v n to be any number, because the last constraint will automatically be satisfied. So, then once this, therefore then you will be left with m plus n minus 1, dual variables to compute and you can do that uniquely from these m plus n minus 1 equation. But again, because your distinguish triangular, because your basis is triangular, therefore can compute the remaining m plus n minus 1 dual variables by backward substitution.

So, since v n is 0, you can choose, you can immediately conclude the u i's corresponding to v n, and then because they will be a cells present in the last row. And then you can go on by backward substitution, you can compute the other this thing.

(Refer Slide Time: 59:09)



So, then, once you have the u i's and v j's, you can find out the relative prices, and you can proceed with the, so even the optimality criteria would be, this optimality criteria is that C ij minus u i minus v j or greater than or equal to 0, for all i j. So, if you find a cell this is less than 0, you will be will proceed with the simplex algorithm; and that means, that cell will be ready to enter. And we saw that, if you to the basic set, you add a non-basic cell, it has a theta loop. So, in the theta loop, u plus theta minus theta, you get a new basic feasible solution; again, update your dual solution and so you can proceed with this simplex algorithm. I mean I do not have to spell out the thing, because it is all very clear. So, let us quickly look at the this thing here; so for example, by the northwest corner rule or the same problem, so we have these this is the starting basic feasible solution. And as I told you, if I put the corresponding, so this will be column, for column so v and I am putting it to 0, then immediately you see that, since the sum of these two numbers should be equal to this. So, my new 4 will be 5, is it ok? Once this is 5, then remember, the number here this plus this must add up to minus 3; that means, this should be minus 8, is it ok? Because you see for the basic cells the dual constraints have to satisfied as equality. So, here the corresponding dual equality constraint is, see this is 1 2 3 4, so v 4 plus u 4, this is 4 4 should be equal to minus 3.

So, now I know that u 4 is 5; if u 4 is 5, v 4 will become minus 8. Now, this is minus 8. So, this is backward substitution, minus 8, and remember here, this is 2. So, the number here will be 10, because 10 minus 8 will be 2 with equation has to be satisfied; the inequality has to be satisfied; then, once you have this as 10, this is 2, so I need here this will be minus 8 again, because minus 8 plus minus 2. Once this is minus 8, I go up and here this is 14, because 14 minus 8 is 6 and 6 is equal to 6. Once you determine this, then you come here and this will give you how much? minus 12, because 14 plus minus 12 will be 2. So, this is minus 12, here this is 4; so, this must be 16. And if that is 16, this is 10, so this must be minus 6. And you can immediately see backward substitution method gives you easy way to compute the dual solution. Once you have the dual solution, I can now compute the relative prices for the non-basic cells which is again simple, because it has to be 12 minus, so add up these to numbers this is 8; so, 12 minus 8 is 4; yes, please sit down with the paper and a pencil and you can do the work out the thing.

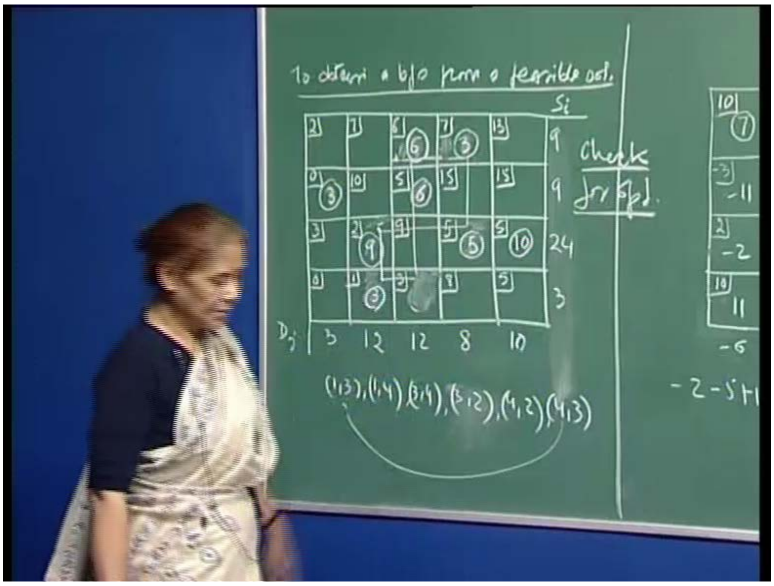
So, now here this is 1 minus, this is the 8, 1 minus 8 is minus 7; so, this is negative. Then, this is 8 minus 16 minus 8, 8 minus 16, this is 0. Similarly, come here and this will be minus 3. And then, that is 8, so minus 8 minus 11. Then, this will be 3 minus 6, 3 minus 6 will be minus 3; and this one is minus 1, so 14 minus 15. And here, this will be 2 minus 4, 2 minus 4 will be minus 2; yes please check my calculation also, 4 minus 2, minus minus 2, so this is 6; then this will be 1 minus 10 minus 9. And here, this one will be 10 minus 1, so 11. And this is minus 7, minus 7 so minus 2 plus 7, so this become 5; this is minus 2 and then you have minus u I, which is 5 plus 12, 7, 5 that is right.

So, then this will be 5 minus of 3, so which will become 8. So, this is so this is your relative prices; we will go for the most negative one and later on will have a discussion, why? Therefore, let us say, I go for this one, so that means, you have to scan and find out which is the most negative one, but remember the Bland's rule said that, you can choose the first negative one. So, I do not have to compute. For large problems, we will have to do something like this, because for example, when I hit this one, I should have just stop here and this added to bring this into the basis, so anyway.

So, now you see that, you have to form a theta loop, that means, this is going to enter the basis, so this will be this, yes up and downs; so, horizontal, the vertical, horizontal, vertical. So, here, theta, if you put this plus, theta will be minus theta, plus, minus, plus, minus, so that means, theta can be equal to 8, because this is minus, plus, minus, plus, minus; so, the minus ones required you to be it is 8 11 and 13. So, the best value that you can choose for theta possible this theta equal to 8; that means, now you will have a new basic feasible solution, and actually you can also see, once you get a little feeling for the algorithm, is that you see these cells do not get disturb.

So, I just need to update part of the dual solution; I do not have to even update the dual solution. And then, you can see that you will also not need to update all the relative prices; you can just that will come by experience, but anyway from large problems, you need there are good programs, that in for the transportation problem. But you see the simplification that comes through the because of the special structure; the transportation problem has a very special structure.

(Refer Slide Time: 65:11)



So, you do not need so very different computations. And you can just see how the things are progressing; therefore, you get the next basic feasible solution and then you have to check for optimality. So, here also, I would like you to see that this problem, for example, if you compute the dual variables, then this probably will done out to be an optimal solution; that means, now this check for optimality for this.

So, what I wanted to do is, after you have updated this, check for optimality. So, I want you to work it out yourself, obtain the dual solution and then you will see that all relative prices will be non-negative. So, I think we will stop here and then continue with the other aspects of the transportation problem.