Advanced Operation Research

Prof. G. Srinivasan

Dept of Management Studies

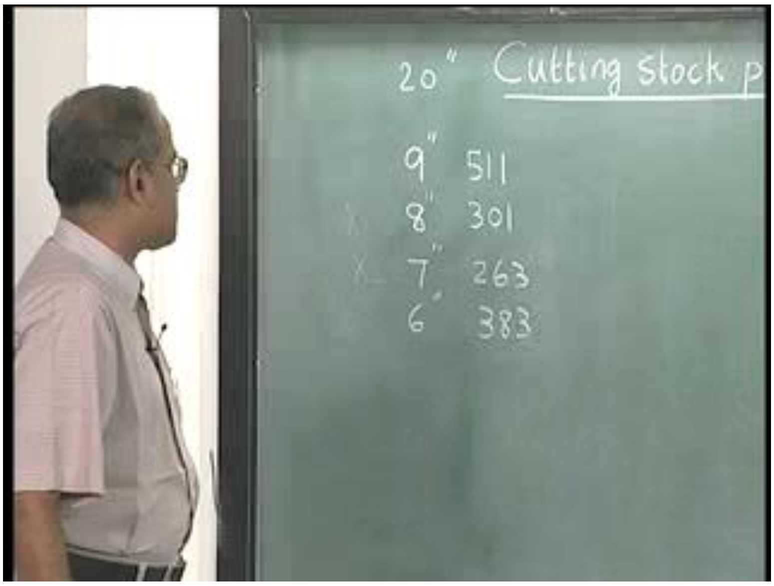
Indian Institute of Technology, Madras

Lecture - 4

One Dimensional Cutting Stock Problem

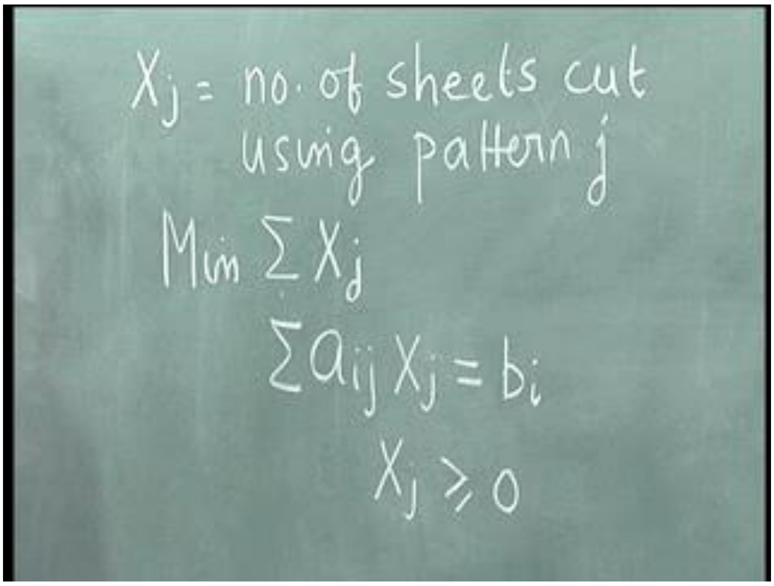
We continue the discussion on the cutting stock problem. The problem is to cut 511 of 9-inch sheets, 301 of 8-inch, 263 of 7-inch and 383 of 6-inches, out of given 20-inch sheets.

(Refer Slide Time: 00:23)



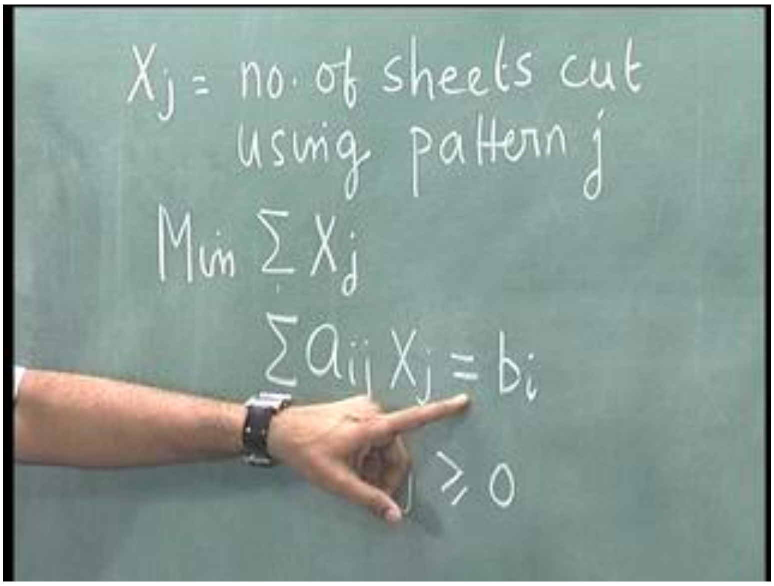
We have already seen that the problem reduces to one of minimising the number of sheets that we would require to cut 511 9-inch sheets and so on. The problem is if we define X_j equal to number of sheets cut using pattern j, the problem will be to minimise sigma X_j, which minimises the number of sheets that we cut subject to the condition sigma a_{ij} X_j is equal to b_i and X_j greater than or equal to 0.

(Refer Slide Time: 00:56)



In the last lecture we saw that if we consider the patterns where the waste can even exceed the minimum size required, which is 6, then the constraint becomes an equation. This is the problem that we wish to solve, if this problem were a linear programming problem. We also know that X_j being the number of sheets that are cut, cannot be a continuous variable. So the problem actually is an integer programming problem, where we would say X_j greater than or equal to 0 and integer. We have already seen that, at the moment, we do not know the techniques of integer programming. We would relax the integer assumption, treat the problem as a linear programming problem and try and get an integer solution from the linear programming solution. We will try and solve this in more than one stage. In the first stage what I am going to do is to tell you that it is enough to solve it as a linear programming problem, even though the linear programming problem can give fractional values for the X_js while the X_js have to be integers. Then we will go on to show how we can get a very good solution to the integer programming problem from the LP optimum. That is the first thing we will do and the second thing we will do is to try and solve the LP, the linear programming problem by exploiting this equation.

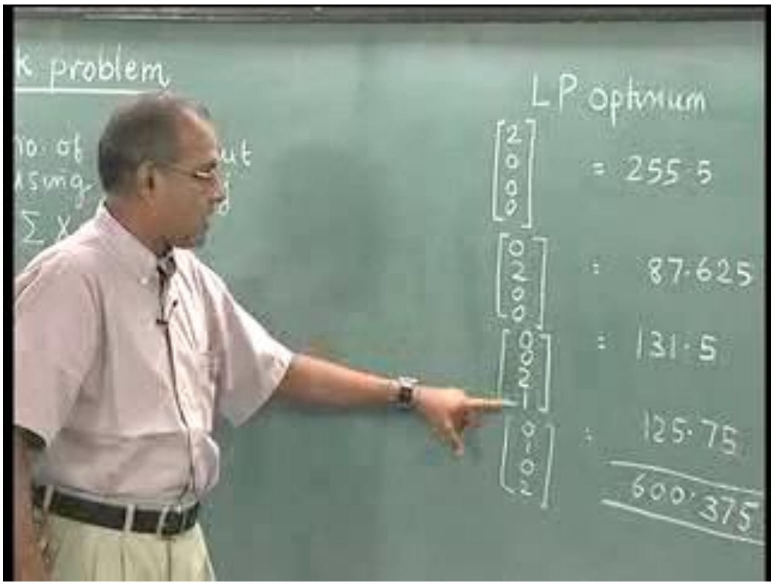
(Refer Slide Time: 03:20)



I have already mentioned that if the primal has equations, the corresponding dual variable is unrestricted inside, so we would exploit that. Then we would develop a simplex like algorithm, which uses ideas such as primal feasibility, dual feasibility and complementary slackness to solve the LP. While we develop that algorithm, we will also explain the column generation idea where we will not store explicitly all the patterns, but generate patterns or columns based on the solution to a sub problem that we will be solving. We will see three things, the first of which is to show that the LP optimum is good enough and from the LP optimum we can actually get a very good solution to the integer programming problem.

Let us, for the purpose of this discussion, assume that we have actually optimally solved the linear programming problem, which is to cut 511 9-inch, 301 8-inch, 263 7-inch and 383 6-inch sheets out of a given number or large number of 20-inch sheets and to minimise the number of such sheets that we will be looking at. Let us assume that we know the optimum solution to the linear programming problem, from which we will show why this solution is good enough and how we can get good solutions to the IP. This is the optimum solution to the linear programming problem.

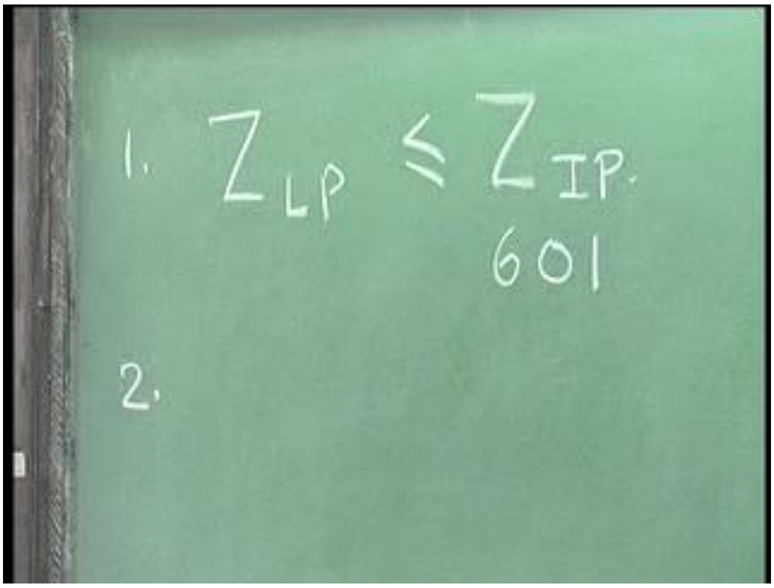
(Refer Slide Time: 04:58)



There are four types required, 9, 8, 7 and 6 which means there are four constraints here, which also means that there will be four basic variables and there will be four patterns in the solution. We can see that these are the four patterns which are in the solution; 2 0 0 0, means 2 into 9 plus 0 into 8 plus 0 into 7 plus 0 into 6, which means we are able to cut two 9-inch sheets from a 20-inch and we use 255.5 such sheets. Please note that this is an optimum to the linear programming problem and therefore this can take a fractional value. Similarly, the pattern 0 2 0 0 is used, which means we cut only 8-inch sheets and two such sheets, 2 8-inch sheets from every 20-inch and we do that with 87.625 number of sheets. Similarly, we use 0 0 2 1, which means we cut 2 into 7 plus one into 6 and use up all the 20; waste is equal to 0 in this case. We use up all the 20 and we use 131.5 such sheets. 0 1 0 2 is, 1 into 8 plus 2 into 6 which is also 20, which means there is no wastage in this pattern. We use 125.75 such sheets, or sheet we use or we cut this pattern out of 125.75 sheets.

In all we have used 600.375 sheets based on the LP optimum. What more can we understand from the LP optimum? The given problem is an integer programming problem and we have relaxed the integer assumption and we have solved the corresponding linear programming problem. If the linear programming says, we need 600.375 sheets it means the integer programming problem would mean that it will have a solution of 601 or more. So the LP optimum is always a lower bound or a lower estimate of the integer optimum. We can say comfortably from this that we would require 601 or more sheets. The first thing that we learnt from this solution is that Z_{LP} is less than or equal to Z_{IP} and it is a lower bound on the Z_{IP} which is the optimum solution to the integer programming problem.

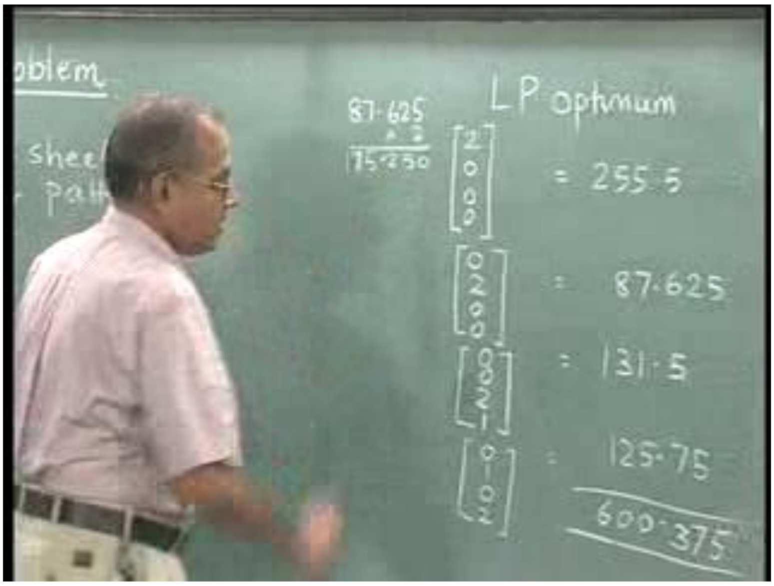
(Refer Slide Time: 07:35)



Since the problem is an all-integer programming problem, the Z_{IP} has to be an integer and therefore 600.375 become 601. We can say definitely that for this problem, we need 601 or more sheets. Also, if we are able to get a feasible solution to the integer programming problem with 601 sheets, then it is optimal to the IP because 601 is a lower bound to the integer programming problem. If we are able to get a solution with 601, then we have got the optimum solution. Before we proceed further let us also verify that this optimum solution meets this demand, so that, that will be useful to us. 9-inch sheets are cut only using this pattern. We do not cut 9-inch from these three, so for every sheet we cut 2 9-inches so 255.5 into 2 is 511 and we require 511 of 9-inch sheets.

8 inch sheets are cut using this pattern and this pattern, so 2 into 87.625 plus 125.75. 2 into 87.625 would give us 170. This is 87.625 into 2, this is 175.25 plus another 125.75 would give us 301; so, we get 301 8-inch sheets.

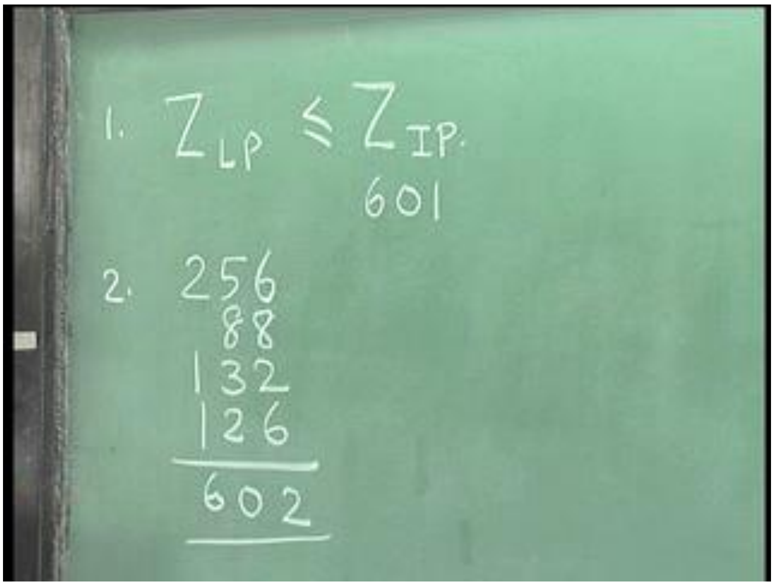
(Refer Slide Time: 09:35)



7-inch sheets are cut using only this pattern. So 131.5 into 2 is 263; so 263. 6-inch are cut using this and this; so, 125.75 into 2, this is 251.5; 251.5 plus 131.5 is 383, so we require 383. Therefore we have got this solution. This will meet all the requirements. This was more for verification purposes.

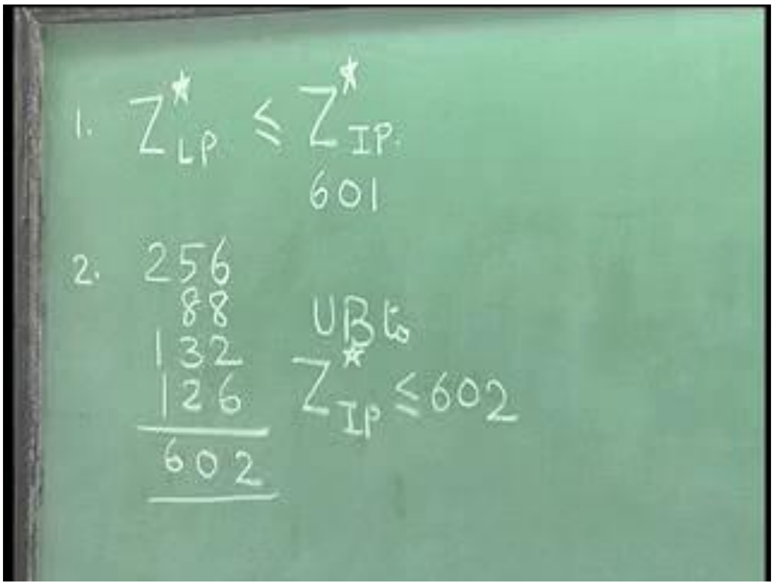
Second thing that we can do is, if instead of 255.5, if we say that we use 256, which means each one of these, if we try and put it to its upper integer value, which means we are looking at a solution with 256 sheets cut using this pattern, 88 cut using this pattern, 132 cut using this pattern and 126 cut using this pattern.

(Refer Slide Time: 11:01)



If we assume that each one of these goes to its upper integer value, automatically the solution will be feasible because right with 255.5, 87.625, 131.5, 125.75, it meets these numbers. So by increasing this, it will definitely meet these numbers. Such a solution with 256, 88, 132, and 126 would be feasible to the linear programming problem. It would also be feasible to the integer programming problem because we are able to meet all these demands with an integer number of sheets cut using each of these patterns. This is a feasible solution to the corresponding integer programming problem and this uses, 8 plus 6 equal to 14, plus 2 equal to 16, plus 6 equals to 22. 2 plus 2 equals to 4, plus 7 equals to 15, plus 5 equal to 20, and 6.

(Refer Slide Time: 12:04 min)



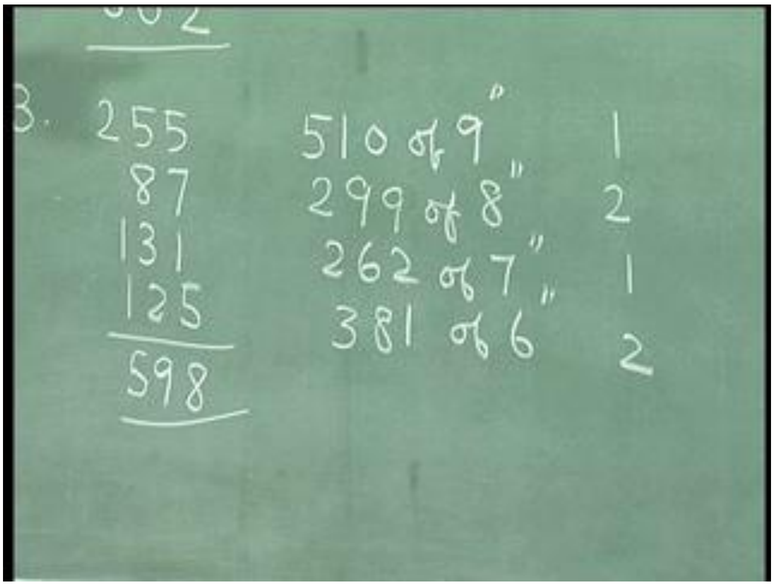
This uses 602 sheets. So, we have a feasible solution to the IP with 602 sheets and this act as an upper bound to Z_{IP} star, if you might call it, the optimum solution. Z_{IP} star is less than or equal to 602, because 602 is now an upper bound to this.

We have a feasible solution; every feasible solution to an integer programming problem is an upper bound. Here we find that the gap between the lower bound and the upper bound is 1; but in general the gap can only be less than or equal to 4, because there are four variables in the solution and each variable is brought to its upper integer value. So utmost, the increase is 1 in each of these four. The maximum increase that we can get is only 4, which incidentally is equal to four patterns. If we agree to accept a solution, which is greater than the optimum by utmost K, where K is the distinct number of types of cuts that we want to make.

In this case there are four, so if we agree that a solution which is within K of the optimum, in this case, within four of the optimum is acceptable to us. Then we can surely accept the solution with the upper integer values of these. It also happens that when we do the upper integer rounding, it does not exceed by 4, it is only exceeding by 1. But in the worst case, if we agree to have a solution that exceeds the optimum by K or within K, then we can accept this straight away as the solution. The only thing that remains as far as we are concerned is, we already have a solution with 602. We know that the lower bound is 601. The solution cannot be 600. If we are able to get a solution with 601, then it is certainly optimum. If we are not able to get a solution with 601, we are still not worse off. Two things can happen. The optimum may still be at 601 and we may not reach it or the optimum itself is 602. Even in the case that the optimum is 601 and we do not reach it, we are still worse off only by one extra sheet because we already have a solution with 602.

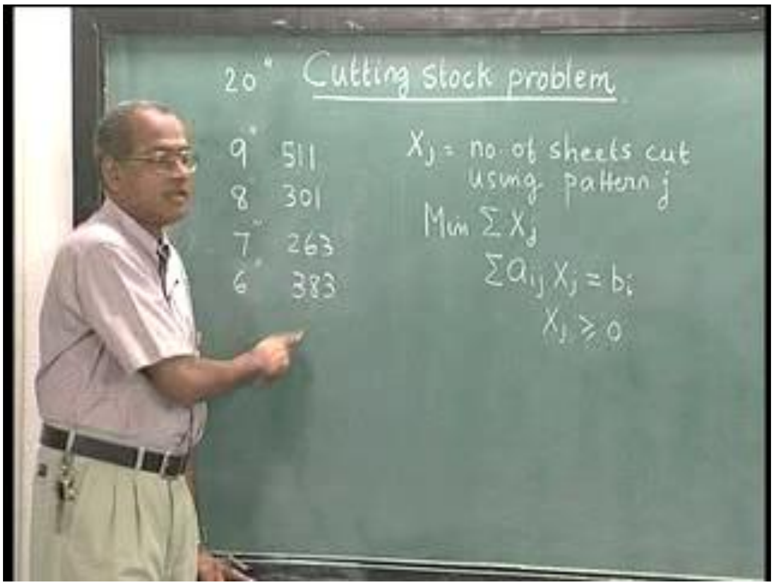
Let us look at another thing.

(Refer Slide Time: 15:24)



Let us now take each of these and round it off to the lower integer value. Let us say that we are going to use 255 sheets of the first type, we are going to use 87 sheets of the second type, we are going to use 131 sheets of the third type and we are going to use 125 sheets of the fourth type. Now, clearly this will be infeasible, because we have reduced the number. So, we will not be able to meet the demand of 511, 301, 263 and 383 with this.

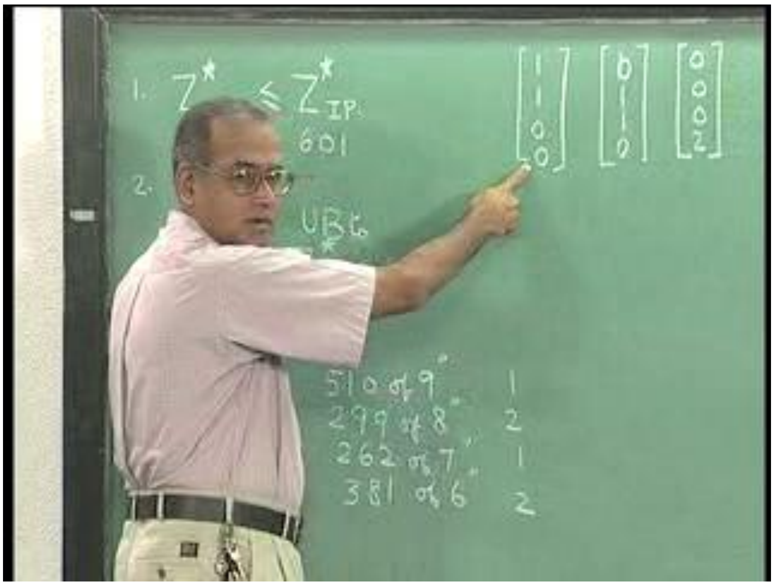
(Refer Slide Time: 16:06)



Let us go back and see how much of demand we are able to meet or how much of demand we are not able to meet, if we bring these things to the lower integer value. 255 sheets of the first one would give us 510 of 9-inch, 255 into 2 is 510, we require 511. Now 8-inch we take from here, as well as here. So 8-inch will be 2 into 87, which is 174. 174 plus 125 is 299. So we have 299 of 8-inch as against 301 in the 8-inch. So 87 into 2 is 174 plus 125 is 299. 7-inch sheets we cut only using this. So 131 into 2 is 262 of 7-inch, as against the requirement of 263 that we have here. 6-inch, we cut from this as well as this, so, 131 plus 2 into 125 which is 250. 250 plus 131 is 381of 6-inch. We are short by 1 of 9-inch, 2 of 8-inch, 1 of 7-inch, and 2 of 6-inch. This is the shortage that we have, if we bring down each of these into its lower integer value. Now, by bringing down into the lower integer value, let us see how many sheets we actually consume. We actually consume 5 plus 7 12 plus 1, 13 plus 5 18, 6 14 17 19, 3 4 5. So, using 598 sheets, we are able to meet 510, 299, 262, 381 and we are short by 1, 2, 1 and 2 sheets.

We already know that the optimum solution can only be 601 or more. If we are able to show that in three additional sheets, we can meet all these demands, then we have a feasible solution with 601. Let us see whether we can do that.

(Refer Slide Time: 20:02)



So, we are short by 1 of 9, 2 of 8, 1 of 7 and 2 of 6. What we can do is, we can now have a pattern which is 1 1 0 0, which means, we cut 1 of 9 and 1 of 8 with that. Now, we have 0 1 1 0, which means we cut 1 of 8 and 1 of 7 and we could have 0 0 0 2 from the third. We can now take three more sheets and cut 1 1 0 0 from 1; 0 1 1 0 from another, 0 0 0 2 from the third. All three patterns are feasible because 9 plus 8 is 17, 8 plus 7 is 15, 2 into 6 is 12, so it is possible to cut. This way we now have an extra 9, so this is met. We have 2 8s, this is met; we have 1 7; this is met and we have 2 6, this is met. So, these 598 sheets plus these 3 sheets give us 601 sheets and we have a feasible solution with 601.

We also know that the lower bound is 601; therefore one of the optimum solutions is this plus this, with 601 sheets. Also note that there are lot of alternate optimum, because several ways we can actually get this. We could have done 9 plus 6, we could have done 8 plus 6 here and so, there are several ways of getting the three extra sheets. Even here there could be several ways of doing it. Now, we have got the optimum solution through a simple manipulation. From the LP optimum, we have got the optimum solution to the integer programming problem. For this problem instance we have been able to do this comfortably. For other problem instances, it may not be as easy as we did here. But the principles that we have seen, the three very basic principles that we have seen, will help us in identifying a very good solution to the integer programming problem using the linear programming optimum. The three principles that we saw are the following. One is the optimum solution to the LP provides a lower bound to the IP. The upper integer value of the objective function value of the LP optimum, which is from 600.35 into 601, is a very good lower bound to the integer programming problem.

Creating all of these into the upper integer value would clearly give us a feasible solution and that feasible solution, in the worst case, will exceed the optimum only by K, where k is the number of distinct sheets that we cut, which is 4 in this example. That is the second principle.

Third principle is lower integer values of these will be clearly infeasible, but it will give us an estimate of how much less is this from the lower bound; as well as how much less these are from the demands and then by a quick algorithm, it may be possible to try and get the solution as we got here.

These three principles help us in getting the integer programming optimum from the linear programming optimum. Since this part is available which means, since we now know how to approach the IP solution from the LP optimum, we now say that we are not going to solve this problem as an integer programming problem. We are going to solve this problem as a linear programming problem to try and get this LP optimum, because we now know how to get the IP solution from the LP optimum. So the first part of the discussion on the cutting stock problem is to try and study the IP solution from the LP optimum, which we have completed right now. From the LP optimum, we have shown how we get to the IP solution.

Second part of the discussion on the cutting stock problem is to actually solve this LP and use the ideas of column generation, where we do not store every possible pattern, instead we generate patterns depending on the state of the solution. We will now look at the second part where we actually try and solve the LP to get this solution 255.5, 87.625, etc., with 600.375; so, we will look at that.

(Refer Slide Time: 26:52)

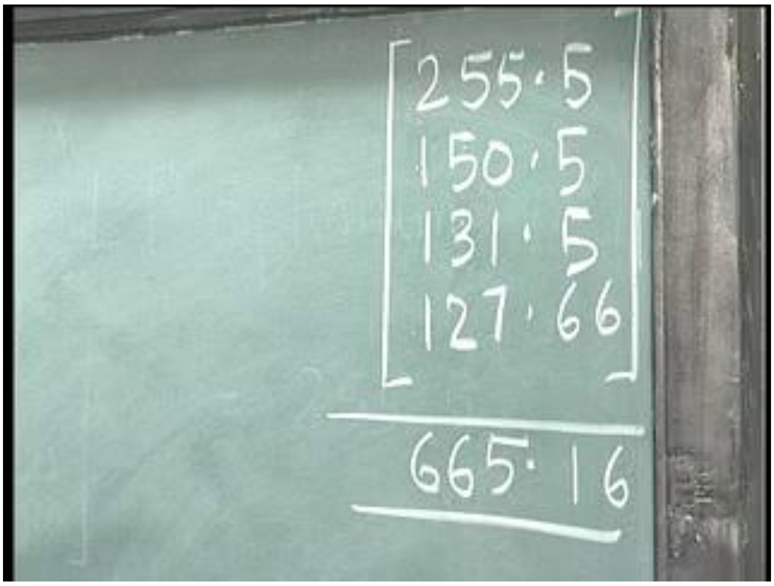
Let us assume that we have 20-inch sheets and we are going to cut these four types. The four easiest patterns to look at are patterns where we cut only 9, only 8, only 7 and only 6. If we cut only 9, the pattern that we will have is 2 0 0 0. If we cut only 8 the pattern will be 0 2 0 0. If we cut only 7 it would be 0 0 2 0. Even though, with some amount of common sense, we can say that 2 into 7 is 14 plus 1 can be cut, so that, all twenty can be used.

We also realise in this example that the number of 6-inch sheets are higher, the demand is higher. It would certainly be worthwhile to try the pattern 0 0 2 1 instead of 0 0 2 0. However, when we begin the LP optimum, we do not want to think on those lines. We would rather say that we would start with patterns where we are cutting only one at a time, which means if we take a 20-inch sheet and we are going to cut only 7-inch and not bother about the rest of them, we will end up doing 0 0 2 0. So, we will take this 0 0 2 0 as a starting pattern.

Now, as far as 6-inch is concerned, we will take 0 0 0 3, because 3 is possible. We will now say that these four are the four patterns; we may call them as X_1, X_2, X_3, X_4 but these four are the four patterns that we will use. Now, the basis matrix B corresponding to these four patterns is given this way. If we choose to cut only these types of patterns, how many sheets are we going to cut? So, if we use the pattern 2 0 0 0 and cut 9-inch sheets out of this, then for each we are going to get 2 9-inch sheets.

So, we would require 511 by 2, which is 255.5 sheets we will require to meet the demand of 511.

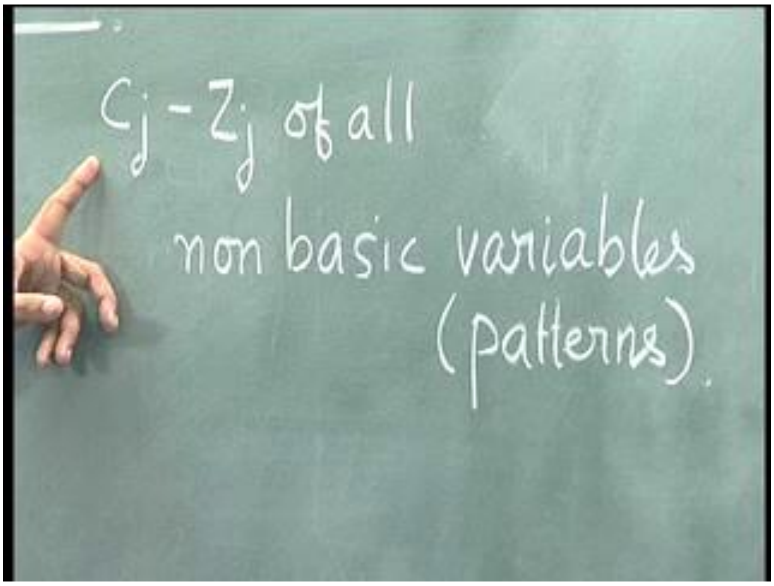
(Refer Slide Time: 27:32)



Similarly, we are going to cut only 8-inch sheets; so, two sheets at a time, so we will require 301 by 2, which is 150.5 sheets of this pattern, we will cut. Similarly we will require 263 divided by 2, which is 131.5 sheets that we require to cut using the pattern 0 0 2 0, so that our requirement of 263 is met. Now 383 has to be divided by 3, because every sheet we get 3 6-inches. So we will require 383 divided by 3 which is 127 .66.

If we follow this pattern then we would require a total of 6. Right now let us keep it as 6 6. It is actually 6 6 6 and so on. We will keep it as 6 6. So 15 plus 6 is 21, 2 9 10 15, 3 6 11 16, 2 3 4, 6. So, we require 665.16 sheets if we use this pattern. This is a feasible solution to the linear programming problem, because we will be able to meet the demand of all of these and the X which satisfies all the constraints, all the variables X is greater than or equal to 0. This is a feasible solution 665.16. We now need to check whether this is optimum.

(Refer Slide Time: 31:35)

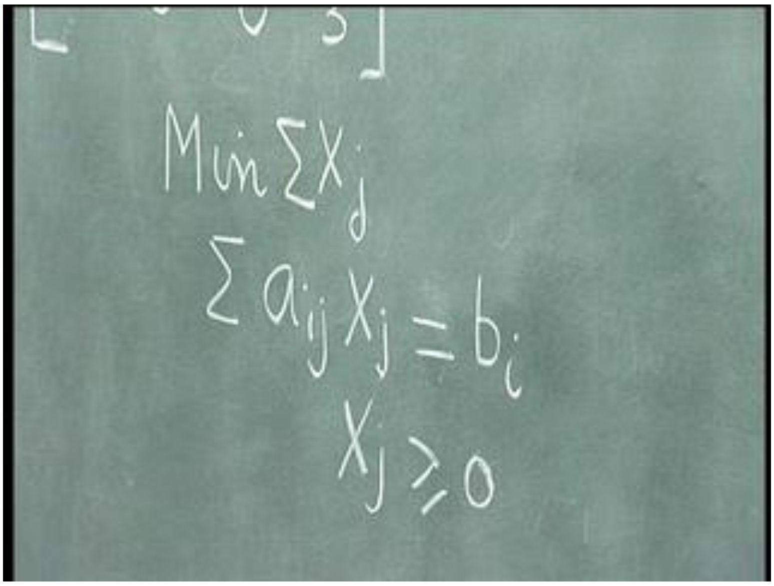


This will be optimum provided we evaluate C_j minus Z_j of all non basic variables or in this case of all non basic patterns. If we have listed all the possible patterns, then it may be possible to try and find out the C_j minus Z_j of all non basic patterns, which means all those patterns that are not here and the problem is a minimisation problem, therefore a negative C_j.minus Z_j or a positive Z_j minus C_j will enter the basis.

There are two things, one is we have not listed out all the patterns, so only when we list out all the patterns, we will be able to compute the C_j minus Z_j for all these patterns. Now that we have not listed all these patterns, we are not able to compute the C_j minus Z_j. So what we are going to say is that, if there is a pattern which has a negative C_j minus Z_j, then such a pattern can enter the basis. We will try and find out if there is a pattern which has a negative value of C_j minus Z_j. If such a pattern exists, if we are able to find a pattern with the negative value of C_j minus Z_j, then such a pattern can enter the basis. We will now try and see whether there exists a pattern and try and find out that pattern, if such a pattern exists.

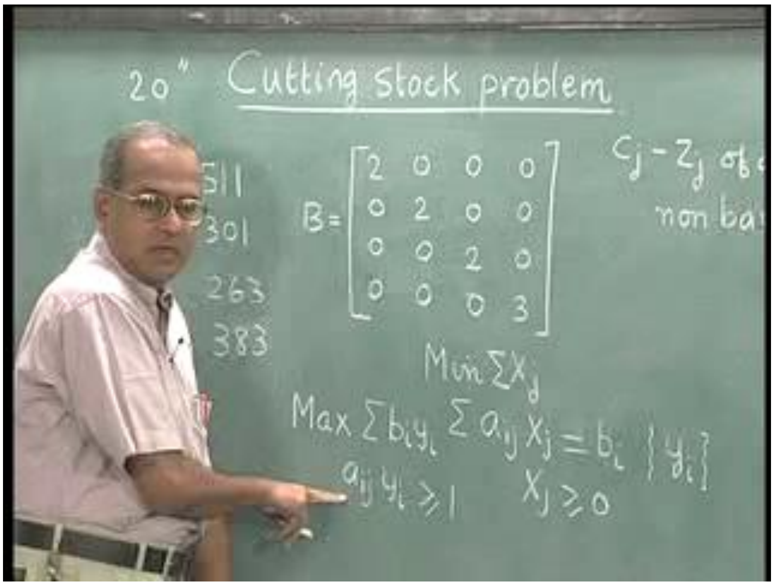
In order to find that out, let us go back to the formulation again.

(Refer Slide Time: 32:58)



We have already seen that the problem is to minimise sigma X_j, where X_j is the number of sheets cut using pattern j, subject to the constraint sigma a_{ij} X_j is greater than or equal to b_i, where b_i is the requirement and X_j greater than or equal to 0. We have already seen that by considering a large exhaustive set of patterns where the waste need not be less than 6, then the inequality will now become an equation, where we will say it is equal to b_i. In this particular example, there are four constraints, so there are four equations. Let us write the dual of this problem. The dual of this problem, let us say that it has y_i as the dual variable, and there will be four dual variables, one each associated with each of these constraints.

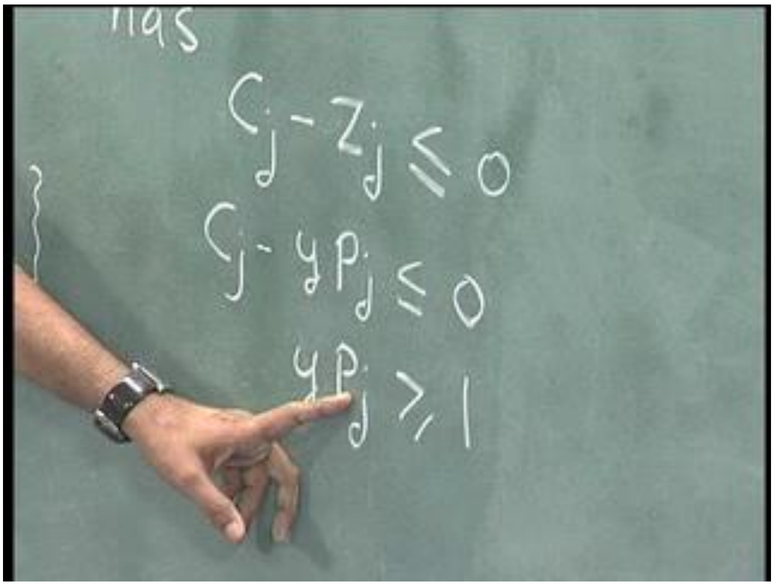
(Refer Slide Time: 34:04)



If we define this y_i here, then the dual will be a maximisation problem, which will be maximise sigma b_i y_i, such that, $a_{ij}y_i$ is greater than or equal to one. The a_{ij} comes from here; we should actually look at the transpose of that particular matrix, but, for the sake of notation I am using the same a_{ij}. Now, each y will take an element from this. The dual will have $a_{ij}y_i$ is greater than or equal to 1, because each X_j has an objective function coefficient of 1.

This is the value and we also have y_i unrestricted in sign. The unrestricted in sign comes because all the primal constraints are equations. Therefore, all the dual variables are unrestricted in sign. Here, the greater than or equal to comes because primal is a minimisation problem, dual is a maximisation problem. The primal has the correct type of variable, which is greater than or equal to here. Therefore, the dual will have the correct type of constraints, which will be less than or equal to 1 from this. The dual will be maximise b_iy_i such that $a_{ij}y_i$ is less than or equal to 1.

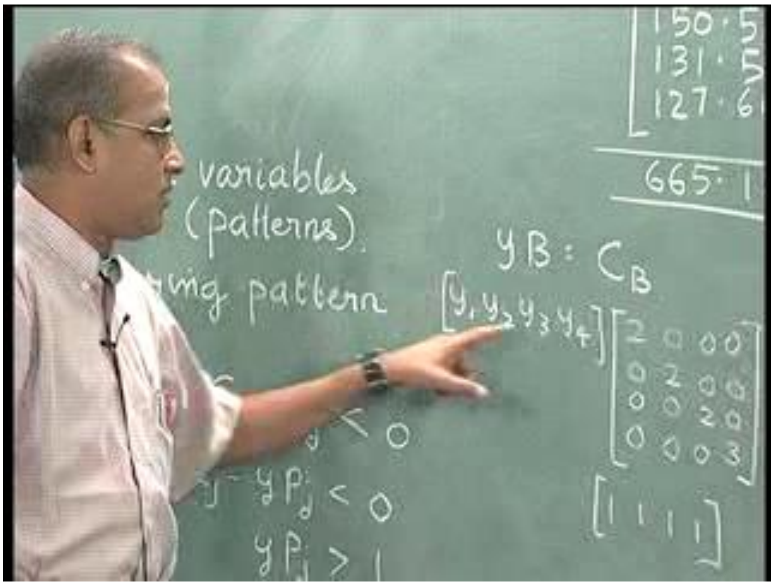
(Refer Slide Time: 36:49)



We are now interested in finding if there is a pattern that can enter the basis, which means such a pattern should have a negative C_j minus Z_j. So, an entering pattern should have a negative C_j minus Z_j less than or equal to 0. Every pattern has an objective function coefficient of 1, because the primal is sigma X_j for all C_j's are 1. This is C_j minus Z_j less than or equal to 0. C_j minus yp_j less than or equal to 0, which means yp_j greater than or equal to 1, then such a p_j can enter. If there is a pattern P_j, such that yp_j is greater than or equal to one, then such a pattern can enter and the pattern has to be feasible, therefore the pattern should satisfy, if the pattern is A, B, C, D then the pattern should satisfy the condition 9 into A, plus 8 into B, plus 7 into C, plus 6 into D is less than or equal to 20, because we should be able to cut all feasible patterns from 20-inch. Therefore, any feasible patterns contains an A, B, C, D, such that 9 A plus 8 B plus 7 C plus 6 D is less than or equal to 20 and if we have a dual y from the given basic feasible solution, you can find out y and yp_j is greater than 1, strictly negative value will enter, so that would give us yp_j greater than 1, then such a pattern will enter the basis.

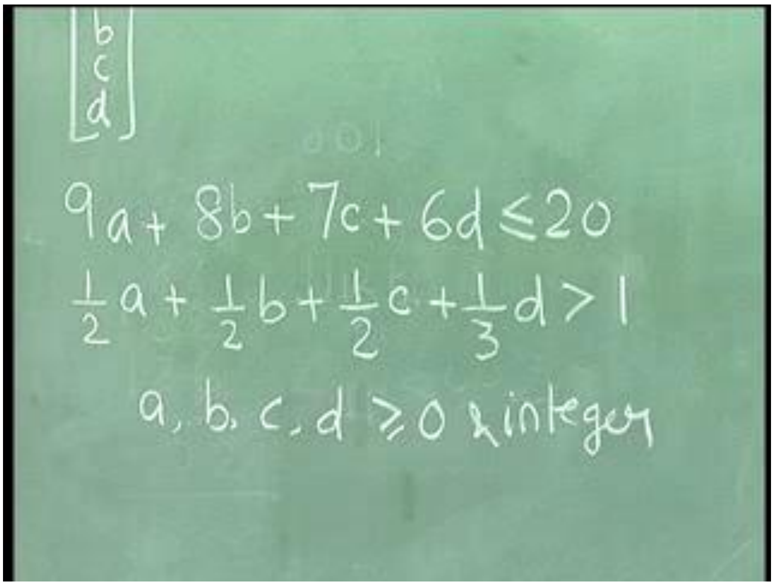
So, our job is to find out a pattern given the dual y, to find out a pattern p_j such that yp_j is greater than 1 and p_j is A, B, C, D, where 9 A plus 8 B plus 7 C plus 6 D is less than or equal to 20. Let us first find out the value of y. Now y, which is the dual is given by y B equal to C_B or y is equal to C_B B inverse.

(Refer Slide Time: 39:29)



So, y B equal to C_B, so let the dual y be y_1, y_2, y_3, y_4 into B which is 2 0 0 0, 0 2 0 0, 0 0 2 0, 0 0 0 3 is equal to C_B which is 1 1 1 1. Why the dual is y_1, y_2, y_3, $y_{4;}$ the basis matrix B is repeated from here. C_B is the objective function coefficient which is all made up of 1s. Now this matrix is very close to an identity matrix, except that the 1s' have certain numbers. So, it is very easy to compute y_1, y_2, y_3, y_4 and we need not actually invert this explicit. Simple computation would give us y will be equal to 1 by 2, 1 by 2, 1 by 2, 1 by 3, because from this 2 y_1 plus 0 y_2 plus 0 y_3 plus 0 y_4 is equal to 1, which would give us 1 by 2. Similarly, y_2 is 1 by 2, y_3 is 1 by 2 and y_4 will become 1 by 3. We have now found out the dual y. Now, we go back and say that, if there is a pattern which is P_j which is given by an a b c d, then the pattern should actually do two things.

(Refer Slide Time: 40:31)

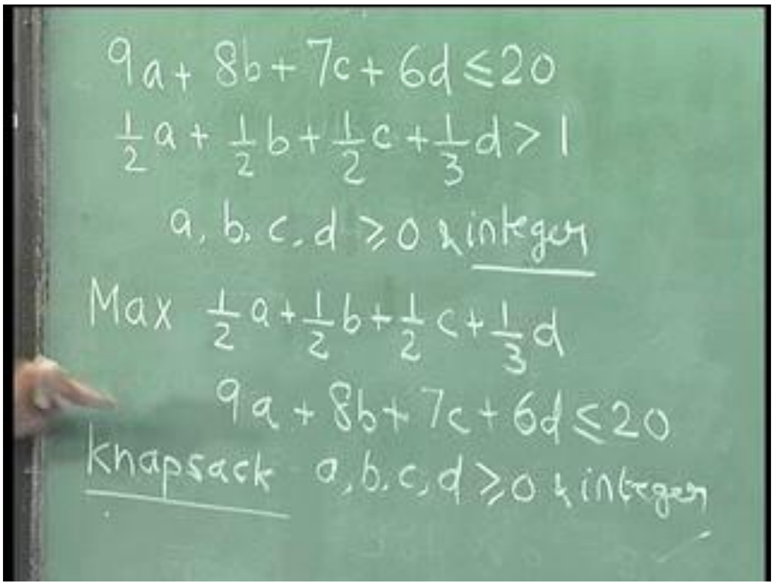


If we have a new pattern a b c d, such that 9 a, plus 8 b, plus 7 c, plus 6 d less than or equal to 20, which makes the pattern feasible and half a, plus half b, plus half c, plus 1 by 3 d greater than 0, then such a pattern can enter the basis. We have to find out the existence of such a pattern. Also we have a, b, c, d greater than or equal to 0 and integer. The integer comes because all the patterns will have only an integer number there. So, 9a plus 8b plus 7c plus 6d is less than or equal to 20; half a plus half b plus half c plus 1 by 3d greater than or equal to 1, because from here we have said that entering pattern will have yp_j greater than 1. So, half a plus half b plus half c plus 1 by 3d is greater than 1 and a, b, c, d greater than or equal to 0 and integer. We have to actually solve this problem to try and find out a, b, c, d such that a, b, c, d satisfies all these three, then such a, b, c, d can enter the basis.

We now move away from the linear programming and then we start concentrating on how to solve this particular problem. Let us take a closer look at this problem. This problem by itself is not a linear programming problem, because of the presence of integer values of a, b, c, d. We cannot relax this, because every pattern should have an integer number of 9, 8, 7 and 6 that we can cut. This problem becomes an integer programming problem within a linear programming problem. We will now try and solve this problem. Another difficulty here is that this has a very strict kind of an inequality, which says it has to be greater than 1, not greater than or equal to 1. In linear programming or integer programming, we are used to greater than or equal to, or less than or equal to, not strictly greater than. The better thing to do is to treat this as a constraint and to treat this as an objective function and say that we are now interested in maximising 1 by 2a plus 1 by 2b plus 1 by 2c plus 1 by 3d, subject to the condition 9a plus 8b plus 7c plus 6d less than or equal to 20; a, b, c, d greater than equal to 0 and integer. We solve this problem and if we say that if the optimum solution to this problem has an objective function value greater than 1, then it means we have found out an a, b, c, d which satisfies all this.

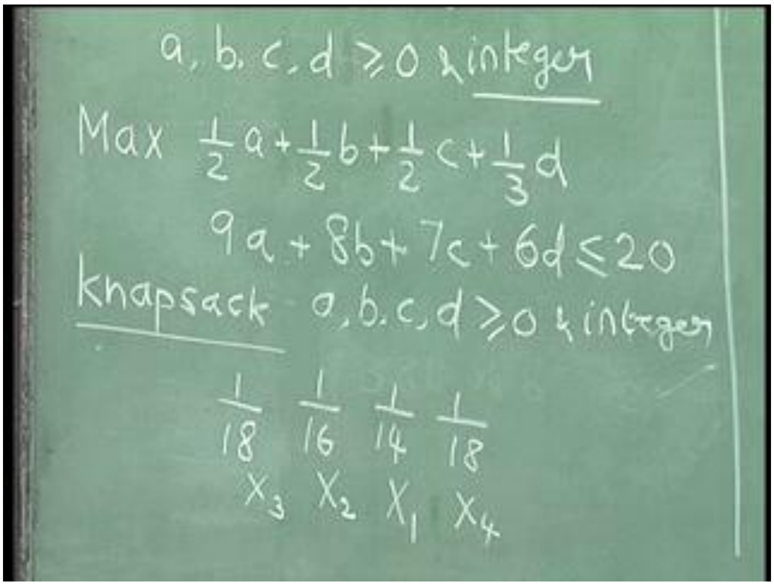
So instead of finding out a solution to a problem containing only inequalities and some restrictions on the variable, we convert it into an optimisation problem, where we have an objective function, we have a constraint, we have a single constraint with the integer and then we say that if the objective function value is greater than 1 then we have found such an a, b, c, d.

(Refer Slide Time: 44:32)



This problem is called a Knapsack problem. This problem can be described as follows: there is somebody who is going for a mountaineering trip or a trekking trip. The person has to choose all the things he or she can put in a sack or in a bag that this person can carry. The single constrained can represent either a weight restriction or a volume restriction on the bag and the objective function will represent some kind of index of usefulness of the items. You want to put those items into the bag, more than one number; it is not a 0 1 Knapsack, it is an integer Knapsack. You can put more than 1 number of that. Correspondingly the usefulness increases, so the problem is a well known problem called a Knapsack problem. There is a very efficient algorithm to solve the single constrained Knapsack problem. We will now look at that algorithm to solve the single constrained Knapsack problem.

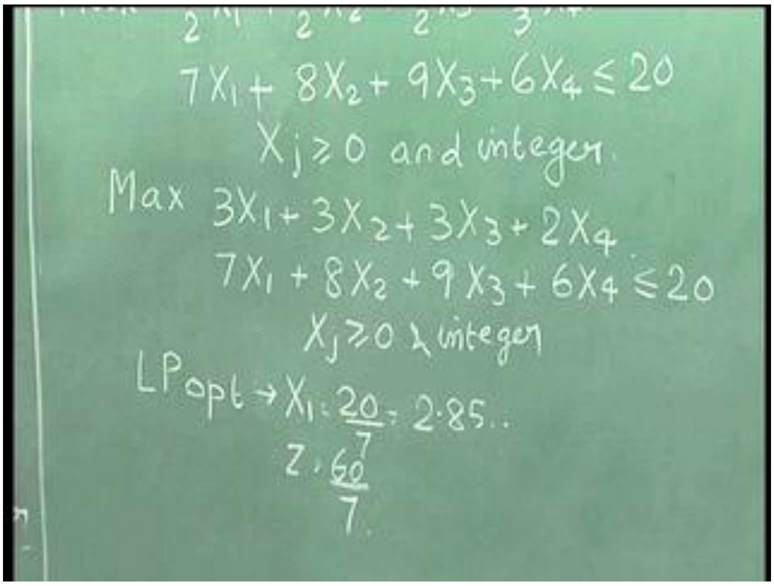
(Refer Slide Time: 46:07)



What we do now is since a, b, c, d are now variables that we have to find out, we will use the notation X_1, X_2, X_3, X_4 instead of a, b, c, d. We will also renumber it depending on a certain computation. The first thing we need to do in a single constrained knapsack problem is this. There is an equivalent between the single constrained knapsack problem and the corresponding single constrained linear programming problem. That is, if you relax this integer restriction, then the problem becomes a single constrained linear programming problem. A single constrained linear programming problem has a very simple solution that, that variable which has the largest ratio of the objective function to the constrained will be in the solution.

What we do now is we are now going to redefine the a, b, c, d, as X_1, X_2, X_3, X_4 in the decreasing order of the ratio of the objective function to the constraint. This is 1 by 18, this is 1 by 16, this is 1 by 14 and this is also 1 by 18. 1 by 18 comes, because 1 by 2 divided by 9 is 1 by 18. 1 by 2 divided by 8 is 1 by 16. 1 by 2 divided by 7 is 1 by 14 and 1 by 3 divided by this is 1 by 18. What we will do is we will number these variables instead of a, b, c, d as X_1, X_2, X_3, and X4 in the decreasing order of this. So, this is the biggest one, so this variable becomes X_1. This is the second one, so this variable becomes X_2. This is the third one which becomes X_3. This is also equal so this becomes X_4. One of these can become X_3 and the other can become X_4. The problem is now rewritten to maximise.

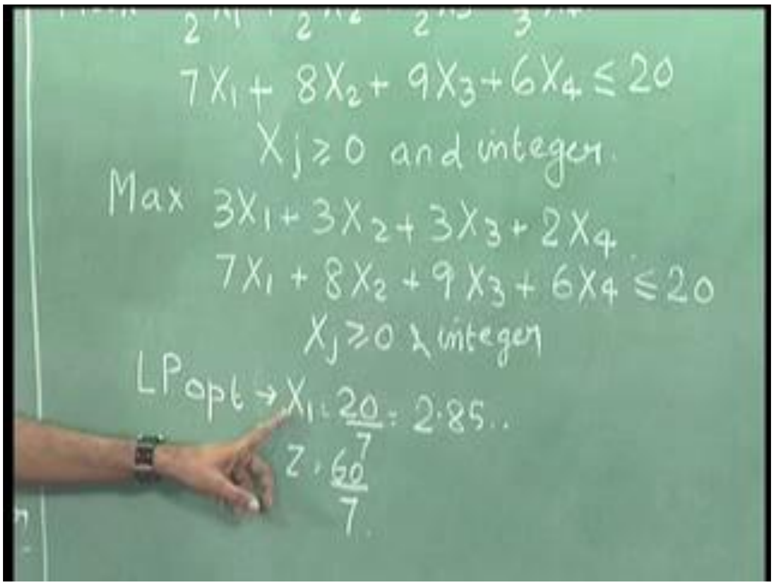
(Refer Slide Time: 48:21)



This is your variable X_1 so maximise half X_1. This is your variable X_2 plus half X_2. This is your variable X_3 plus half X_3 plus 1 by 3 X_4, subject to the condition this is your first variable, so, $7X_1$.This is your second one, $8X_2$ plus $9X_3$ plus $6X_4$ less than or equal to 20; X_j greater than or equal to 0 and integer.

These objective function coefficients are now fractions. Just to make it a little more comfortable, we can bring all of them into integer by simply multiplying by the LCM, which is 6. The problem will now become maximise $3X_1$ plus $3X_2$ plus $3X_3$ plus $2X_4$ subject to $7X_1$ plus $8X_2$ plus $9X_3$ plus $6X_4$ less than or equal to 20; X_j greater than or equal to 0 and integer. Now, for a moment, if we remove this integer restriction, the optimum solution to the problem is given by X_1 equal to 20 by 7 with Z equal to 3 into 20 by 7 which is 60 by 7. So we first take the LP optimum. LP optimum will be X_1 equal to 20 by 7 with Z is equal to 60 by 7. We now go back and say that, so this is Z is equal to 60 by 7. Again we go back to the integer programming, linear programming idea and say that LP optimum is a lower bound and therefore the better lower bound is the upper integer value of 60 by 7 which is 9. So, the lower bound is 60 by 7, but the upper integer value is 9 which is a lower bound to this. From this we can also say that X_1 can take the value 0 or 1 or 2 and cannot take more than that.

(Refer Slide Time: 51:29)

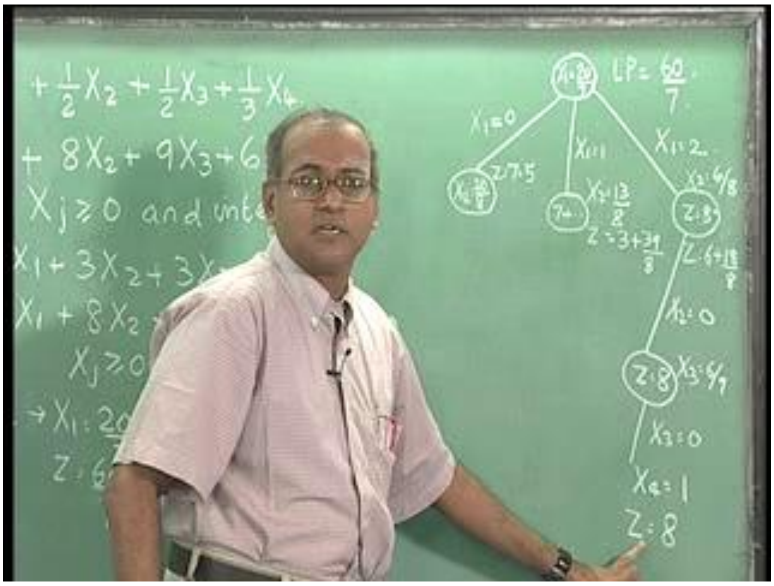


We start with a node here which says, LP is equal to 60 by 7. From this X_1 takes 2.85. So, X_1 is equal to 0. X_1 is equal to 1 and X_1 is equal to 2. Now, when we fix X_1equal to 0, what will happen is, we have removed this. So the LP optimum will now become X_2 equal to 20 by 8 and Z is equal to 60 by 8. So, this is X_1 is equal to 20 by 7. This will give us X_2 is equal to 20 by 8, Z is equal to 60 by 8, which is 7.5. We can only think of a solution with 8 and more from this, whereas here we can think of a solution with 9 and more from this. Problem is a maximisation problem. Since the problem is a maximisation problem, the linear programming is actually an upper bound to this. Therefore, this is 8 point something so we could look at 8 from here. Now this is 7.5 so, from this we could only look at 7 from here and so on. Now when $X_1$is equal to 1 that we have here, now when X_1 is equal to 1, we are going to use up 7 units from here. We have remaining 13 units that are available. So X_2 will be 13 by 8. So X_2 is 13 by 8 and Z is equal to 13 by 8 into 3, which is 39 by 8 plus another 3 that comes from here. So, 3 plus 39 by 8 which is 7 point something. So, a solution of 7 is only possible from here. This is 7 plus something. So the 39 by 8 is 8 times 4 are 32, 8 times 5 are 40, so this is slightly less than 5; so 4 is slightly less than 5 therefore, you get 7 point something.

Now, when you put X_1 equal to 2, the solution that you will have is, now by putting X_1equal to 2, you are using up 14 units here, so 6 units are remaining. X_2 will be 6 by 8 and the Z value will be 6 into 3 18 by 8 plus another 6; so Z is equal to 6 plus 18 by

8. This is 2 plus something, so up to 8 is possible. Z is equal to 8 plus is possible. Here a solution of 7 plus is possible. So only a maximum of 7 can be reached for the IP. Here, it is 7 plus something, a value that is less than 8. So only a maximum of 7 can be reached out of this, whereas here a solution of this is 8 plus, so it may be possible to reach a solution of 8. We try and branch from here. Now this is X_2 equal to 6 by 8. X_2 is strictly less than 1, so the only value that X_2 can take is X_2 equal to 0.

(Refer Slide Time: 57:54)



When X_1 is equal to 2 and X_2 is equal to 0, we use up 14 resources from here. This is 0 and so remaining 6 resources are available. So X_3 will be equal to 6 by 9. The objective function will be 18 by 9, which is 2 plus another 6, so again Z is equal to 8. Again from here a solution of 8 is possible. Remember we are trying to maximise it and each one is acting as some kind of an upper bound; so from here still a solution of 8 is possible, whereas, from here a solution of a maximum of 7 are only possible, if we move down from there. Now since the solution of 8 is possible, we again branch from here. This X_3 being 6 by 9, X_3 cannot take 1. So, X_3 will take value 0. So when X_1 takes 2, X_2 takes 0, X_3 also takes 0, which means we have used up 14 units of resource. This is 0, this is 0, 6 units are available. Therefore, straight away X_4 will take 1 and contribute 2 to the objective function. This will be 2 into 3, 6; so this gives us Z is equal to 8. Now this is an integer feasible solution with X_1 equal to 2, X_2 equal to 0, X_3 equal to 0, X_4 equal to 8, X_4 equal to 1, with Z equal to 8.

This is a feasible solution to the integer programming problem, proceeding from here we cannot get solution with 8; it can only be 7 or less. Therefore, we say that we have reached the optimum solution with Z equal to 8. We have now solved this Knapsack problem and how we use this Knapsack solution and go back to the linear programming problem, we will see in the next lecture.