Advanced Operations Research

Prof. G. Srinivasan

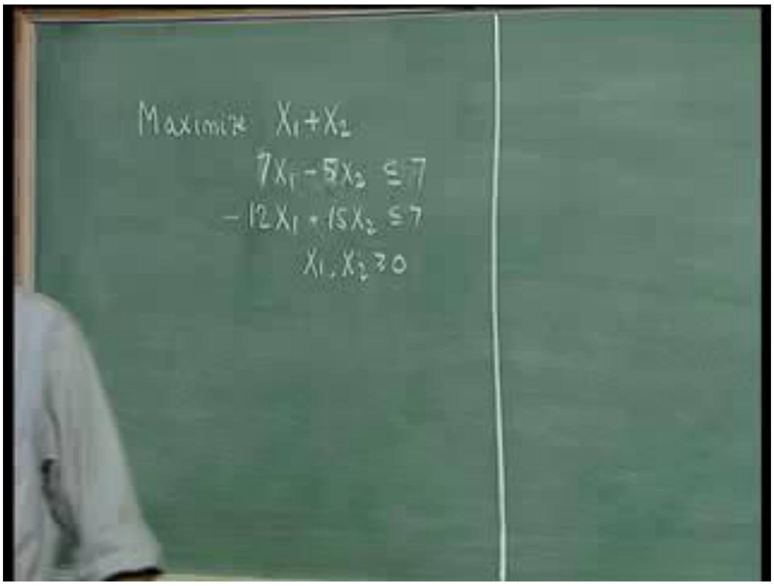
Department of Management Studies

Indian Institute of Technology, Madras

Lecture- 17
All Integer Primal Algorithm

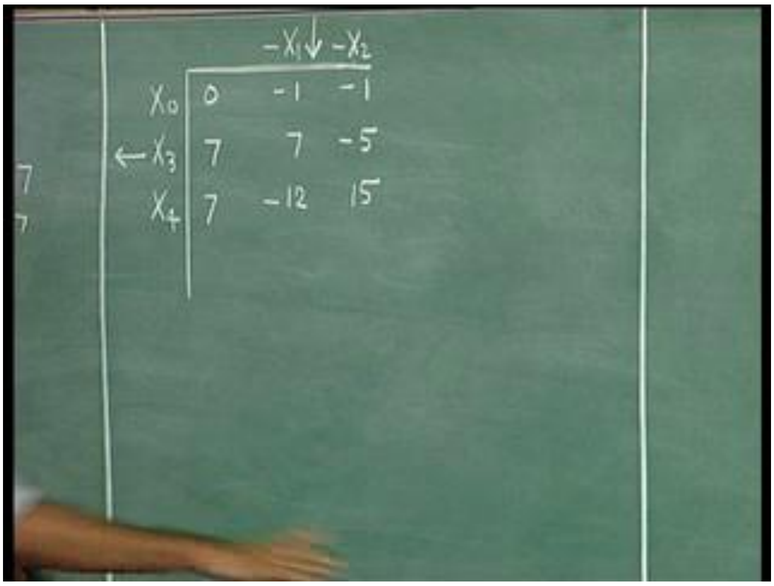
We continue our discussion on solving integer programming problems.

(Refer Slide Time: 00:18)



We show an efficient representation of the simplex algorithm. We first take the same problem that we are solving as an IP; first solve it as a relaxed linear programming problem and demonstrate a slightly more efficient way of representing the simplex table. Now this problem has two slack variables X_3 and X_4 that would come in and these two slack variables qualify to be the starting basic variables.

(Refer Slide Time 00:54)

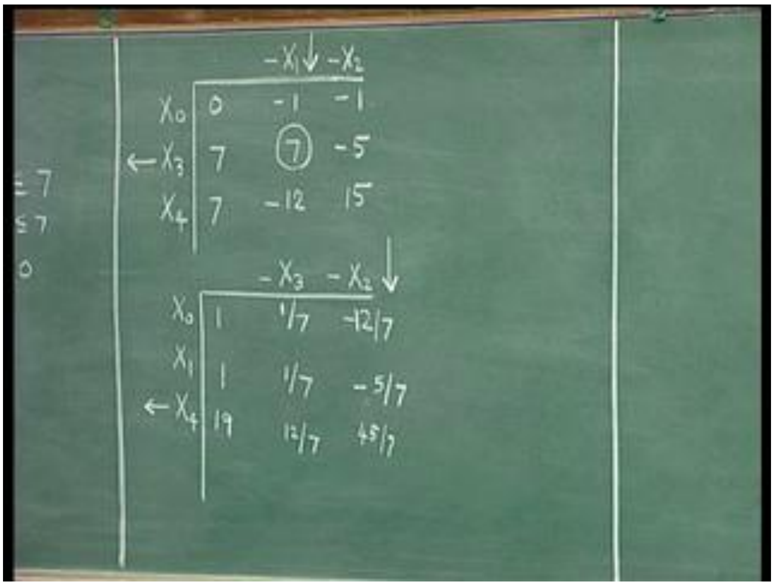


The simplex table looks like this with an X_0. The two basic variables X_3 and X_4, will figure here. The right hand side will come as the first column and the two non-basic variables X_1 and X_2 with the minus sign, will figure there. Starting with the X_3 and X_4, we would get a basis that is feasible to the primal and infeasible to the dual. These things are written as minus 1 and minus 1, so that it becomes X_1 plus X_2 and maximize. So, it appears as it is. You get a plus X_1 plus X_2 that you maximize and since we have written it in terms of minus X_1 and minus X_2, you would get a minus sign here. The minus sign indicates that at present the dual is infeasible.

Now, this would be at 0. This would give us X_3 equal to 7 minus $7X_1$ plus $5X_2$. This would be 7 minus $7X_1$ plus $5X_2$ and X_4 is 7 plus $12X_1$ minus $15X_2$ so, plus $12X_1$ minus $15X_2$. The plus 12 will become minus 12 because we have a minus here. Now this solution is feasible to the primal and infeasible to the dual, which is indicated here. So, the most negative dual variable will enter. In this case, there is tie for the entering variable and we choose to enter X_1. Now we use the same steps in simplex to find out the leaving variable, which is found after computing theta. Theta is 7 divided by 7 is 1. 7 divided by minus 12 we do not do that so there is only 1 candidate for leaving variable. So, variable X_3 automatically leaves.

Now, we create the next simplex table which runs like this with the following rows.

(Refer Slide Time 03:09)



Now, variable X_1 replaces X_3 in the basis. So X_0, X_1 and X_4 with minus X_3 and minus X_2. This is the pivot element. Rules are: pivot first becomes 1 by pivot element. So this becomes 1 by 7. Rest of the elements in the pivot row, get divided by the pivot. So 7 by 7 is 1 and we get a minus 5 by 7 coming here. Rest of the elements in the pivot column, get divided by minus pivot. So this becomes 1 by 7. This becomes 12 by 7 and then, we have to fill the remaining columns using this rule.

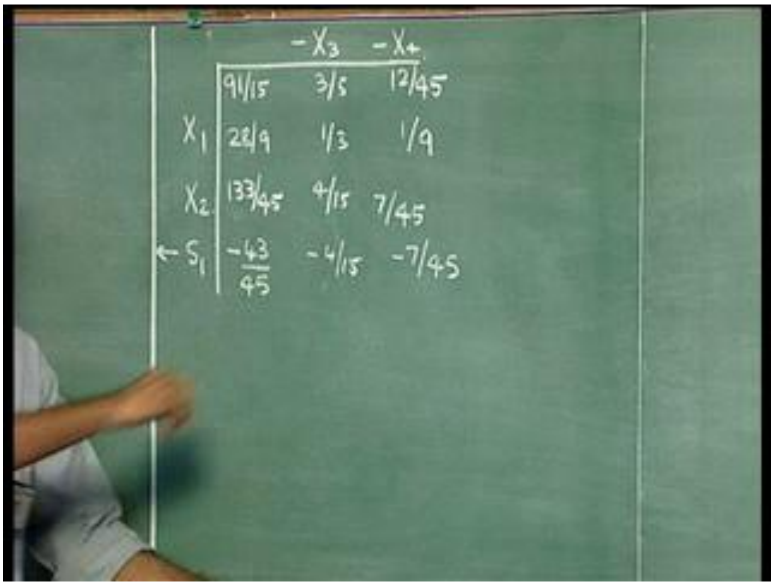
The number that you will have here will be the corresponding number. This being the pivot column this element; so 0 minus minus 1 into 1 is plus 1; 7 minus minus 12 into 1 is 7 plus 12 which is 19; minus 1 minus minus 1 into minus 5 by 7. This will be three minuses, so minus 1 plus 5 by 7 is minus 2 by 7. Let us check again, minus 1 into minus 5 by 7 will be minus 12 by 7. 15 minus minus 12 into minus 5 by 7 will be 15 minus 60 by 7, which is 45 by 7.

Now, we should go back and check. We have the simplex table already here. We go back and check, you would get the same terms there. This is what you would have at the end of the first iteration of the normal simplex table, when you write the basic variables in terms of the non-basic variables. This represents a part of the old simplex table, the part corresponding to the present non-basic variables alone and the basic variables remain as they are. A typically four column problem now becomes three column. The reduction is not very significant but later you will realize that it actually becomes significant.

Now we proceed again. Again this is primal feasible and dual infeasible. So this 12 by 7 will enter. We have to find out theta and the corresponding leaving variable. 1 divided by minus 5 by 7 is not carried out, because it is a negative here. So, there is only one case to leave, which is this. X_4 is the leaving variable and X_2 will replace X_4. So, the theta will be 19 divided by 45 by 7 which is 133 by 45.

Now we go back and write the next one here.

(Refer Slide Time: 06:47)



X_4 replaces X_2. So, I have minus X_3 and minus X_4 here, so I have X_1 and X_2. X_2 replaces X_4. So X_1 and X_2 become basic variables and so on. This is the pivot, so again caring out the iteration pivot element becomes 1 by pivot. This becomes 7 by 45. Rest of the elements in the pivot row gets divided by the pivot element. So, 19 divided by 45 by 7 is 19 into 7 by 45 which is 133 by 45. 12 by 7 divided by 45 by 7 is 12 by 45. Rest of the elements in the pivot column gets divided by minus pivot element. So, minus 12 by 7 divided by minus 45 by 7 is plus 12 by 45; minus 5 by 7 divided by minus 45 by 7 is plus 5 by 45 which is plus 1 by 9. Minus 5 by 7 divided by minus 45 by 7 is 5 by 45, which is 1 by 9. Now, the remaining will be: 1 minus minus 12 by 7 into 133 by 45 is what we should do. 7 will go 19 times 133; between 12 and 45, we would get 4 by 15. So, we get 19 into 4 is 36 by 15; 12 by 45 is 4 by 15 so 19 into 4 is 76 so 76 by 15 plus 1 which is… (09:27 Not Audible)

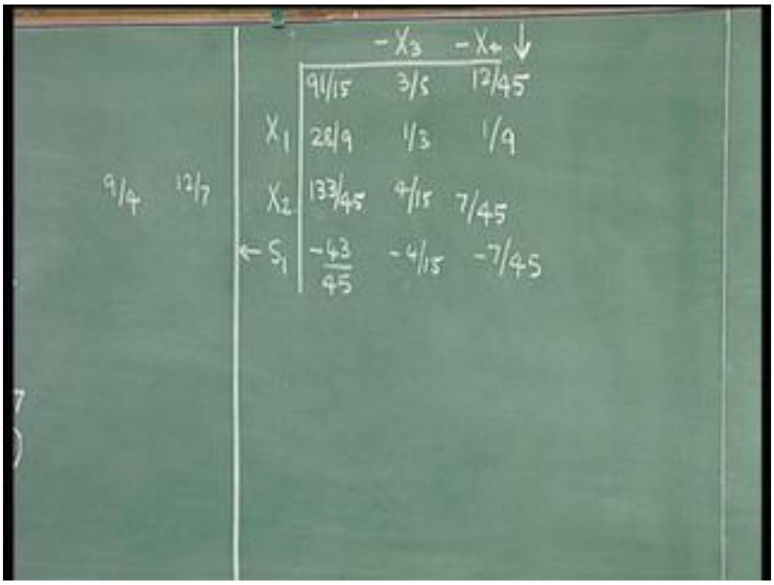
We need to do it again. See the old element; this is the pivot column, so this is the corresponding pivot column. This is the element with which we do. So 1 plus 12 by 7 into 133 by 45 is what we should be doing. We get 1 plus 12 by 7 into 133 by 45; 19, 3 into 4 is 12. 76 plus 15. We get 91 by 15, which is 273 by 45, which is the optimum. This is 1 plus 5 by 7 into 1 by 9, which is 1 plus 5 by 63; 1 minus (minus 5 by 7) into 133 by 45. So, you get 19 times, 19 into 5 is 95. Let us do it, 1 plus 5 by 7 into 133 by 45, 19 you get 28 by 9. Now this would become 1 by 7 minus minus 12 by 7 into 12 by 45. So, you will get 1 by 7 plus 144 by 45 into 7. So, this is 189 by 7 into 45, 27 by 45, is what you get here, which is 3 by 5.

In fact, this can be written as 4 by 15, so that multiplication and division is made easier. This will become 1 by 7 minus minus 5 by 7 into 4 by 15, which is 1 by 7 plus 20 by 105; 35 by 105, which is 7 by 21, which is 1 by 3. So I will get 28 by 9, 1 by 3, 1 by 9 and so on. Now this represents the optimal solution, because this is feasible to the primal, as well as feasible to the dual. The two steps of the simplex algorithm are now shown this way. At the moment, it might appear to be little more complicated and difficult, but with practice, and as we work out more examples, we will realize, particularly in the case of cutting plane algorithms, this method becomes easier. Otherwise we end up having simplex tables which become bigger and bigger iteration after another.

How do I get 1 by 3 here? 1 by 3 comes as the result of this. 1 by 7 minus minus 5 by 7 into 4 by 15, gives me 1 by 3. Now, this is optimal to the linear programming problem. This is not optimal to the IP. Therefore, we need to create a Gomory cut. Now the fractional element here is 1 by 9. The fractional element here is 43 by 45. So, we use this to create a Gomory cut. We will put one arrow mark here, indicating that this is going to be the row that is going to the give us Gomory cut. Now, we do not have to go back and write it as an inequality and then create. You can do it very easily from this itself. The Gomory cut is going to introduce variable X_5, X_5 that we have. You can write it as X_5 or you can write it as S_1, where S_1 represents an additional negative slack or surplus corresponding to the first Gomory cut. So it is a matter of terminology. You could just keep it as X_5 or S_1. S_1 would clearly tell you that S_1 corresponds to the negative slack variable that has been introduced in the first Gomory cut. This is always going to be a positive quantity, so just write the fractional elements but with a negative sign. This become minus 43 by 45. 4 by 15 would simply become minus 4 by 15. 7 by 45 would simply become minus 7 by 45. The only thing you have to do is, if you had a negative here, for example, if you had a minus 4 by 15, then you will get a minus 11 by 15 here, because this minus 4 by 15 would have become minus 1 plus 11 by 15 and plus 11 by 15 will become minus 11 by 15 here.

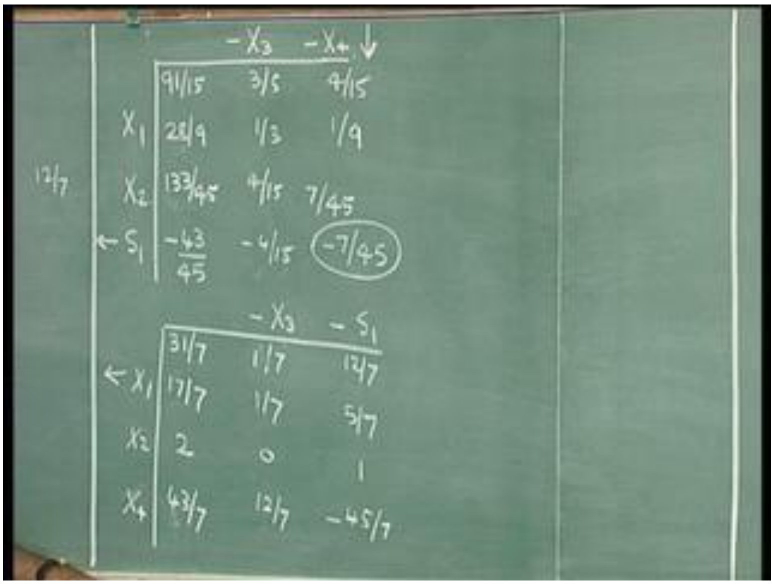
It is very easy to write the Gomory cut once you know the row from which the cut is generated. To repeat, identify the row from which you are going to create or generate the Gomory cut. Write the fractional portion of this with a negative sign here. If this quantity is positive and exceeds 1 then, write only the fractional portion with a negative sign. If it is negative then, make it a negative integer and whatever you have added to make it a negative integer or whatever balance put a minus and bring it. For example, if it is minus 4 by 15, it would have become minus 1 plus 11 by 15; that 11 by 15 will become minus 11 by 15 here. So, Gomory cut will always have negatives in all the places that we have here. So, you can easily write this cut and that is the big advantage. The bigger advantage is, since this table represents the basic variables in terms of the non-basic variables what will happen in this particularly is that these three columns will remain intact, no matter how many variables are there. The rest of them will only become basic variables and the size will become bigger. You will get additional rows but you will not get additional columns. You can do a dual simplex iteration on this, which we will proceed now. Once we have created this, we go back and we have to do a dual simplex iteration; so, you have to carefully remove this arrow.

(Refer Slide Time: 17:28)



This arrow only indicated briefly the row from which the Gomory cut is generated. Now that the cut has been generated, do not worry about that arrow any more, we are going to do a dual simplex iteration. This arrow will now indicate that this is a leaving variable. Now, go back and identify the entering variable. To do that you have to divide, 3 by 5 minus all with the minus sign; so, you get a minus 3 by 5 divided by minus 4 by 15 which is minus 3, minus 3 by 5. Again it is a maximization problem. If you want choose a minimum theta, based on theta being positive; you can define theta being positive or negative. If you want your theta to be positive, you can simply do minus 3 by 5 divided by minus 4 by 15. Minus 3 by 5 will become minus 9 by 15 divided by minus 4 by 15 is plus 9 by 4 here. You get minus 12 by 7. So, 12 by 7; 12 by 7 is smaller than 9 by 4, so this will be the entering variable. Now, this being the entering variable, we will create this.

(Refer Slide Time: 18:58)



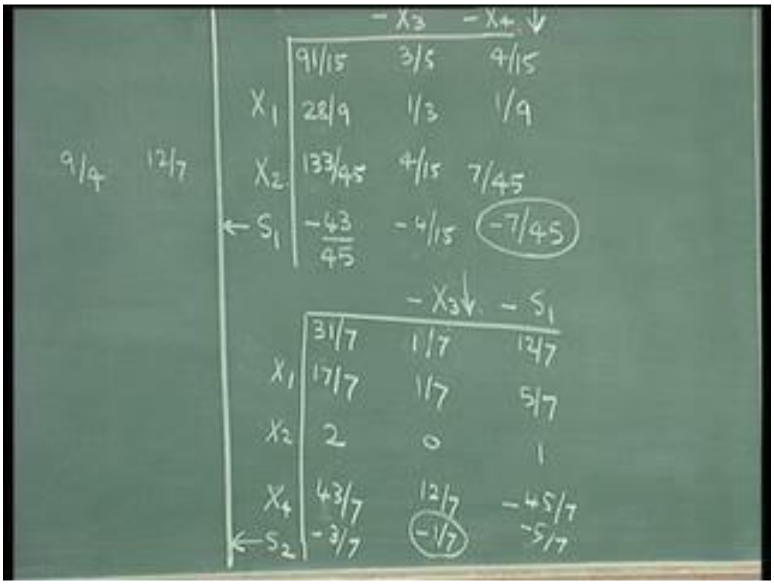
So, X_4 will replace S_1. You will have X_1, X_2 and X_4. You get a minus X_3 and minus S_1. This will be the pivot element. Pivot elements becomes 1 by pivot; so, you get minus 45 by 7, rest of the elements in the pivot row are divided by the pivot. Remember in a dual simplex iteration pivot element has to be negative, so, we have a negative pivot here. The reason being, in the next iteration this will become a positive quantity, because we are dividing by a negative. So, dividing by the pivot, I get 43 by 7. This is 12 by 45 divided by 7 by 45, so I get 12 by 7. The remaining elements in the pivot column are divided by the negative of the pivot, so 12 by 45 divided by 7 by 45 is 12 by 7. 1 by 9 is 5 by 45 divided by, I get a 5 by 7 and I get a 1 here. Now, I have to go back and do this. Actually you will realize that the moment, particularly in a table like this compared to a table like this (Refer Slide Time: 20:44), when the pivot row happens to be the last row of the table, doing it manually becomes a little more easier than the pivot row coming in the middle. As you proceed you will understand this. If the pivot row happens to be one of the intermediate rows or the middle rows, it becomes that bit more difficult when you do it manually. But when the pivot row happens to be the last row, you can easily carryout the computations, if you are doing it manually.

Let us do that. All you need is 91 by 15 minus 3 by 5 into 43 by 7; 91 by 15 minus 129 by 35 91 by 15 minus 12 by 45 into 43 by 7. You do this little carefully; 91 by 15 minus 12 by 45 into 43 by 7. I do not think I can cancel anything here. This gets multiplied by 9 or this gets multiplied by 21, so you get 91 into 21 minus 12 into 43 divided by 45 into 7 which become 31 by 7 here. We already have those numbers anyway but you need to multiply and make sure that you get the right number there.

This becomes 28 by 9 minus 1 by 9 into 43 by 7. This becomes 28 into 7 which is 196, minus 43 which is 153 by 27. So, by 9 into 7 right 63. This is 17 by 7. This is 133 by 45 minus 43 by 45 which is 90 by 45 which is 2. 3 by 5 minus 12 by 45 into 12 by 7, which is 3 by 5; minus 144 by 12 by 44 into 12 by 7. So, 12 by 45 is 3 by 9 or 12 by 45 is 4 by 9. 3 by 5 minus 4 by 15 here. So 3 by 5 minus 48 by 105, you get 3 by 5 minus 48 by 105; 15 by 105, which is 3 by 7. I get a 3 by 7 here? 15 by 105 is 1 by 7. I get 17 by 7. What do I do? 1 by 3 minus 1 by 9 into 12 by 7; 1 by 3 minus minus 4 by 21 so, 3 by 21, which is 1 by 7. You get a 1 by 7 here? 1 by 7. 4 by 15 minus 12 by 45 is another 4 by 15. You get a 0 here. So, this is what you have at the end of this iteration.

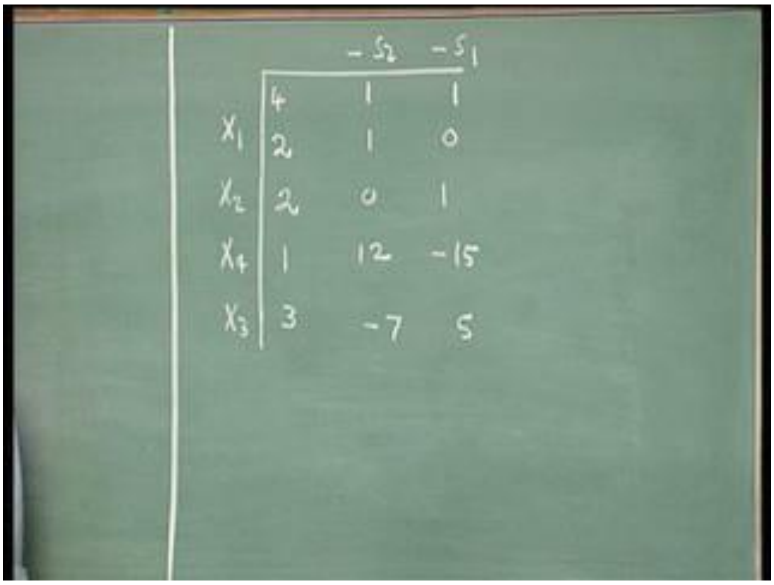
Let us proceed again. This is again a solution that is infeasible to the integer programming problem. We need to create another Gomory cut. Take the one which has the largest fractional value, this has 3 by 7 and this has 1 by 7. This becomes the row from which the Gomory cut is generated.

(Refer Slide Time: 26:01)



Gomory cut would give you an S_2 here. 17 by 7 is 2 plus 3 by 7, so I get a minus 3 by 7 here. 1 by 7 will become minus 1 by 7 and 5 by 7 will become minus 5 by 7. We do not have negatives so we do not have to worry about the other things nor do we have numbers which are greater than 0. If you have a positive number greater than 0, fractional portion will come as a negative sign here. If you have the negative number, then you have to convert into a negative integer plus a positive fraction and put a minus for the positive fraction. The last row is the leaving row. Now 1 by 7 divided by 1 by 7 is 1. 12 by 7 divided by 5 by 7 is 12 by 5; so, variable X_3 again enters and when variable X_3 enters, you have a table which looks like this.

(Refer Slide Time: 26:59)



I have minus S_2 and minus S_1. I have X_1, X_2, X_4 and X_3.This is your pivot, 1 by 7 is the pivot. Pivot element becomes 1 by pivot, so I get a minus 7 here. This is divided by the pivot; therefore, I get 3 and I get 5. This becomes divided by negative of the pivot so I get a 1. 1 by 7 divided by 1 by 7 is 1, another 1, 0 and 12. 31 by 7 minus 1 by 7 into 3, 31 by 7 minus 3 by 7 is 4. 17 by 7 minus 3 by 7 is 2. 2 minus 0 into 3 is 2. 43 by 7 minus 36 by 7 is 1. 12 by 7 minus 1 by 7 into 5, 12 by 7 minus 5 by 7 is 1. You can actually in some sense stop here but, it is all right, it is better to do the whole thing. The fact that you have an integer for the primal is enough. Right here, you could have stopped because this is going to be in any case a dual simplex iteration. Therefore, the optimality condition would be satisfied. You could have stopped right here, but let us complete it. So, 5 by 7 minus 5 by 7 is 0. 1 minus 0 is 1. Minus 45 by 7 minus 60 by 7 is minus 105 by 7 which is minus 21, hence optimal.

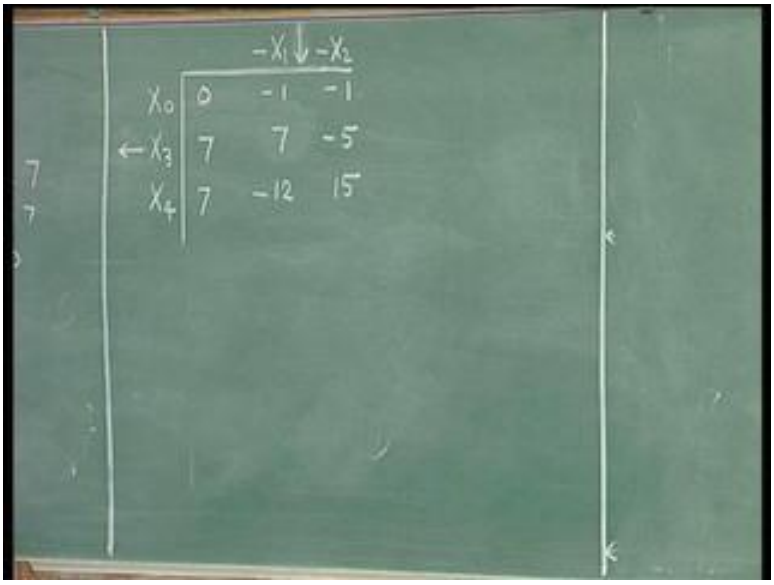
This is how you carry out the Gomory cutting plane algorithm using the slightly efficient representation of the simplex table. Computationally, it is only as good or as bad as the simplex iteration. This becomes handy, particularly when you solve it by hand. As well as, in problems, where the normal tabular form of the simplex would mean that at every iteration I am including a row as well a column; whereas here, I keep the number of columns fixed. Number of columns is equal to number of non-basic variables plus the right hand side. The number of rows increases with every iteration and the advantage is that the Gomory cut is always represented in the last row or in the additional row. As I had indicated earlier, when the leaving variables come in the last row, it becomes that bit easier to carry out the computations by hand corresponding or in compared to a situation where, the leaving variable comes in between. This is the efficient representation of the… (Refer Slide Time: 30:42). This is minus 15, what happens there? 105 by 7 which is minus 15. This is the solution to this.

Now, let us look at another algorithm. We said we have two more algorithms which are there. Something called the all integer dual algorithm and all integer primal algorithms. Let us try to solve this problem using the all integer dual algorithm and the all integer primal algorithm. We will first take the all integer primal algorithm. The problem that we have on hand is the correct example to do it by the all-integer primal algorithm. A primal algorithm by definition is an algorithm that starts with the solution that is basic feasible to the primal, dual is infeasible and carries out iterations till the dual becomes feasible. Simplex is a good example of a primal algorithm.

(Refer Slide Time: 31:29)

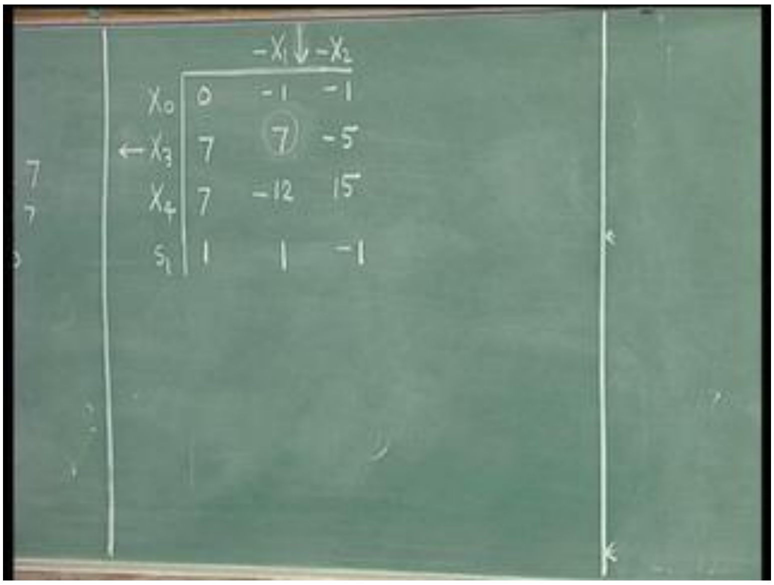
Any maximization problem assuming non-negative coefficients in the objective function with less than or equal to constraints and non-negative right hand side is a fit case for a primal algorithm. This problem has that. We will take this problem as an example problem and try to work out the all integer primal algorithm. This is the first table as we have created the same way.

(Refer Slide Time: 32:14)



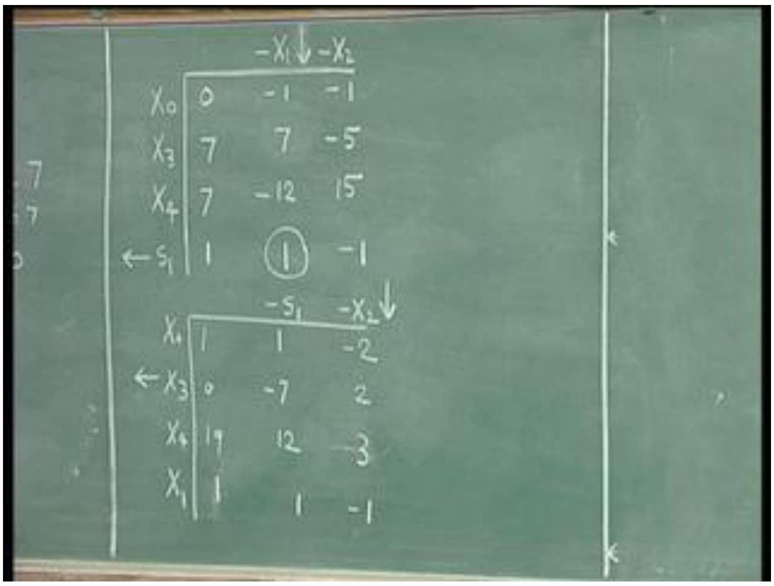
This is feasible to primal and infeasible to the dual. Most negative enters; this enters. We find out the theta, 7 by 7. Right now I am not going to put a pivot here. I am going to write a 7 here. We begin this with the same X_3, X_4 as basic variables, X_1, X_2 as non-basic variables and the first table will appear as it appeared in the earlier case. X_1 will enter and we have to find out the leaving variable exactly as we did by the simplex algorithm. 7 divided by 7 is 1. There is only one case candidate for a leaving variable. This becomes the row, X_3 will not become the leaving variable but, we will try to make a cut directly here. In the Gomory's cutting plane algorithm, we solved the problem up to the LP optimum and from there we made a cut. Now, what we will do here is we will not to take it to the LP optimum, right here we will decide to make a cut here using this row as the cut generating row and the cut will be written here, as we did in the Gomory cutting plane. This would introduce a new variable called S_1. This is a temporary pivot that you have created. I will show it by a light circle.

(Refer Slide Time: 33:52)



What you have to do here is divide every element of the pivot row by the pivot and write the lower integer value corresponding to that. 7 divided by 7 is 1, 7 divided by 7 is 1, minus 5 by 7 will become minus 1. Write the lower integer value corresponding to this. In the process, what will happen is, now this goes. The pivot element will always become 1, so this is your pivot. Now, perform a simplex iteration. You anyway have a dual that is infeasible, as a result of which you started the simplex iteration. Enter X_1 now. This will not go. S_1 will leave the basis. You will have the next table which will appear like this.

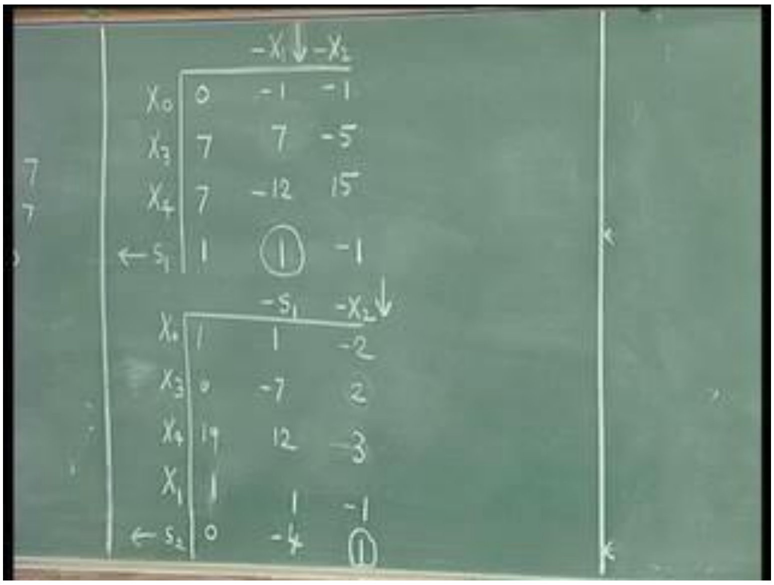
(Refer Slide Time: 35:04)



This will have minus S_1 minus X_2; X_0, X_3, X_4, and X_1. X_1 will enter, S_1 will leave. What is the advantage of this? Pivot element is always 1. Therefore, the integer nature of this table is maintained. If the pivot is a plus 1 or a minus 1 then, the integer nature of this table is maintained because the places where you may get a fraction is, when you divide by the pivot or minus of the pivot. Both these will become integers. In all other places you are only multiplying and subtracting or adding. The places where you divide are the pivot row and the pivot column. The element becomes 1 by pivot, the rest of them get divided by the plus pivot and the other divided by the minus pivot. The pivot element is plus 1 or minus 1, in this case it is plus 1. The way you generate a cut, you ensure that the pivot element is always a plus 1 and therefore, the integer nature of this table is maintained. We will do that now. First you have to divide the pivot element. 1 by pivot is 1; divide the rest by the pivot element, so 1, minus 1. Also to that extent two other things happen. The feasibility of the variables that appears here is maintained because the number becomes itself.

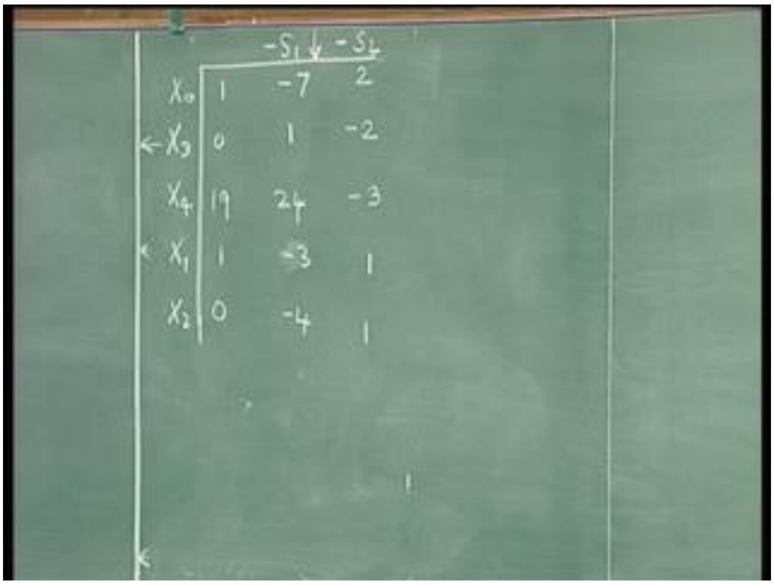
To a certain extent here, you are going to divide by the negative of the pivot. Whatever the non-optimality here that has created this table, the corresponding position becomes optimal with respect to the dual. You divide by minus 1 to get a 1, minus 7 and 12. To that extent this minus 1 becomes plus 1. In some sense you are moving towards the optimality of the dual, at least with respect to one variable, which was not optimal in the earlier case. Now, repeat the whole thing. 0 minus minus 1 into 1 is plus 1. 7 minus 7, is 0. 7 plus 12 is 19. Minus 1 minus minus 1 into minus 1 is minus 2. Minus 5 plus 7 is plus 2. 15 minus 24 is minus 9. To repeat - minus 5 plus 7 is plus 2. 15 minus 12 is plus 3, so this is what you get here. Again primal is feasible and you will also realize the way you have generated this cut, primal will never become infeasible. Primal is always feasible, because this cut comes out of a row corresponding to minimum theta. Therefore, primal feasibility is maintained and integer property is maintained plus one dual variable that was non-optimal, at least to that respect, that one variable gets a positive sign. Now, this will enter (Refer Slide Time: 38:41), because this is infeasible to the dual. Go back and calculate theta; 0 divided by 2 is 0, 19 divided by 3, this does not exist. You get this as the row that is generating your cut and the cut is generated by an S_2.

(Refer Slide Time: 38:59)



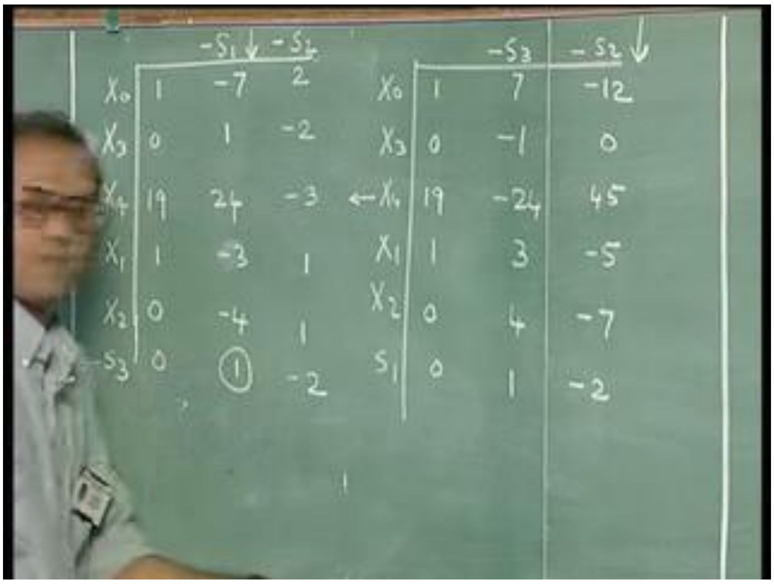
This is the temporary pivot. 0 divided by 2 is 0, minus 7 divided by 2 is minus 4, lower integer value. 2 divided by 2 is 1. This is the actual pivot and this goes and this …. Now, do the next iteration here.

(Refer Slide Time: 39:25)



I have a minus S_1 and a minus S_2. Start with the X_0, X_3, X_4, X_1 and X_2. Pivot element becomes 1 by pivot, so I get a 1 here. Divide the rest of the pivot row by the pivot element, so I get the same 0 minus 4 and 1. Pivot column gets divided by negative of the pivot. So I get a 2 here. I get a minus 2, minus 3, plus 1, and a minus 1. Very quickly, carrying out the rest of them, 1 minus minus 2 into 0, because I have a 0 here this column will repeat. So I get 1, 0, 19, and 1; minus 2 minus 1 into 1, so I get a minus 1. 2…1 minus minus 2 into minus 4 is 1 minus 8, which is minus 7. Minus 7 plus 8 is 1, 12 plus 12 is 24 and 1 plus 4 is 5. 1 minus 4 is minus 3. This is infeasible, so this variable will enter. This is the entering variable. Minimum theta, 0 divided by 2 is 0. So, minimum theta is 0. This is not Gomory cut. This is a different cut and the minimum theta is 0. That row corresponding to minimum theta will be the cut generating row from which the cut is made. Only then can you ensure feasibility of the primal in the next iteration or by doing so you ensure feasibility of the primal in the next iteration. Again this is the entering variable; again I find a 0 here. Therefore, this will be the cut generating row. I will create an S_3 here.

(Refer Slide Time: 43:00)

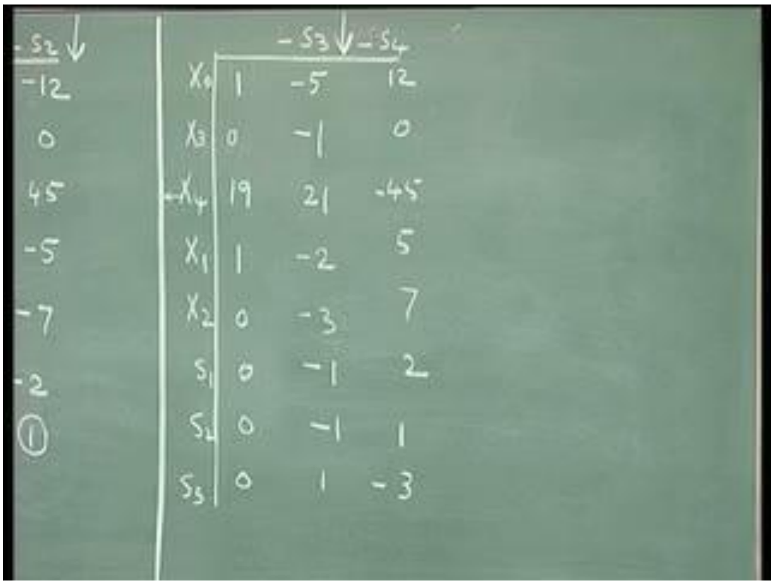


This is a temporary pivot, so I get 0, 1, and minus 2 coming here. This is the actual leaving row or leaving variable. You will do the next iteration. We will just show the next iteration right here. The next iteration will have minus S_3 and minus S_2. It will have an X_0, X_3 X_4, X_1, X_2 and S_1. This is your pivot; pivot becomes 1 by pivot, so 1, 0, minus 2 and because of a 0 here this column will repeat. So, I have 1, 0, 19, 1, and 0 before that you need to write this 1. So I get a 7, minus 1, minus 24, plus 3 and plus 4. This will become 2 minus minus 7 into minus 2; 2 minus 14 is minus 12, minus 2 plus 2 is 0, minus 3 plus 48 is 45, 1 minus 6 is minus 5, 1 minus 8 is minus 7. This is what you get at the end of this table. This becomes the entering variable because of a negative there. Compute theta to begin with. This is not a candidate for theta. This is the candidate. This is not, this is not and this is not. There is only one candidate for theta, so this becomes the row.

(Refer Slide Time: 45:20)

This becomes the temporary pivot. so divide again 19 by 45. First generate an S_4. 19 by 45 is 0, minus 24 by 45 lower integer value is minus 1 and a plus 1 here. This becomes the actual pivot and now this leaves, this enters, this X_4 goes and you write the next table here.

(Refer Slide Time: 45:50)



Minus S_3, minus S_4; X_0, X_3, X_4, X_1, X_2, S_1, S_2. Pivot is 1, so pivot row becomes the same row 0 minus 1 1. Pivot column becomes, rest of them become minus of that. So I get a 12, 0, 45 with a minus, plus 5, plus 7 and plus 2. Because of this 0, this column will repeat; 1 0 19 1 0 0. Now, what do I have? I have a 7 minus minus 12 into minus 1 is 7 minus 12 which is minus 5; minus 1, minus 24 plus 45 is 21, 3 plus 3 minus 5 is minus 2, 4 minus 7 is minus 3, 1 minus 2 is minus 1. Again I have a negative that is coming here, so this enters. I want to see what the candidates for theta are. There is again only one candidate for theta because this is not a candidate; this is not, this is not, this is not, this is also not because of negative. This is the only one that comes and this still not going to help us because there is a 21 and 19 here. Nevertheless, we proceed with this. This becomes a temporary pivot, dividing by the pivot I create an S_5 here with again a 0 1 and minus 3. 21 by 21 is 1. Minus 45 by 21 is minus 2 point something which becomes minus 3. Again carry out another iteration which is shown here.

(Refer Slide Time: 48:07)

I have a minus S_5 and a minus S_4; X_0, X_3, X_4, X_1, X_2, S_1, S_2, S_3. This is the actual pivot, so you get a 0, 1, minus 3. This becomes negative of it, so you get a 5, 1, minus 21, 2, 3, 1, 1. Again because of the 0, the same column will repeat. I have 1 0 19 1 0 0 0. Now, this will become 12 minus (minus 5) into minus 3; 12 minus 15 is minus 3, 0 minus 3 is minus 3, minus 45 plus 63 is 18, 5 minus 6 is minus 1, 7 minus 9 is minus 2, 2 minus 3 is minus 1, 1 minus 3 is minus 2. Again this enters because of the negative sign here. There is only one candidate for theta, which is this but, precisely this is what we want because there is an 18 here and a 19 here. You will create an S_6 here dividing by… this becomes temporary pivot. I get a 1, I get a minus 2 and I get 1 here. This will be the actual pivot and you will have to enter S_4, S_6 will leave and carry out another iteration. We will do that little later and hopefully this iteration would give us the optimum.