**Fundamentals of Operations Research**

**Prof. G. Srinivasan**

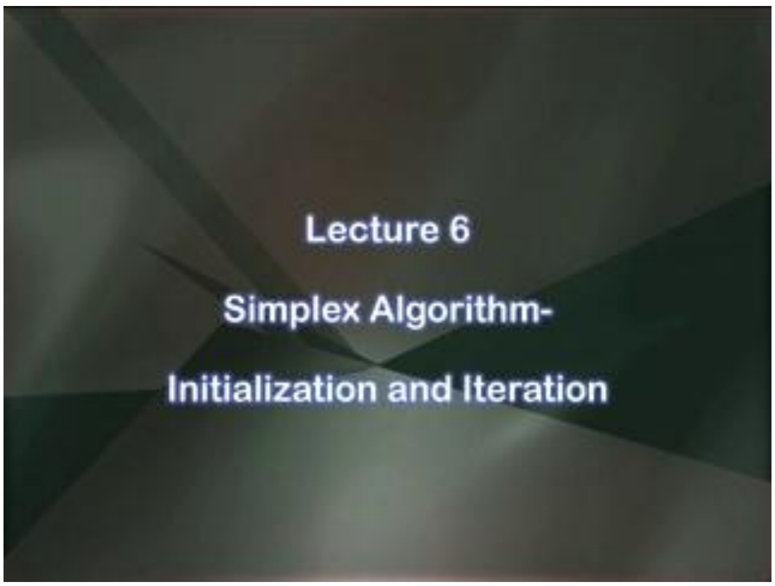
**Department of Management Studies**

**Indian Institute of Technology Madras**

**Lecture No # 06**

# Simplex Algorithm Initialization and Iteration

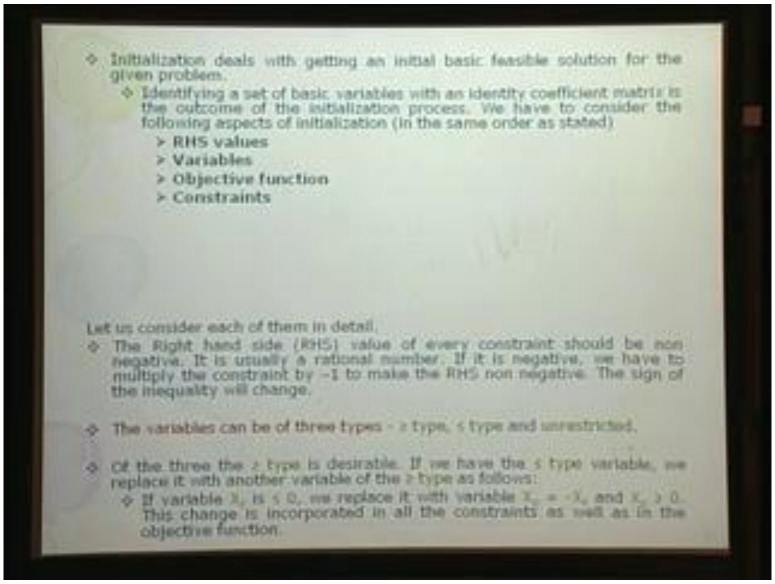
(Refer Slide Time: 01:10)




In the last lecture we solved a minimization problem using the simplex algorithm and we introduced artificial variables when they were necessary. We also indicated the greater than or equal to constraint because it has and gives a negative slack. We will try to introduce an artificial variable in the simplex algorithm and we also said that we have to reduce the number of artificial variables introduced in the problem because they do not exist in the problem.

There are some other aspects of Initialization that we will see in this lecture will then go on to look at some aspects of Iteration and Termination with respect to the simplex algorithm.
Now what are the various aspects of Initialization that we have to see?

(Refer Slide Time: 02:06)



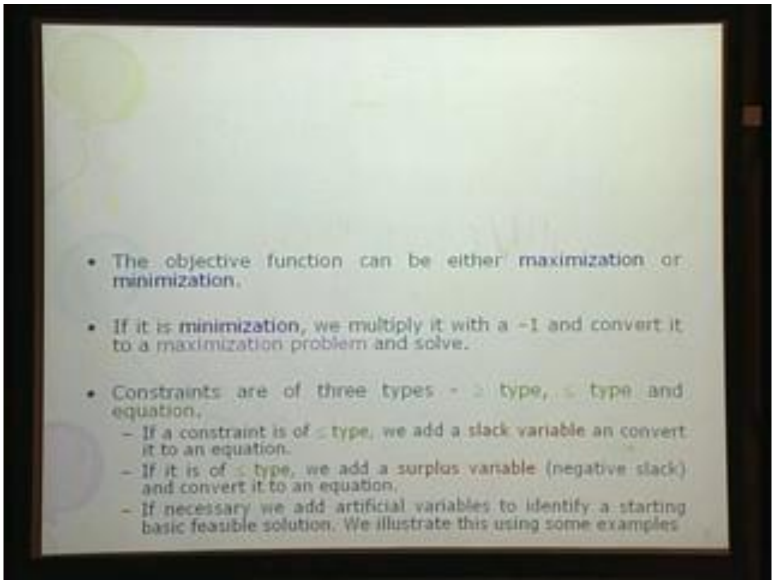

One is, we have to look at the right hand side values. We need to look at the variables objective function and constraints. Let us consider each one of them. Right hand side value of every constraint should be non negative and we have already seen this. It is actually a rational number. If it is negative, we have to multiply the constraint by a minus one to make the right hand side non negative. The sign of the inequality will change. If there is a less than or equal to it will become greater than or equal to and vice versa. That is the only thing we need to look at.

What we need to ensure is when we start this simplex table we have to make sure that the every constraint has a non negative value in the right hand side. The aspects we need to look at are the variables. Now variables can be of three types. As we have already seen, we could have greater than or equal to type less than or equal to variable and an unrestricted variable.

We are seeing formulations that involve greater than or equal to most of the problems or for example, will have greater than or equal to variables. Very few instances we might have an unrestricted variable. We have seen one formulation which had an unrestricted variable. Less than or equal to is very rare and we include it only for the sake of completion.

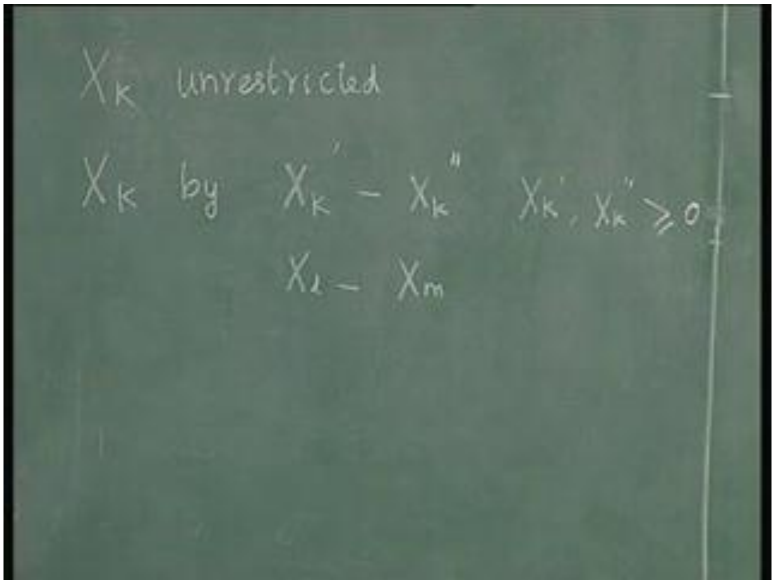
Now we also know that when we start this simplex Iteration all variables have to be converted to the greater than or equal to type. We cannot have unrestricted or less than or equal to when we start this simplex Iteration. So what we do there is of the 3, the greater than or equal to is desirable. If we have the less than or equal to type variable we replace it with another variable of the greater than or equal to as follows. If a particular $X_k$ is less than or equal to 0 then we replace it with another variable $X_p$ equal to $– X_k$ and $X_p$ will become greater than or equal to 0. This change is incorporated in all the constraints as well as in the objective function.

(Refer Slide Time: 04:01)




Therefore it is very easy to handle if variable is less than or equal to type. Objective function can be either maximization or minimization. There is also one more thing that needs to be checked before we do is, what happens if the variable is unrestricted in sign. Go back to the previous slide; the variable can also be unrestricted in sign. So when variable is unrestricted in sign what we do is the following.

(Refer Slide Time: 04:39)




If we have a particular $X_k$ unrestricted, we replace $X_k$ by 2 variables. You can call them $X_k$ dash – $X_k$ double dash or it could be some $X_1 – X_m$. We replace every unrestricted variable with two variables. $X_1– X_m$ or $X_k$ dash – $X_k$ double dash and incorporate this change in the constraint as well as in the objective function. With the additional condition that $X_k$ dash and $X_k$ double dash both are greater than or equal to 0. Now if this unrestricted variable is in the solution and takes a positive value then this will be in the solution and takes a positive value. If this is in the solution and has a negative value for the given problem then this one will be in simplex table and this will take a positive value so that the net will be this negative.

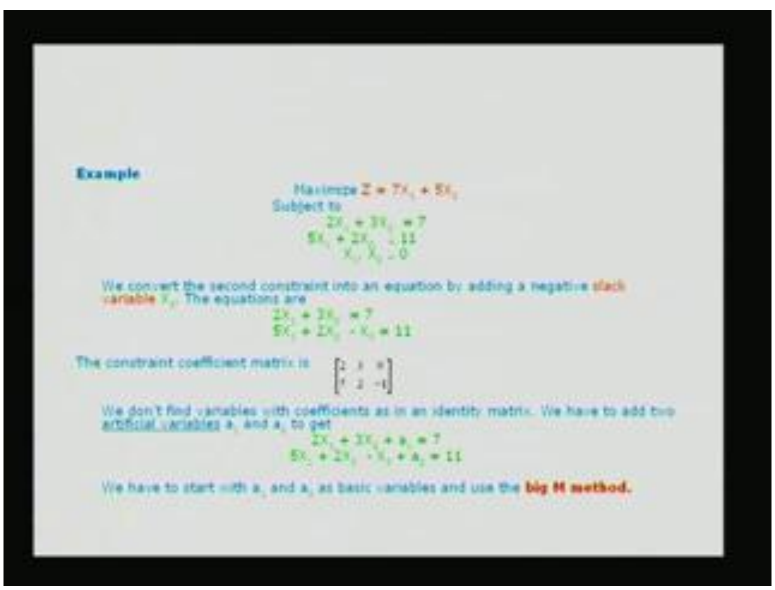
If this is not in the solution of the original problem then both will not appear as basic variables in the simplex. We will see this later through an example. Now let us now go back to the two types of objective functions. Objective functions can be either a maximization function or a minimization function. As far as we have solved problems of both, we have solved maximization problems as well as one minimization problem. The standard thing to do is to convert the minimization into maximization/multiplying the objective function with a – one, convert it to a maximization problem and solve it, so that consistently we can use the largest $C_j – Z_j$ as rule to identify the entering variable. Constraints are of 3 types, greater than or equal to type, constraints less than or equal to type and equation.

If a constraint is of the less than or equal to type, this is assuming that the right hand side values are non negative or after converting them to non negative values. If a constraint is of the less than or equal to type we add a slack variable and convert it to an equation. Slack variable is a positive slack variable. If the constraint is of the greater than or equal to type then we add a surplus variable or a negative slack variable and convert it into an equation.

(Refer Slide Time: 07:07)


For example if we have a constraint, $X_1 + X_2$ greater than or equal to 7, then we write $X_1 + X_2 - X_3 = 7$. This $- X_3$ is a negative slack variable. This $- X_3$ being a negative slack cannot qualify to be an initial basic variable therefore we may have to add artificial variables. So if necessary we add artificial variables to identify a starting basic feasible solution. We illustrate this through several examples.

(Refer Slide Time: 07:42)

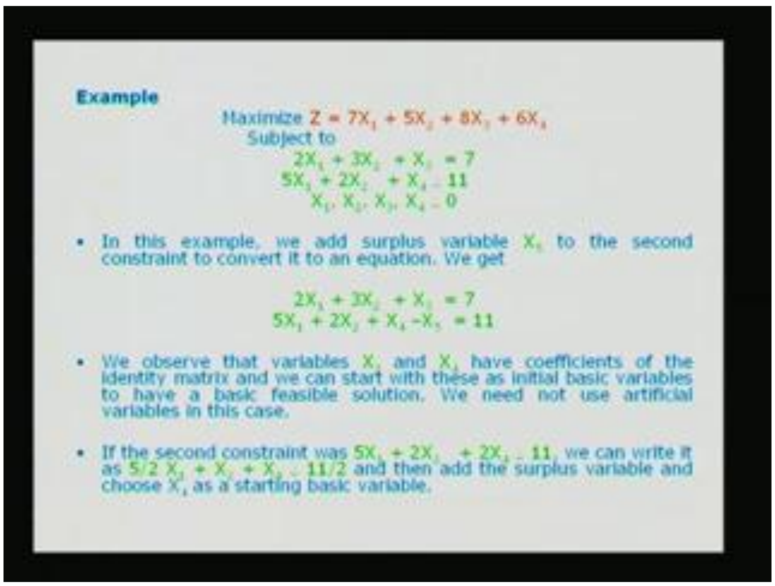



Let us take the first example which is like this. Now the first constraint is an equation, second constraint is greater than or equal to type inequality. Already the right hand side values are non negative; the variables are greater than or equal to 0, the objective function has maximization.

We do not need to do any of the conversion to the standard form that we are comfortable with. Now we have to look only at the two constraints $2X_1 + 3X_2$ is already an equation = 7; $5X_1 + 2X_2$ is greater than or equal to 11. Now this greater than or equal to inequality is now converted to an equation by adding a negative slack. So we have $5X_1 + 2X_2 - X_3 = 9$. We write the matrix form (2 5 3 2 0) – 1. Corresponding to variable $X_1$ you have (2, 5). Corresponding to variable $X_2$ we have 3 and – 2. Corresponding to variable $X_3$, we have 0 and – 1. But we do not find any identity column there. We do not find any variable which can qualify to be a basic variable. In this case we need to add two artificial variables $a_1$ and $a_2$.

First one will become $2X_1 + 3X_2 + a_1 = 7$ and the second will become $5X_1 + 2X_2 - X_3 + a_2 = 11$ and then we start the simplex table with $a_1$ and $a_2$ as basic variables and you can use either the two phase method or the big M method to solve this problem. So this is the case where we have an equation and an inequality. We need to introduce two artificial variables.

(Refer Slide Time: 09:23)



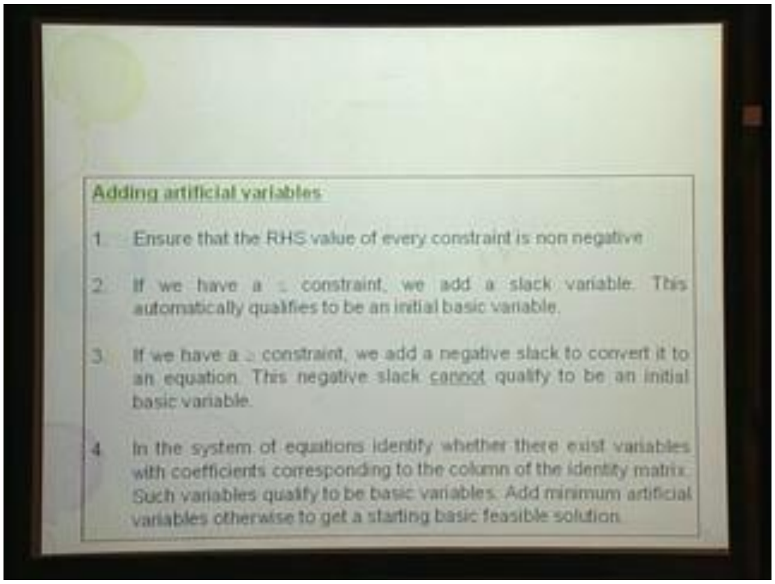

Let us go to another example. Let us take this example which again has an equation and an inequality. Once again the right hand side values are non negative and the variables are all greater than or equal to 0. The first constraint is an equation so we do not do anything. Second constraint is an inequality which is of the greater than or equal to type so we add a negative slack and convert it to an equation to get $2X_1 + 3X_2 + X_3 = 7$ ; $5X_1 + 2X_2 + X_4 - X_5 = 11$.

Now we realize that the variable $X_3$ has a +1 coefficient and a (1, 0). It does not figure in the second constraint. Variable $X_4$ which is a part of the problem has a +1 coefficient does not figure in the first constraint so it has a (0 1). Now $X_3$ and $X_4$ becomes eligible to be the initial basic variables. So this is an example where we do not have to add an artificial variable at all even though we add an equation and we add a greater than or equal to. It is not that we blindly add an artificial variable whenever we have an equation or greater than or equal to. The only thing we need to do is convert it to a set of equations. If basic variables exist we use them, if they don't exist, we add.

We observe that $X_3$ and $X_4$ in this example have coefficients of the identity matrix and we can start with these as initial basic variables. We need to use artificial variables at all in this case. You can also do a few things for example in the second one, $5X_1 + 2X_2 + 2X_4$ is greater than equal to 11. We can still do this and divide by 2 and make this $X_4$ and now $X_4$ will have a (0 1) and we can start.

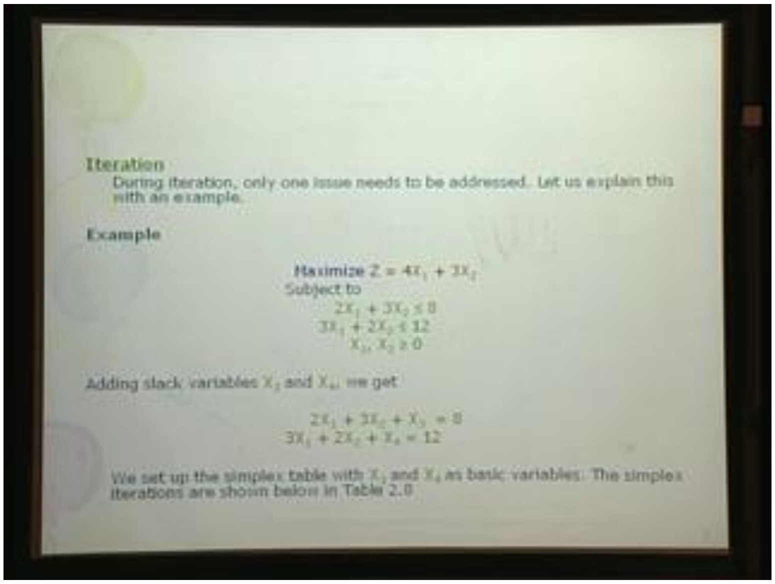
So the message that we are trying to convey through these examples is that it is not absolutely necessary to add an artificial variable for every greater than or equal to while it is absolutely necessary to add a negative slack to convert it to an equation. All we need to do is given the constraints; we first add these slack variables and bring it to an equation form. If we can identify basic variables, from there we start the simplex Iteration using the basic variable. If we are not able to identify, only then we add artificial variables. In the process we minimize the number of artificial variables added to a problem.

(Refer Slide Time: 11:56)



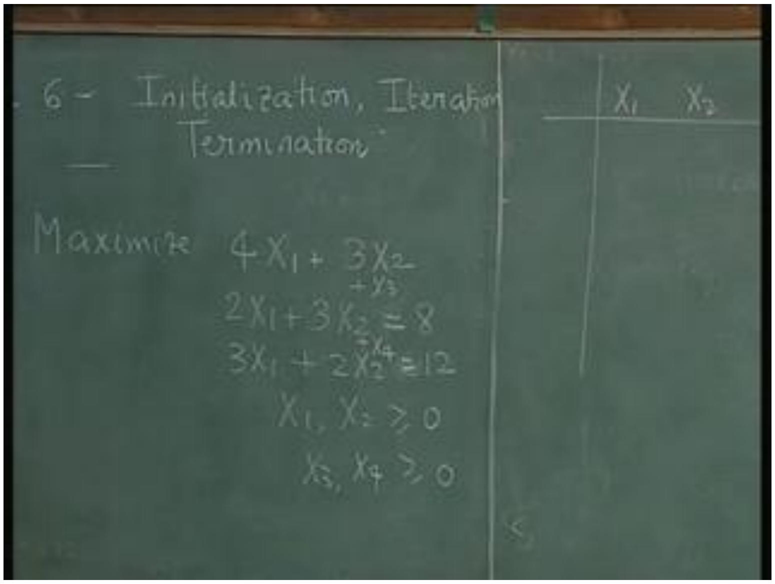

Now these are the rules about artificial variables. We ensure that the right hand side is non negative. If we have a less than equal to we add slack variable which automatically qualifies to be a basic variable. If we have a greater than or equal to we add a negative slack to convert it to an equation. A negative slack cannot qualify to be an initial basic variable. The system of equations whether there exist variables with coefficients corresponding to the column of the identity matrix qualifies to be basic variables. Others add minimum number of artificial variables otherwise, to get a starting basic feasible solution. Fewer the artificial variables the better it is.

(Refer Slide Time: 12:35)



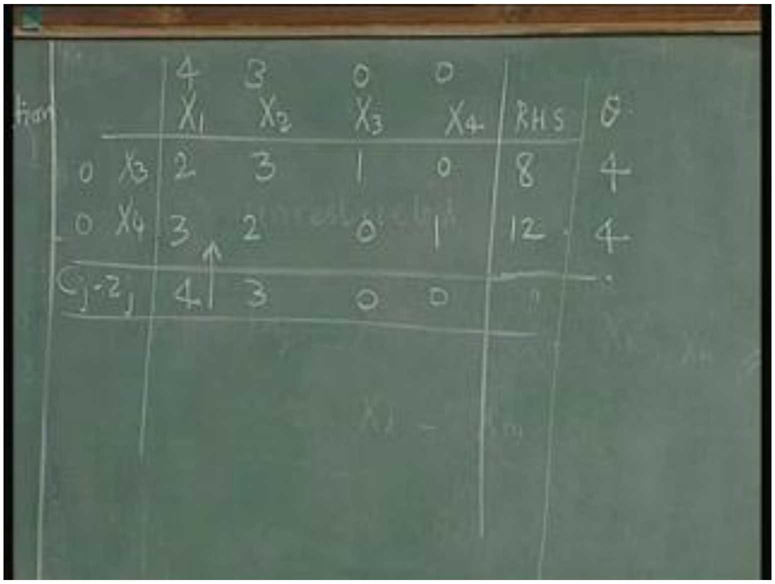

The next aspect is we need to look at is called Iteration. Can something happen during the Iteration of this simplex algorithm? Now we try to understand that using a different example here.

(Refer Slide Time: 12:59)




The example is to maximize $4X_1 + 3X_2$ subject to $2X_1 + 3X_2$ less than or equal to 8; $3X_1 + 2X_2$ less than or equal to 12; $X_1$ and $X_2$ greater than or equal to 0. So we have less than or equal to constraints and hence we add slack variables.

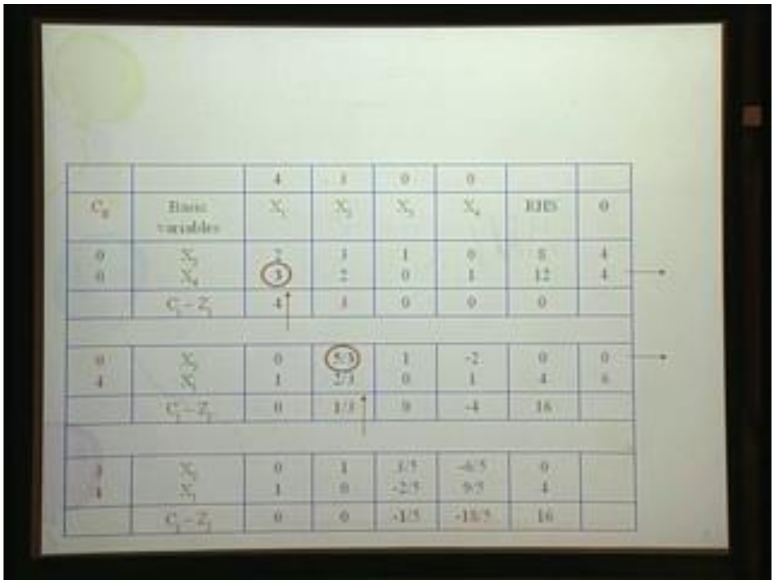
(Refer Slide Time: 13:30)




Slack variables are automatically qualified to be basic variables. So we have $X_{1,}$ $X_{2,}$ $X_3$ and $X_4$ which are the slack variables. So I would get + $X_3 =$ 8, + $X_4$ =12; $X_3$ and $X_4$ greater than or equal to 0. $X_3$ and $X_4$ qualify to be starting basic variable. So we start with $X_3$ and $X_4$ here and have 4 3 0 0 2 3 1 0 8 3 2 0 1 = 12 0 0 4 3 0 0 Now variable $X_{1,}$ with largest positive $C_j – Z_j$ enters. To find out the leaving variable we need to compute theta. Theta is 8/2 = 4; 12/3 = 4; so we find something new, something interesting that we are not able to find a unique leaving variable. So far in all our examples we could easily find a minimum theta.

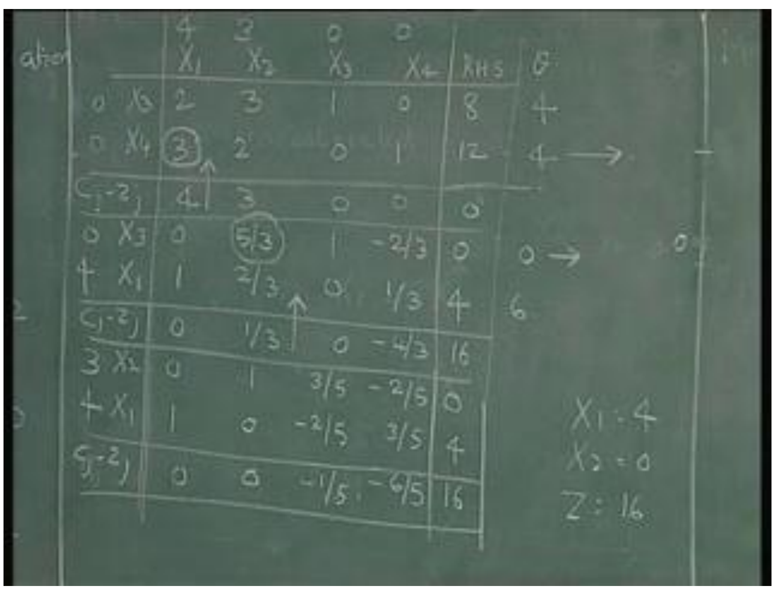
There was no tie for the minimum theta. Now there is a tie for to the minimum theta that we have. We need to break this tie by leaving out $X_3$ or $X_4$. We could do either. Let us assume there is no tie breaking rule so we either leave out $X_3$ or $X_{4.}$ Let us go to the next type now.

(Refer Slide Time: 15:25)



Let us look at what happens if instead of $X_3$ we leave out $X_4$. We could either choose to leave $X_3$ or choose to leave $X_4$. It is assumed that we want to leave $X_4$ instead of $X_3$ and let us continue our Iteration. So, variable $X_4$ leaves the basis.

(Refer Slide Time: 15:44)




Now when variables $X_4$ leave the basis, this is that pivot element, so you have $X_3$ here and $X_1$ here. We have 0, we have 4 divided by the pivot element (1 2/3 0 1/3 4). I need to get a 0 here (Refer Slide Time: 16:24). This row – 2 times this would give a 0. So 2 – 2 into 1 = 0; 3 – 4/3 is 5/3; 1; (0 – 2) into 1/3 is = – 2/3; 8 – (2 into 4) is = 0. This is also something which we are encountering for the first time. For the first time, a right hand side value has become 0. So let us continue. We have $C_j - Z_j$; $X_3$ and $X_1$ being basic variables will have 0, this is 3, 3 – 8/3 is = 1/3; 0 – 4/3 will give me – 4/31 into 4 = 16. Now variable $X_2$ enters the basis.

We need to find out theta. We already have a 0 here and we have a positive number here. This theta is 0 and we still compute this theta. Remember again we would leave out theta only when this number is 0 or negative. As long as this number is positive, even if this is 0 we compute theta which is 0. So 4 divided by 2/3 will be 6 and this is minimum theta which goes. This is the pivot element that we have. So now we have $X_{2,}$ $X_{1.}$ Now this is 3, this is 4 now.

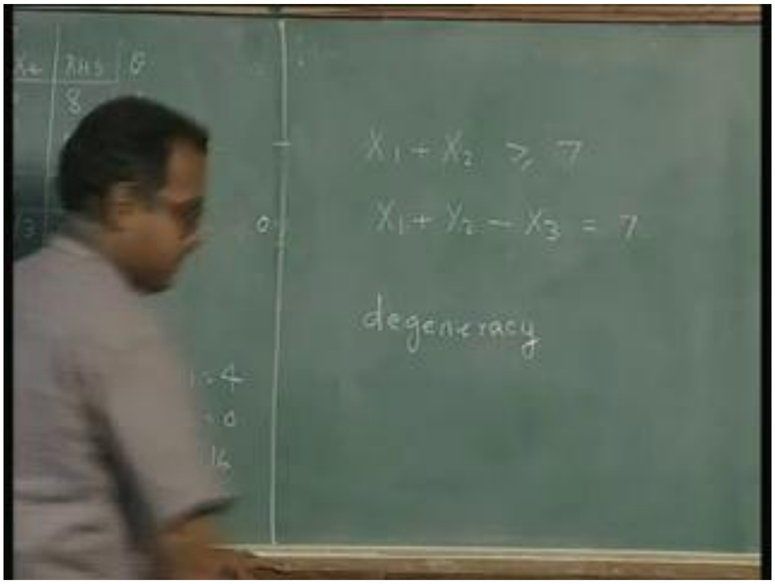
Dividing by the pivot element which is (5/3 0 1 3/5 – 2/5 0). Now I need a 0 here so – 2/3 times this (Refer Slide Time: 17:50) will give 0. So this row – 2/3 times this row, (1 0) 0 – 2/3) into 3/5 is = – 2/5; 1/3 – 2/3 into – 2/5 so 1/3 + 4/15 which is 9/15 which is 3/5; 4 – (2/3 times 0) which is = 4. $X_1$ and $X_2$ are the basics. So I have 0. This is 4 into – 2/5 = 8/5 so I have a + 8/5 and this will be – 12/5 and this is still 16. Now this variable enters $X_{3.}$ This is – 1/5.
So this is 9/5; 8/5 is = –1/5. So I get a – 1/5 here. This is – 6/5 + 12/5 which is 6/5 so I get – 6/5. Now all this $C_j - Z_j$ are negative. The algorithm terminates. The optimum solution is $X_1 = 4X_2 = 0$; Z =16.

What are the new things that we have seen in this? The first thing that happened was there is a tie for a leaving variable. We resolved the tie arbitrarily. We could have left out variable $X_3$ but then we decided to leave out variable $X_4$.
Now because we have a tie here, we also had a 0 here, the 0 came only because we had a tie here. More importantly this problem actually has taken 3 Iterations to terminate with the objective of function value 16 but right in the second Iteration we have a solution with objective function value. 16 solution with $X_1$= 4; $X_3$ = 0; $X_2$ = 0; $X_4$ = 0 are non basic. Here we have $X_1$ = 4; $X_2$ = 0; $X_3$ = 0; $X_4$ = 0 are non basic.

Here we have the same solution which is $X_1$ = 4; rest of them 0 but the algorithm was unable to terminate here. It was still indicating that variable $X_2$ can enter the basis. Whereas only here for the combinations $X_{1,}$ $X_{2,}$ $X_3$ and $X_4$ are non basic. For the same solution the algorithm now is able to terminate. The algorithm is not smart enough to understand that right here the optimum solution has been reached. It is still within the boundary or capability of simplex, because simplex would evaluate the non decreasing solutions. It started with 0, it went up to 16.

It did not increase, it did not go down. It didn’t evaluate a poorer solution than this but it evaluated a solution which is the same. It repeated it once again. Now in this phenomenon you end up doing more Iterations than what you actually have to which is happened in this case. If the algorithm has been good enough or smart enough it should have identified that the optimum is right here and it need not have carried out one more Iterations to get the same solution and then satisfy the termination condition.

(Refer Slide Time: 23:12)




This phenomenon where we do additional Iterations without realizing that optimal has been reached or without moving forward without any improvement and then terminating is called degeneracy. Degeneracy is a phenomenon that happens in simplex Iterations where we end up doing additional Iterations which do not add value to the problem or problem solution.

Degeneracy always happens if there is a tie for the leaving variable. A tie for a leaving variable would ensure that one basic variable takes 0 in the next Iteration and the moment a basic variable takes 0 in the Iteration, what will happen is this will always be a candidate to leave because a minimum theta will be 0 and because this is the candidate that, leaves the solution will not improve. In fact you can show that, in every simplex Iteration, the improvement in the solution, improvement in the objective of function, is actually the product of the minimum theta and this $C_j – Z_j$.

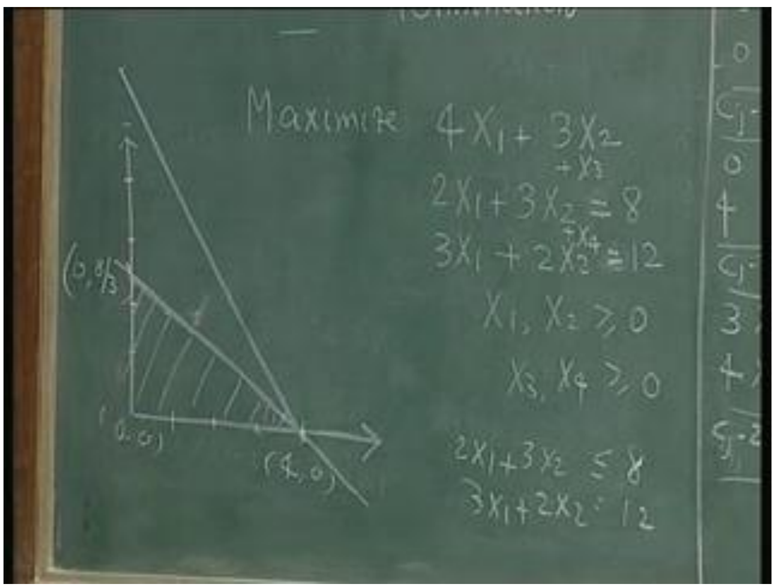
In this case the minimum theta is 4 either way we have taken. $C_j – Z_j$ is 4 entering $C_j – Z_j$ so the product is 16 and so 0 becomes 0 + 16 which is 16. In this case because of the presence of this 0 and minimum theta being 0 no matter what your $C_j – Z_j$ is, the solution does not exist because of a tie in leaving variable and because of which you got a 0 in one of the basic variables, simplex cannot move forward. You end up doing extra Iterations and before you terminate it.

We will go through in the set of slides where we show that this kind of phenomenon degeneracy is actually in this example, it had happened at the optimum. Degeneracy can happen inside during an intermediate Iteration and it can still terminate. In fact what will happen is suppose for a different problem the optimum is not 16, somewhere in the middle, we have got into a degenerate situation where there is a tie and you will end up doing Iterations without improving the objective function because of this 0. Now if you have to improve and then come to the optimum somewhere in one of the Iterations, you will find that for a 0 here the entering variable would have a negative. So you will not be evaluating this theta. We will be evaluating minimum theta which will now become positive. If it turns out somewhere in middle of some iteration, the entering column will have a negative or a 0 corresponding to the 0 here.

So this theta will not be evaluated. Now your minimum theta will become positive $C_i - Z_j$ will be positive. Product will be positive and it will move further. The one thing about degeneracy is, it is a phenomenon that happens during the Iteration. If the problem has an optimum the problem will terminate. Degeneracy will not prevent the problem from terminating. Degeneracy will only play its part by going through unnecessary Iterations which do not add value or which don't improve the objector function.

Now let us see why this degeneracy has happened. In this example we have a convenient 2/2 so you could go back and rather graph and see what happens to the graphical representation of this problem. So let us do that.

(Refer Slide Time: 26:46)




Let us draw the graph here. The first constraint is $2X_1 + 3X_2$ is less than or equal to 8.
So we would have (4, 0) as 1 point and you could have (1, 2) as another point. So this is another point this is see you could have (1, 2) as another point. So this is the first line. Second line is $3X_1 + 2X_2 = 12$; so you again have (4, 0) and (0, 6). So this is the feasible reason. It satisfies both the constraints. Now in a normal 2/2 problem you would expect four corner points.

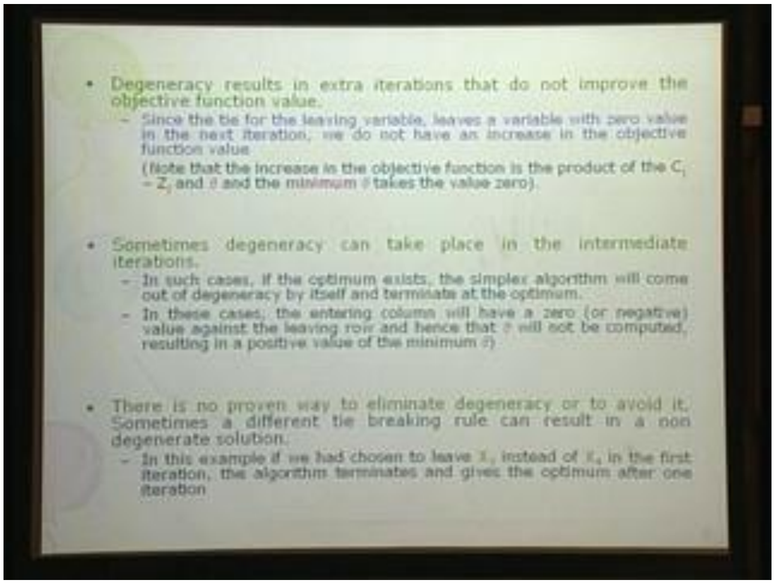
Now you have three corner points (0, 0), (4, 0) and (0, 8/3). Now you can also assume that there is a fourth corner point which is actually sitting on top of the same point. Now this point is not only the intersection of these two. It is also the intersection of this (Refer Slide Time: 27:46) and the X axis as well as this on the X axis. There is basically more than one point sitting on the other. What simplex would do is, it started with (0 0). It then identified a basis and this was $X_3$ and $X_1$. It moved to this point but it was not able to say that it is optimum.

So it tries to move to another point which is sitting on top of it which is $X_1$, $X_2$ (this point) and once it reaches this basic combination, it is able to identify. The point is simplex has its weakness. If the problem exhibits degeneracy then simplex is unable to say that every point here is optimum. It waits. It searches till it gets the correct basis for which it can say that it is optimum. So this phenomenon is called degeneracy.

Now degeneracy does not prevent you from getting a solution. If the problem has a solution then simplex will terminate. The only thing is if it gets into a degeneration situation it will go through extra Iterations. Now is there a way to overcome degeneracy? The answer is actually there is no way to overcome degeneracy. One thing that we can do is, many times it is said that if there is a tie for leaving variable consistently, chose the one with a smallest subscript. Smallest subscript rule works well. It will ensure termination but it will not reduce or minimize the additional Iterations that you will have to work and in any case if the problem has a solution even if it exhibits degeneracy it will terminate. So degeneracy from a certain point of view is harmless. It is not going to prevent us from getting the solution. The only difficulty with degeneracy is it will end up making more Iterations then what we actually need. Degeneracy is a limitation of the simplex algorithm.

It is not a characteristic - it is of course a characteristic of the problem. For example if we had used the graphical method or an algebraic method, we would easily have got the solution. We would any way get this solution using simplex except that we need to do one or more Iteration or many more Iterations. As I said degeneracy can happen in the middle of the algorithm and if the problem persists still, then as a solution it will come out of degeneracy. As I said earlier by encountering a situation where against a 0 right hand side, we would have negative value or a 0 for entering column. So theta will not be 0. Theta will become a positive value and then you will move towards the optimum solution. So this is the only thing that can happen during this simplex Iteration and that is the only aspect we need to touch.

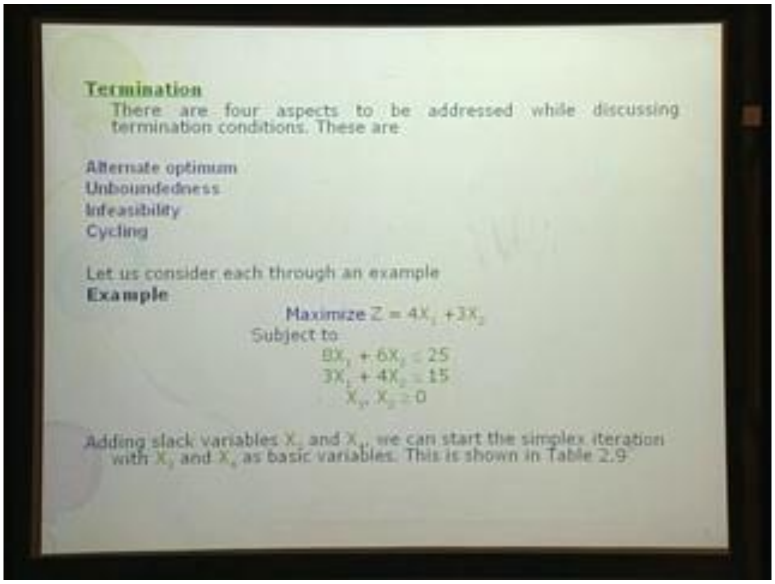
(Refer Slide Time: 31:10)




Now degeneracy results the extra iterations that do not improve the objective function value. The reason is that there is a tie. So the next Iteration has a 0 etc. Sometimes degeneracy can take place in the intermediate stage of the algorithm. In such cases if the optimum exists simplex will overcome by itself and terminate at the optimum. In these cases entering column will have a 0 or a negative value against the leaving row.

There is nothing proven to eliminate degeneracy or to avoid it. Sometimes a different tie breaking rule here would have given us the optimum in less than one iteration. So it can happen. In this example if we had chosen to leave $X_3$ instead of $X_4$ in the first Iteration, the algorithm terminates and gives the optimum after one iteration. You can take it as an exercise and see whether such a thing is happens. Then we will be able to solve to get a optimum two iterations instead of three as in this example.

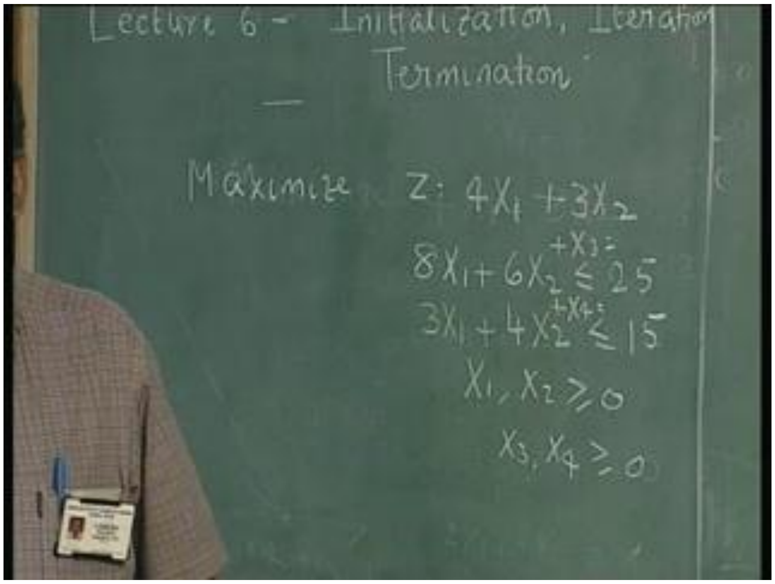
(Refer Slide Time: 32:04)




Now we move to the last part of the simplex which is called as the Termination. We have seen some aspects of Initialization and Iteration. We now see various aspects of termination.

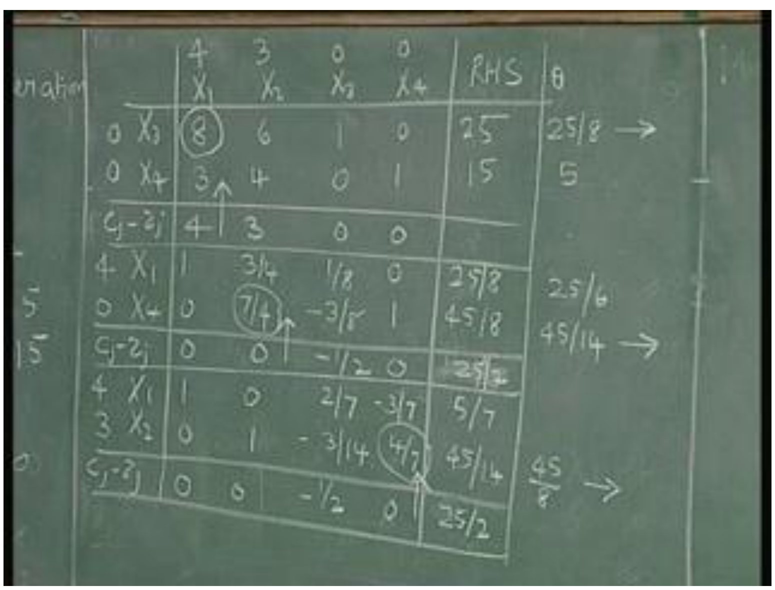
Now there are many issues that need to be looked at in terminations. We have listed all of them here. There are basically four issues that we need to look at. These four are called Alternate optimum unboundedness infeasibility and Cycling. We will concentrate on the first three; we will only briefly indicate the effect of cycling and its role. We will not do cycling through an example in this set of lectures. So we take examples. Show each one of them the alternate optimum the unboundedness and infeasibility.

(Refer Slide Time: 32:54)




Maximize $4X_1$ + $3X_2$; $8X_1$ + $6X_2$ less than or equal to 25; $3X_1$+ $4X_2$ less than or equal to 15 greater than or equal to 0. Once again we have a problem with right hand sides positive so we do not need to do anything here. We have less than equal to type constraints so we had slack variables to get + $X_3$ = 25; + $X_4$ = 15; $X_3$, $X_4$ greater than or equal to 0 and we can start the simplex Iteration with $X_3$ and $X_4$ as basic variables.

(Refer Slide Time: 34:06)




Let us do that 1 0 (25) 3 4 0 1 (15). We write this $C_j - Z_j$; $X_3$ and $X_4$ being basic variables will have $C_j - Z_j$ is = 0 and we will have 4 – (0 into 8) + 0 into 3 is = 4 and 3. Now variable $X_1$ with the largest positive $C_j - Z_j$ enters the basis.

So to find out the leaving variables we need to compute the theta so 25/8 and 15/3 which is 5 the smaller one is 25/8 so this leaves the basis. This is the pivot element that we have. So now $X_1$ replaces $X_3$. We have $X_1$ and $X_4$. We have a 4 here and a 0 here once again this is the pivot element. So we divide every element of the pivot row by the pivot element to get (6/8, 3/4, 1/8, 0, 25/8). We need a 0 here so this row – 3 times this row would give us a 0 so 3 (– 1 into 3) is 0; 4 – 3 into 3/4 = 4 – 9/4 is 7/4.

0 – 3 into 1/8 is = – 3/8; 1 – 3 into 0 is 1; 15 (– 3 into 25/8) = 15 – 75/8; 120 – 75 is 45 so we get 45/8. $C_j – Z_j$, $X_1$ and $X_4$ are basic variables so you get 0 and 0. Now this is 4 into 3/4 is = 3.
0 into 7/4 is = 0 so 3 – 3 = 0; 4 into 1/8, this is 1/2; 0 into – 3/8 is 0; 0 – 1/2 is = – 1/2.
100/25, 25/ 8 into 4 + 0 into 48 is 100/8. Now let us look at this solution. First let us check whether this solution is optimum. Look at the $C_j – Z_j$ now, the non basic variables are $X_1$, $X_2$ and $X_3$. Both of them do not have a positive $C_j – Z_j$. So there is no entering variable. The algorithm terminates as we know. The only difference in this case is there is a non basic variable $X_2$which has a 0. So far in all our examples when the algorithm terminated all the non basic variables had negative values of $C_j – Z_{j.}$ So we were very sure that entering this negative value cannot improve or increase the objective function. Remember we are consistently solving a maximization problem. Now a 0 here can do something. The question now is do we enter this variable $X_2$ which has a 0 $C_j – Z_j$ and what happens when we do so?

Now when we enter this we are very sure that if we enter a $C_j – Z_j$ with the negative it will bring down the value of the objective function because the improvement is the product of $C_j – Z_j$ and theta. Theta can never be negative so if we enter a negative $C_j – Z_j$, the value is going to come down. When we enter a positive $C_j – Z_j$ the value will go up assuming that theta is also positive. When you have a 0($C_j – Z_j$), the question is do I enter this?

Let us see what happens when we enter this. Let us enter variable $X_2$ with a 0($C_j$ – $Z_j$) corresponding to the values of theta would be 25/8 divided by 3/4 which should be 25/6, 25/8 into 4/3 would give us = 25/6.
45/8 divided by 7/4 will give us 45/14; and 45/14 is smaller than 25/6. So this variable leaves.
Now this is the pivot element. We continue with the simplex Iteration. So we have $X_1$and $X_2$ with 4 and 3 divided by the pivot row, every element of the pivot row by the pivot element to get 0 1 – 3/8 into 4/7 is – 12/56, which is – 6/28 which is – 3/14. This is 4/7 and this is 45/14.
Now we need a 0 here so this – 3/4 times this would give us a 0
So 1 – 3/4 into 0 as 1; 3/4 – 3/4 into 1 is 0.
1/8 – 3/4 into 3/14
So 1/8 + 9/56; 16/56; 16/56 is = 8/28; 4/14 and 2/7, this is – 3/4 into this.
So 0 – 3/4 into 4/7 is – 3/7; 25/8 – 3/4 into 45/14
So 25/8 – 135/56; 175 – 135 is = 40; so 40/56; 40/56 is 20/28 which is 5/7
Now $C_j – Z_j$, $X_1$ and $X_2$ will have 0s 4 into 2/7 is = 8/7, 8/7 – 9/14; 16/14 – 9/14 is 7/14 which is 1/2.
You get a – 1/2; – 12/7 + 12/7 is 0; 20/7 + 135/14 is 20/7 +135/14 is
40 + 135 = 175; 175/14
Now 175/14 is what we have 5/7 45/14
So 20/7 + 135/14 would give us 25/2 which is 100/8
So we get100/8 or this can be written as 25/2, 175/14 divided by 7 is = 25/2

Now we get the same value of the objective function which is understandable because our $C_j - Z_j$ was 0. Even though theta was positive it did not improve.

This is expected now in fact the same $C_j - Z_j$ row is repeated. This $X_5$ which is a non basic variable now has a $0(C_j - Z_j)$ and wants to enter. If we have entered this then we should enter this and we try to enter this. What happens? We enter this only to find out theta there is only one value because there is a negative here.

So this will be the living variable 45/14 divided/4/7 is = 45/14 into
So this will be 45/8
There will be only variable. This will be the pivot.
If you do one more iteration what will happen? $X_5$ will replace or $X_4$ will replace $X_2$.

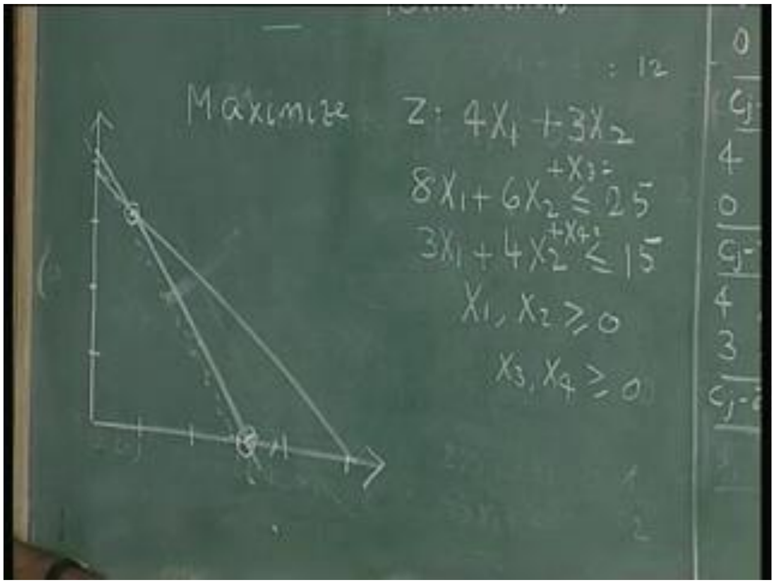
So $X_1$and $X_4$ will be the basic variable. If you do one more iteration, you will end up getting exactly this and now with this you will try to enter $X_2$ again and so on.
So if you apply the termination condition very strictly you will not terminate at all and you will get into an infinite loop. So the termination condition has to be redefined that, if you have a situation where the maximum $C_j - Z_j$ is 0 and it tries to enter then it indicates alternate optimum. It indicates this is optimum. Both of them satisfy our termination condition.
So termination condition has to be rewritten by taking care of this alternate optimum. This problem has right now two optimum solutions as shown by simplex.

One would be $X_1$ equal to 25/8; $X_4$ = 45/8; Z = this. The other one is X1 = $X_2$ = Z = this.
There are two optimum solution that simplex will indicate. So this is an example of Alternate optimum. Alternate optimum is shown when the algorithm terminates and there is a non basic variable which can enter with 0. If you enter you do one more iteration and in that iteration also something else will try to enter and you will switch back and forth.
This is not continuous loop. The termination condition has to be modified to take care of this happening which is called alternate optimum. Now there are there one more issue so right here we have recognized that the alternate optimum exists.

Now is it necessary to compute the other solution that simplex can show?
The answer is yes. It is always better to compute the other solution.
Now two things can happen. If you look at this solution $X_{1,}$ let us assume $X_1$ and $X_2$ are the decision variables which could represent the number of tables (or chairs) to be produced.
Now this is a case where you produce 25/8 of one product and you do not use $X_4$, the presence of a slack variable in the basis indicates that the resource is not utilized fully. So you still have some resource available with you and you make the same profit of 25/2.

Here you produce both of different quantities and end up making the same profit using all your resources. So depending on the situation the decision maker can either choose this solution or this. It will be apparent that this would be better because you produce only one product. It may be easier. You end up saving some resources and so one could do this. Something else can also happen though not evident from this example. It might turn out that one of the alternate solutions could be integer value. So if $X_1$ or $X_2$ represent products then it would make a lot of sense to choose that solution which is integer value. We automatically can implement it without worrying too much about the continuous variable. Therefore we need to look at alternate optimum termination condition which has to be redefined for the alternate optimum and both the solutions have to be evaluated. Now the next question is in the case of alternate optimum do we have only two solutions or do we have more than two solutions? To understand that let us look at the graph corresponding to this problem.

(Refer Slide Time: 47:11)




So let us go back to the graphical method again and draw the graph, $8X_1 + 6X_2$ less than or equal to 25. So let us look at some points here. So this is this would give us for example (25/8, 0) which is somewhere here and (0, 25/4) which is slightly more than 4 and may be you have a point here.

This is the line $3X_1 + 4X_2 = 15$ would give us 15/4 with a slightly less than four. It would somewhere here and sorry X is = 5.
This is a point and here it is slightly less than (4 3 4) so is the other point. So this is the feasible region corresponding to this problem. Now let us draw the objective function $4X_1 + 3X_2 = 12$.
So I have (3, 0) and I have (0, 4) which is this. So my objective function is line in is like this (Refer Slide Time: 47:30). So what do we observe? We observe that the objective function line is actually parallel to this. If $4X_1 + 3X_2$ is the objective function line you have $8X_1 + 6X_2 = 25$ drawn here.

So they are parallel. So the objective function is parallel to one of the constraints and when the objective function is parallel to one of the constraints and the constraint is binding, if that constraint is going to dictate the objective function somewhere as in this case, we just move, this is the point of intersection, this is your point, this is one of the points that you have here. Basically this is the point which is 5/7 and 45/14.
So when the objective function is parallel to one of the constraints, as the objective function moves it will actually not only pass through two points which are shown as two solutions here.

It would actually touch this completely. It will touch this line and then it will move so it is not that there are only two optimum solutions but actually every point which is between this and this is also optimum.

So you have an infinite number of alternate optimum when alternate optimum exists. Why then simplex is not able to show the infinite number of optimum? The reason is very simple that simplex will only show corner points. Simplex will show this point and that point. The very fact that simplex wants to switch between these two solutions is indicative of the fact there every point that is in between these two corner points is optimum.

So we need to look at alternate optimum explicitly when we are trying to define the termination condition. Alternate optimum is indicated by, the optimality or the termination condition is being satisfied. Firstly, there is a non basic variable with the 0 value of $C_j – Z_j$ which would want to enter. That would give the same value of the objective function but with the different solution.
We will have this phenomenon happening. The other variable will try to enter and if we tried to enter we keep switching between these two alterations.  So termination condition has been taken care of and has been defined for alternate optimum. Second, when we have alternate optimum indicated at any point necessary, it is important to evaluate the other solution as well. For two reasons that we solve, one of them could save some resources. The other reason could be one of them could give us an integer value solution. For these reasons it is advantageous for us to try and evaluate both these. Now going back you also need to understand that alternate optimum does not mean only two optimums. It means many.

Simplex will indicate only those two corner points which the objective function line passes through. But every point that joins these two corner points is also optimum.
So we have seen one aspect of termination which is alternate optimum here. We have already seen degeneracy. There are still two more things to be seen. They are unboundedness and infeasibility which we will be seen in the next lecture.