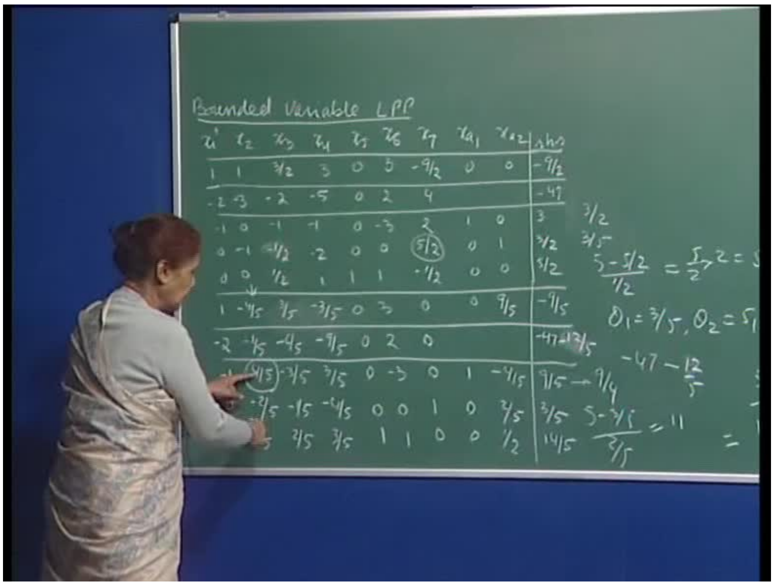
**Linear Programming and it is Extensions**

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

**Lecture No. # 12**

**LPP Bounded Variable Revised Simplex Algorithm Duality Theory Weak Duality Theorem**

(Refer Slide Time: 00:16)




So, let me revisit bounded variable linear programming problems. I will continue with the example that I did last time, because I have a feeling that I did not sort of to show you all the possible pivot steps, so I thought I will complete the example, and so is this was the last table I gave you, and remember we had to do phase 1, phase 2, so this refers to phase 1 objective function, and this refers to phase 2 objective function.

So, we have two artificial variables; so, as we continue, see here this is the first negative, in fact this is the only negative entry in the top row, so x 7 is a candidate for entering the for entering the basis becoming a basic variable, so then we take the ratios.

So, here the ratio would be 3 by 2; here the ratio would be 3 by 5; and for a negative entry remember since the upper bound constraint on each variable is 5, so I will do it as 5 minus 5 by 2 divided by 1 by 2 and this comes out to be 5 by 2 into 2 which is 5; therefore, I choose the minimum, that means, my theta 1 is 3 by 5, theta 2 is 5, and theta 3 is also 5, theta 1 is 3 by 5, because of the positive entries, the ratio this is the minimum ratio.

Then your theta 2 corresponds to the negative entry, which is 5 here; and theta 3 is the upper bound, which is 5, so theta one is the minimum; therefore, I will pivot on this one, so x 7 now becomes a basic variable, and in it is place x a 2 goes out of the basis, so first artificial variable goes out of the basis, I need not make this computation, because remember if an artificial variable leaves the basis at any time then it will not reenter the basis.

You try to figure it out also why, because so in any case I did the computations here for this artificial variable artificial column also there, it was no need to do it fine. Now, you will see that the nu z j minus c j, because we will be making 0s here, both the places, and please check the computations, I will not be very sure, but anyway just make sure that the computations.

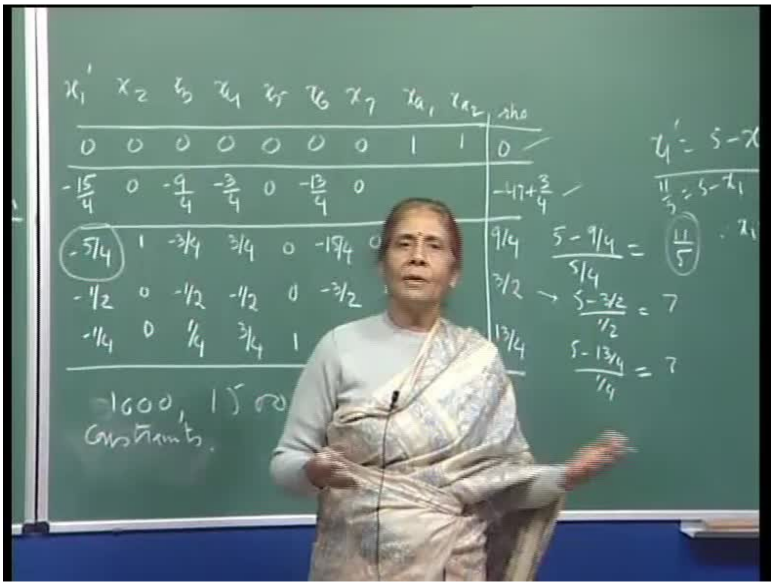
You know the steps, so you have to make 1 here, and 0s, l square, so this is how I do the pivoting, and these are my new right hand side objective function value, this you can also check, maybe we can make the quick check here, see here this is 3 by 5, and to make it 0 here, I have to multiply this by 4 and subtract; so, that means, should be minus 47 minus 4 times we have to subtract, so which means 6 sorry 3 by 5, this becomes 3 by 5, so 3 by 5 into 4, so that is 12 by 5, so you have to multiply this by 4 and subtract from here, so that is 12 by 5 subtractive so.

So, actually this should be minus 12 by 5, this is the new value the objective function, and also make sure that these computations are fine, so this is your next tableau here; and if we we look again at the top row look at the most, if your first negative entry in the top row so this is minus 4 by 5, so this will be a candidate for entering the basis, and here you have one positive entry.

So, this would be 9 by 4, the corresponding ratio is 9 by 4; in this case it will have to be 5 minus 3 by 5 divided by 2 by 5, which you can see is large, 25 minus 3 is 22 22 by 11, so this is 11 then 14 by 5. So, here the corresponding computation will be 5 minus 14 by

5 divided by 1 by 5, so this will turn out to be 25 minus 14, that is 11, that is also 11, therefore your theta 2 is 11, theta 1 is 9 by 4, and theta 3 is 5, so here again you will be pivoting on this 1.

(Refer Slide Time: 05:00)



So, here again this is 4 by 5; so, let me make this as 1 and 0 0s l square, so we do the pivoting, and this is the last column, so make sure that this computation also should be alright; so, you can verify for yourself, the steps here, so anyway this will be your next table, and you see that now this is the end of phase 1, because this value is 0 both the artificial variables.

So, I did not make the computation for this artificial column, you are when this one is entering the basis, it becoming a basic variable this one is becoming non-basic; so, then my the use of artificial variables is over, and I can now start with phase 2, and we see the top row for phase 2 is already there, so I can start from here. Now, just want to make one observation here, you see this can enter the basis now, but you see this is negative, but the corresponding column here is also has all negative entries.

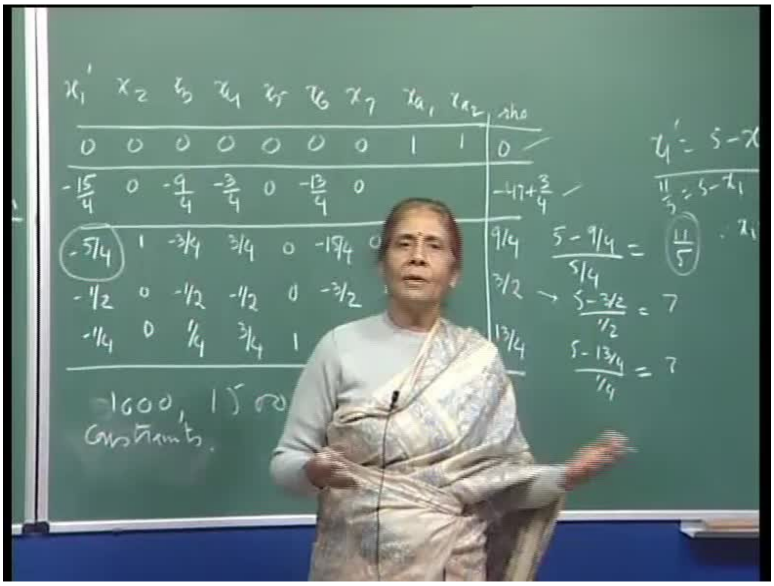
So in a regular simplex algorithm you would have conclude in with the problem is unbounded, but here there is no question of the problem being unbounded, because all your variables are less than or equal to 5, so the variables cannot take very large values here. And in any case remember this is x 1 prime, so you have this relationship that x 1 prime is 5 minus x 1, therefore because we are making this transformation in the table; whenever a variable reaches it is upper bound, therefore this is show in like this; in an any case the bounded variable linear programming algorithm will allows us to continue with the pivoting even when the entries in the correspond the pivoting column are all negative.

So, you know that, so we will do it here, this would be the corresponding ratio will be 5 minus 9 by 4 divided by 5 by 4, so this gives your 20 11 by 5, then this would be 5 minus 3 by 2 divided by half so which is 10 minus 3, 7, 7 this is 7, 10 minus 3, 7, 7 by 2 into 1 by 2, so that becomes 7, and then 5 minus 13 by 4 divided 1 by 4, this is 20 minus 13 that is also 7.

So, this is the minimum ratio, therefore we will pivot on this one; and now, you can continue with the simplex algorithm; so, it that means this variable, when this variable becomes basic x 1 prime is now a candidate for becoming a basic variable, which means that x 1 becomes 0; see, so far x 1 prime was non-basic, so that meant that, your x 1 was at level 5 once x 1 prime becomes basic.

So, it is not that x 1 will become 0, because just see the level at which x 1 prime will come into the basis, this is 11 by 5; so, that means, the corresponding value of x 1 is 1; if x 1 prime is 11 by 5, this is 5 minus x 1, therefore x 1 is 5 minus 11 by 5, which is 25 minus 11, which is 14 by 5, so this is the current value of x; so, remember, when you at the end of the algorithm, when you have optimality with respect to this objective function also you will see all the variables which are at prime level; and correspondingly, use of transformation to get the value of the variable, so this for you can continue with the bounded variable linear programming algorithms; so, I thought that, at least this will give you a left hand sit down; when you look at this algorithm please sit down and check at the entries are all correct or not.

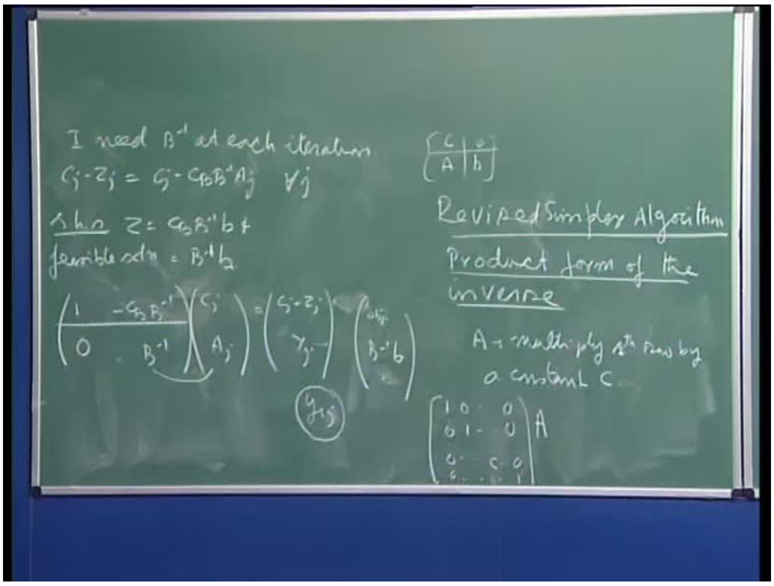
(Refer Slide Time: 08:57)



Now, I want to talk about another aspect of the simplex algorithm, where actually you can…, because you see when the problems are small I can carry the whole tableau along with me, make all the computations, and so no, it does not take much time; but in case you have a problem and which is normally the case, let us say I have thousand variables, and let us say 1500 constraints; even in a computers 1000 variables and 1500 constraints; you do not expect any computer output to have to carry 1500, no I should 4 the other way we should have said that 1500 let say variables.

The number of variables is always more than the number of constraints and you have 1000 variables, 1000 constraints; so, then that means, you have 1500 columns; and if you end up adding artificial variables also then they see the number of variables will may be very large.

(Refer Slide Time: 10:42)




So, your tableau size would be very big, and you cannot glance at it, you cannot look at the whole thing, and there is no point, because you see that computations do not require you to actually. So, let us let us revisit the simplex algorithm; what we are trying to say is that, you need the quantities we need or basically I need B inverse at each iteration.

Once you have B inverse, then I can immediately compute my C j minus Z j which is equal to C j minus C B B inverse A j; so, if I have my original tableau, I have the C j(s) and A j(s), so I can compute this for all j; and I need the right hand side, your the value of the objective function is C B b inverse b and your feasible solution basic feasible solution is given by B inverse b.

So, these are the…, so all this is computable provided, I have my current B inverse, that means, I need to update my B inverse that each iteration; and I showed you that if you carry this, then this becomes…, say remember this was the extended basis; I is m plus 1 by m plus 1 that I am carrying at each iteration, and I updated, and how do I updated the idea behind, it is at…, and so if I have this and I have my tableau.

So, that means, if you have your C entered somewhere, you have your A, and you have your B; so, if I have this stored somewhere in my computer, the starting value you can say is 0; if I have this stored somewhere in my computer, then what I can do is, I can recall a column from here, a non- basic column and if I multiply this by C j A j just the top row.

So, if I keep this extended basis inverse somewhere in fact I do not even have to store this, so I just keep this extended this somewhere, and I have my original problem stored in the computer, I will recall, I start we calling the first non-basic variable second, and so on, so I multiply the top row with this and that gives me C j minus Z j.

So, I will keep multiplying the top row with my non-basic columns, and the moment I hit j for which this is less than 0, I know that this is my pivot by the bland's rule, I know that this is my entering column, this has to become a basic column, and then I will do the this computation, so I will then compute Y j.

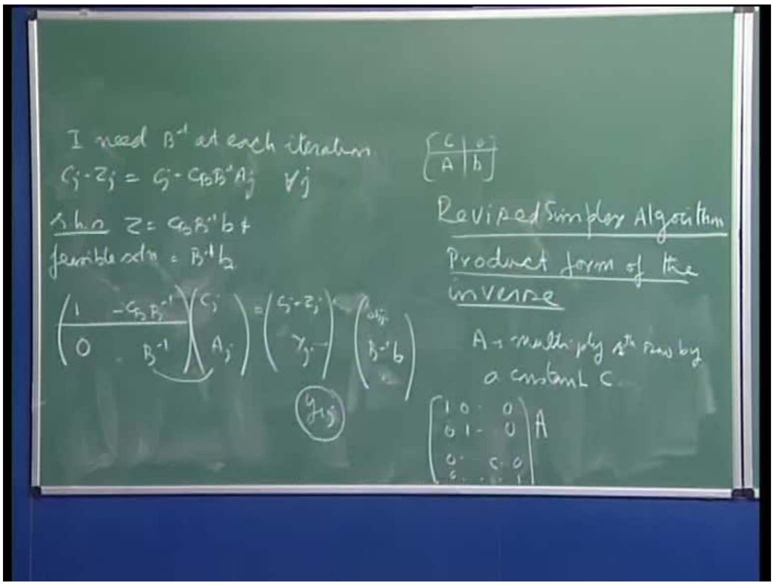
So, only for the column which has to enter which has to become basic, I do this computation, I do not have to do it for all others; and then I do this, and from here I also have seen I can multiply B inverse with B, so I will have the right hand side vector here B inverse B, so suppose you have this, and your objective function value here objective function.

So, you have this B inverse B, so then I will have this columns, so I decide the pivoting element, so I will start taking the ratios, and the moment I get the correspond, so that means, for all positive entries here, I will take the corresponding ratios, and I get the element y r s. now I will.

This is y r j, so this will be my pivot element; once this is my pivot element, I can just pivot on this column, so that means, I keep it this aside, I now pivot on the y r j, and make a 1, there 0 all where; and similarly, remember I do the same thing with this; the row operations I do with this, and I do the row operations with this right hand side column; once I do that, then I get my new basis inverse and I can.

So, you see what is saving, so this is what I am describing to you is the revised simplex algorithm; so, the revised simplex algorithm the idea is, and this is how a computer handles this, so we can store the original problem, we can keep updating our basis inverse at each iteration, and that we know how we do it; we do it by selecting the column which has to become non-basic, which has to become basic, then by taking the ratios I can find out the pivoting element.

(Refer Slide Time: 15:32)




Once I know the pivoting element, then I can pivot here, I do the same computations here, and here, and I get my new basis inverse, new right hand side feasible solution and the objective function value; and then I can continue again, I just take the top row, and the new revise new basis extended basis inverse and then I multiplied by each non-basic column, and the moment I hit the negative C j minus Z j I to selected for pivoting right.

So, you can see that the amount of saving in space computations is tremendous and that is why one can handle large size linear programming problem, but even this kind of computations became too…, when the when this transatlantic problem had to be solved the communication problem had to be solved, this size became even beyond then best computers available at that time.

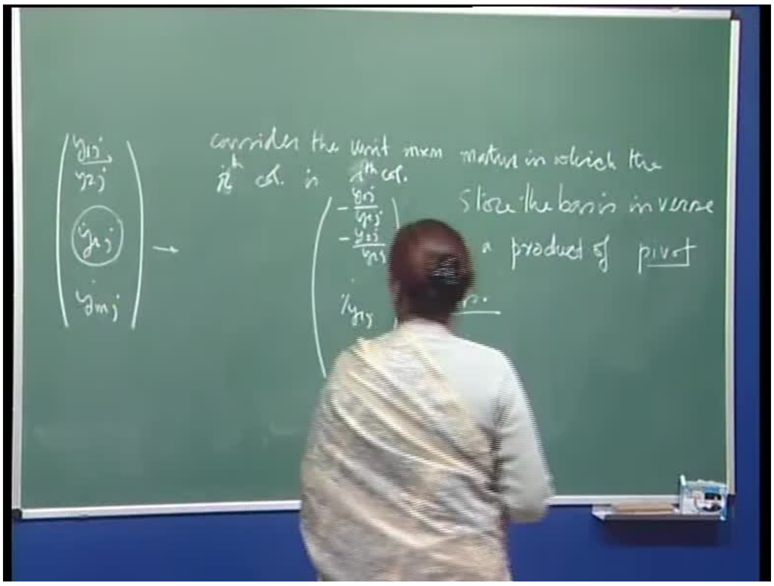
So, we will come to that at the end of the course when I can spend some time on the alternatives to solving linear programming alternatives to the simplex algorithm we will talk about. But current in certainly the program that are now written for the simplex algorithm only use the revised simplex algorithm, because we do not need to convert the whole tableau at each iteration, I do not need that information right.

I just need to know which is the column, which has to enter the basis; and then I just pivot on that column, and I need to update my basis inverse that is all; once I have the new basis inverse then I can compute all the required quantities to continue with the simplex algorithm and yes.

Now, another thing is that even this becomes too big you store every time; so, what they do is, they use the…, let me explain and I have try to the slope product form of the inverse; now, here again this is same idea that I had introduced in the beginning; when I told you that, when you do any row operations on a matrix, for example, if you take a you take a matrix A I want to multiply rth row by a constant C for example.

So, I just multiply want to multiply the rth row of this matrix by the constant C, what does it mean? I can also multiply the corresponding see that means take the elementary matrix 1 0 0 0 1 0, and then the rth row you will have a C here 0 0 0 1; so, this is my elementary matrix which I obtain from the unit matrix by multiplying the rth row by C. So, then this will be my elementary matrix and this when I pre-multiply the matrix a by this matrix, I get the corresponding result, so this we used earlier also then the same idea I am going to continue with here.

(Refer Slide Time: 19:08)




So, when your pivoting on y or the column y j, for example, what do we do, I have with the entries are y 1 j, y 2 j, y r j, and y m j, this is my y j column, and this is my pivot entry, so what will I do I divide by y r j, so if I right. Now, see that means, a consider the unit matrix unit m by m matrix in which the jth column is now, what is the jth column? The jth column will be it will be 1 by y r j.

So, I am just writing the jth column; in the unit matrix I just replace the jth column by whole down in which the rth column, so in which the rth column is…, so this is the rth now, because rth column will have a one here, so this is rth column 1 1 I j, then for example I want to make a 0 here. So, what do I do, I multiply the rth row by y 1 j and subtract, so it this will be minus y 1 j upon y r j, next entry we will be y 2 j upon y r j, and so on, and the last entry here would be minus y m j upon y 1 j; so, the unit matrix in which the rth column has been replaced by this.

Now, if I pre-multiply, so that means, I pre-multiply the current current basis inverse by that matrix in which the rth column of the unit matrix has been replaced by this, because this is also pivoting that you do; you multiply the first row by y 1 j and subtract, so the first, that means, you multiply this row by y 1 j and subtract, when you where you have divided this entry by y r j. So, y 1 j, and you subtract so this 0 all the pivoting element a pivoting operations that you do, so I replace the unit rth column of the unit matrix by this; and then if I premultiply the current basis inverse by that matrix I will get the new basis inverse, and I have not done the calculations for this also, but the same thing will apply.

So, that means, you can instead of keeping the whole thing like this, the B inverse essentially I am just concentrating on B inverse; if instead of keeping this, we can keep it as a product of…, so every time at each iteration you have this column, and that this is the rth column of the unit matrix.

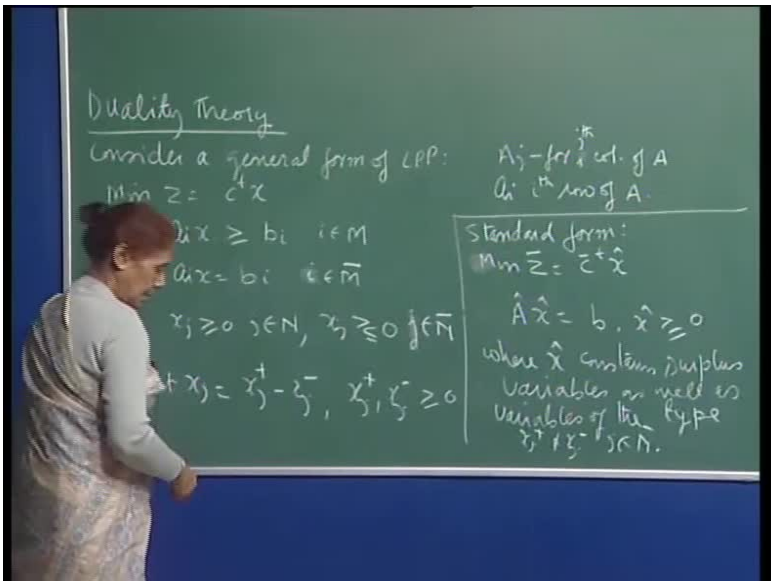
So, then I can immediately write down by corresponding matrix that I need to premultiply with, so in which the rth column is this; so, you can store the basis inverse as a product of I should what may be I will just point a new word here inverse as a product of iteration matrices something or pivot matrices are pivot matrix as product of pivot columns. So, this is my own term, may be let us say, but you understand that I can just store the inverse as a product of this, and so when I needed I can simply multiply them all and then get my B inverse then get all my other quantities and continue with the algorithms.

So, that means, now when you get a output from a given from a any simplex algorithm, you will understand that it will give you the basis inverse, it may give you in this form may be it may m plus 1 by m plus 1, then it will give you the right hand side with, so you can read the basic feasible solution and that is it. So, you can continue with…, so this is a definite saving, it makes a quicker, because you are not computing your number of computations has gone down drastically, and you require less space also it allows you to solve real large problems by the simplex algorithm.

So, this is where we end with the revised simplex algorithm. So, we will continue revisiting; and of course, you can modify this when you are applying this to the bounded variable problem; or if you have phase 1, phase 2, then also you will be having two objective function, two such rows here, and you can do the same thing; and you can simply modified once what is where and why, once you understand that, that you can you can modified.

Papadimitriou, in his book says that linear programming theory is very interesting by itself and you have seen how versatile in fact I have not seen which shown you so many other aspects, which we will see now of the simplex algorithm, but and so that could have been enough the linear programming theory by itself would have been enough but.

(Refer Slide Time: 25:22)



Now, I show you another interesting aspect of a linear programming theory and that is duality. So, let us just first start with…, consider a general form of the LPP of a linear programming problem; and the general form would be where I say that this is minimize Z equal to c transpose x subject to a i x a i x greater than or equal to b b I, I will explain in a minute, where i belongs to M.

So, in equality constraints of the greater kind, and here see the notation is I have been using A j for a column of A, and when I do this, this is the Ith row, for for jth column, here this is for jth column, here this is for jth column of a and a i is the Ith row of a…, so I suppose, but not be confusing the small a refers to the row, therefore it is a row form; therefore, I am writing this as a i x greater than or equal to b i i M and a i x equal to b i for i belonging to M bar, and we say that x j is greater than or equal to 0 for j belonging to N and x j 0 for j belonging to N bar.

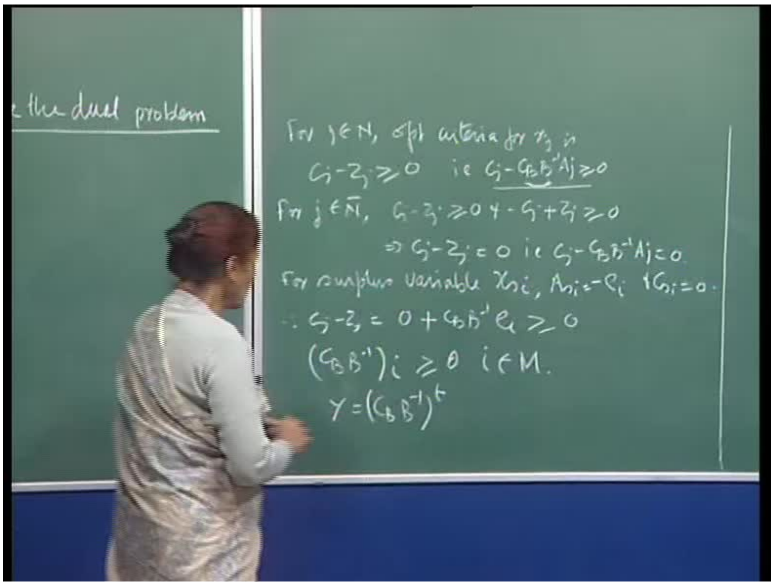
That means, some variables are restricted to be non-negative, some variables are unrestricted, some constraints are inequality constraints, and some constraints are equality constraints, so this will be a general form of the of a linear programming problem; I had earlier showed you that we can convert all linear programming problem to the standard form by either adding slack variables are surplus variables. So, let us do that. So, when I write the standard form, now the standard form would be standard form would be minimize to the standard form what they need to do, because I in the standard form all variables are non-negative, so here we can write for j belonging to N bar.

Let for j belonging to N bar, let x j be written as x j plus minus x j minus, where x j plus and x j minus or greater than or equal to 0; so, this is always possible any number can be written as a difference of two non-negative numbers, so we will replace all x j(s) which are unrestricted in their sign by this this expression x j plus minus x j minus and where x j plus and x j minus are all non-negative. So, what will happen? Say, for example, for a coefficients here, for x j plus will be plus c j, for x j minus will be minus c j; and similarly here the columns the column corresponding to x j the column a j.

Now, you will have a column a j appearing here, and a minus a j also appearing in the constraints; so, standard form therefore would be minimize, may be you can say this is Z bar which is C bar transpose x, because now your C bar has changed, and may be you can see that this is x hat subject to a i or may be in this case I will not use the same this thing; what we can say is, we will simply say that A hat x hat is equal to your b will not change the right hand side does not change, because here I would have added surplus variables minus x as j and you will have this thing, so the right hand side remains the same; and now, you have that this thing that x hat greater than is equal to 0, where x hat contains surplus variables surplus variables as well as variables of the type variables of the type x j plus and x j minus for j belonging N.

So, there x hat is your new set of variables, and this will be your standard form of the linear programming problem. Now, let us look at the optimality conditions for the for the new formulation. So, here for example, what would be the condition.

(Refer Slide Time: 31:11)




Now, you can consider first variable where j belongs to n; so, for these variables for j belonging to N, optimality criteria for x j is C j minus Z j greater than or equal to 0, that is C j minus C B B inverse a j is greater than or equal to 0 for all for j belonging to N this is my optimality criteria.

So, I will just write them down and let us look at this. Now, for j belonging to N bar, remember for j belonging to N bar, you are going to have two columns A j and minus A j, and so for j belonging to N bar my 2 conditions would be I will have C j minus Z j greater than or equal to 0 and minus C j remember minus C j and the Z j, because Z j is C C B B inverse A j right Z j is C B B inverse A j.

So, now, for x j plus it is going to be C j minus Zj for x j minus it will be minus A j, so that become…, and this will be minus C j, this is plus z j same quantity, because the column is the same if the sign which is the is greater than or equal to 0, so what does this imply these two together imply that C j minus Z j is 0. So, for all j belonging to N bar my optimality condition is that C j minus Z j is 0, that is, C j minus C B B inverse A j is 0. Now, consider for surplus variable x s i; what is the corresponding column? Your A as i is nothing but e I, because you have added it to the ith constraint here, so for i in m you added a surplus variable here.

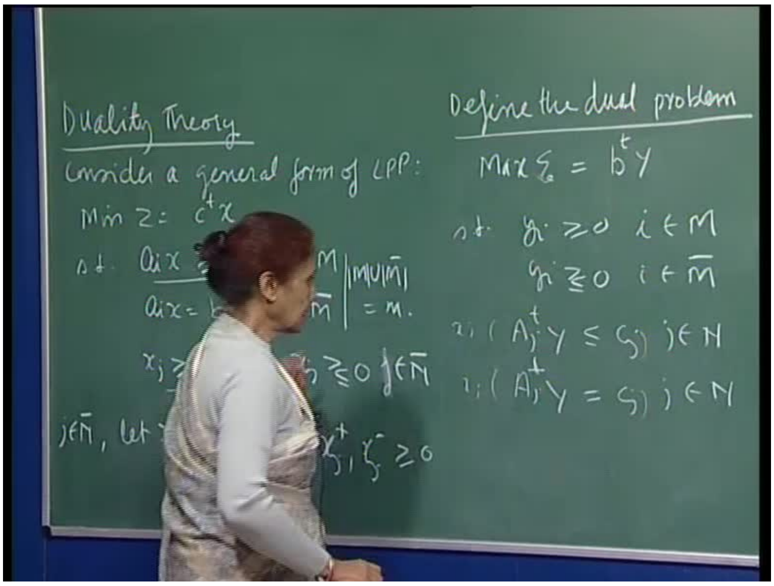
So, the corresponding column is minus e i is minus e I, therefore and your c s i is 0 right, because it is a surplus variable; therefore, C j minus Z j is equal to 0 minus Z j would be what C B B inverse e i and plus. So this should be greater than or equal to 0, is it okay, where surplus variables have I done it rightly your, because we are writing greater than kind so it will be minus e I, that is why it has to be minus e i we are add a surplus variable, so this becomes C B B inverse e i, which means that the vector C B B inverse i is greater than or equal to 0, i belonging to M.

So, the components here, that means, suppose i for all i in M I have added a surplus variable here, therefore the corresponding components of C B B inverse I or greater than or equal to 0 when I belongs to M; when I cannot say anything for i in M bar for the sign of C B B inverse; I cannot say anything the size of the components of C B B inverse; see remember this is an M dimensional vector, and yes I should have said here that M union M bar what is called the cardinality is M, that is (( )) is there, so by total number of variables was N here, that means, N union N bar cardinality of N plus cardinality of N bar is N cardinality of M plus cardinality of M bar is small m.

So, those dimensions are still there, and I am assuming that…, and so here for all i in M the corresponding component C B B inverse must be non-negative; so, once you have this, and you see that this is what they satisfy for j in N, and for j in N bar the condition is that it must be 0, this must be 0.

So, now, let me define the dual problem or what would be best is let I write it here, yes I think that will be best; so, let me now write the dual problem here, so that you can then match the things; so, what I will do is, I will write my dual variable Y the column as B inverse; now, if you are see this is the convention here, I am doing it this is a row vector, so if you then you will have to write this and with the so that it become so that y all are so keep it say for was simpler Y is a column, therefore this is C B B inverse transpose, because this is a row I am writing this as a row vector, so when you take the transpose it becomes a column vector.

(Refer Slide Time: 37:30)




So, y C B B inverse transpose; if I do this, I define the dual, so define the dual problem; see remember in the primal problem, we try to find basis B, which will give me an optimal solution, that means, for which the optimality conditions will be satisfied; and now, I divide the other side of the problem, that is, now I am saying define the dual problem, such that, you are maximizing what is the this thing, we can write this as psi equal to s b transpose b transpose Y subject to let me write it down, and so here this was for I belonging to M remember.

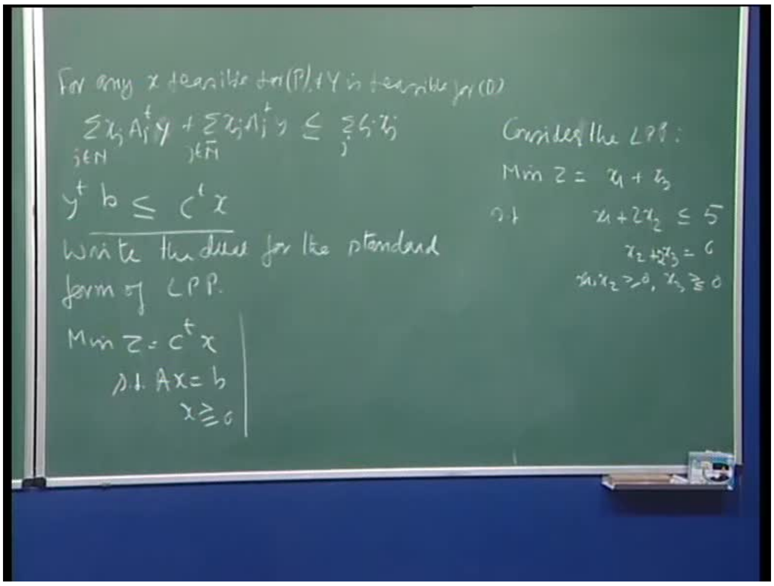
So, now, I will try to define the dual problem keeping these optimality conditions in mind and let us how will define the dual; that means, so you can see that the prime primal problem is concerned with feasibility maintaining feasibility and improving the objective function value; here the primary concerned of the dual problem will be to satisfy the optimality condition, this is the idea.

So, maximize this subject to…, so remember here this is subject to we will say that for I subject to y i greater than or equal to 0, i belonging to M corresponding to this; when corresponding to this y i for i belonging to M bar, because we are not able to say anything about the sign of the corresponding dual variable when I is in M bar; and then for x j non-negative, what do I have here, j belonging to N.

The condition is that, so here this will be you can write this as see this is C B B inverse A j the single numbers so I can write it as the x transpose so that will become A j transpose y this would be less than or equal to C j; see here if you write this, this quantity is less than or equal to C j that is your constraint for j belonging to N, so that corresponds to this; and for when this is unrestricted when your j is unrestricted, then we got this condition that is A j A j transpose Y is equal to C j for j belonging to N.

So, this is your dual problem. Now, I we can also do now, I mean, you can and let see you can immediately write down, so once I have this a dual problem, let us see what kind of information we can extract from it, and what does it how can we use it. See here If you look at this, and I because my for j in n x j is non-negative; so, if I multiply this constraint by x j and add for all j belonging to N, the inequality does not change; and then here because this is equality and the sign of x j is in material, this is unrestricted, so the equality will not change.

(Refer Slide Time: 40:48)




So, let me multiply this by x j and also this by x j, and add what do I get, so this will give me x j A j transpose a Y summation j belonging to N plus summation x j A j transpose Y j belonging to N bar, this will give me less than or equal to because this contributes to less than or equal to psi less than or equal to c j x j. So, that means, now I will add for any x feasible for P; let me write this problem as my primal problem p, and this as my dual problem d, so we will always refer to P D as the primal dual pair, so is feasible for P and Y is feasible for d so for D.

So, I am assuming that I have a y satisfying this set of constraints, I have an x satisfying this set of constraints, then you have c j x j; and here now you can rewrite this as again, see this is the single, this will be see how will you read this, when you sum up as x j A j transpose y, so this is actually summation x j A j transpose Y j belonging to your N union N bar.

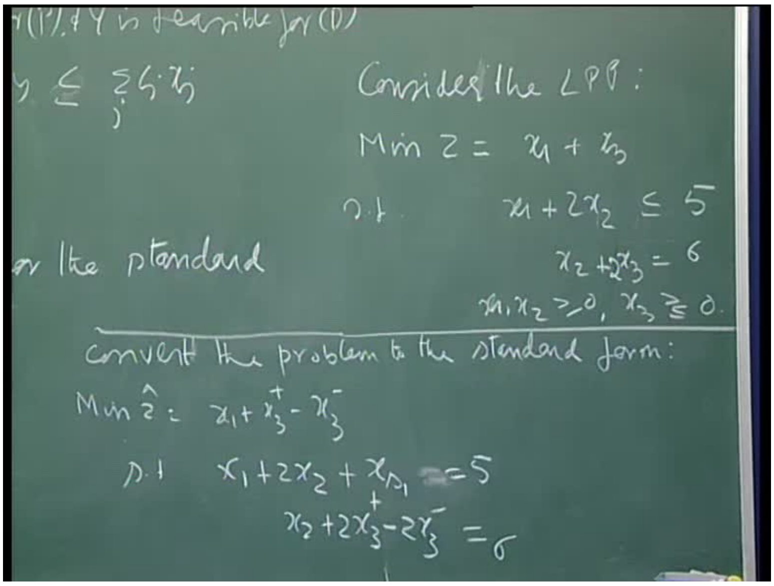
So, this I can rewrite as take the whole this thing this will be Y transpose, see this is the single number, so I can take a transpose and not change, so Y transpose A j x j and thinking of a transpose of this whole thing, so this will…, and remember A j x j is greater than or equal to see the these constraints you can also write as A x greater than or equal to b. So, this whole thing becomes a x, you are summing over j, because y transpose I can take outside; remember, here Y I can take outside, then I am summing over this, which is the same thing as say summing A j x j, so which is greater than or equal to b; so, this whole thing is greater than or equal to y transpose b, and then this is less than or equal to c transpose remember this is over k.

So, this is your c transpose x, this whole quantity, because you have the quality constraints and greater than kind constraints, so this whole thing is greater than or equal to Y b, and so this becomes Y transpose b it is less than or equal to c transpose; so, this is the first lemma of the of a duality theory, which says that, if you have a feasible solution for the primal, and you have a feasible solution for the dual, then the objective function value for the dual will always be less than or equal to the corresponding objective function value for the primal; so, take any primal dual feasible pair you will have this inequality; so in other words what we are showing is that, this is a maximization problem, this is the minimization problem.

So, the max objective function value will always be less than or equal to the min objective function value for the primal. And now, I will as an exercise ask you to write in the dual problem, so write the dual problem, it is the simple exercise, write the dual for the standard form of for the standard form of LPP, which is minimize Z equal to c transpose x subject to A x equal to b x greater than or equal to 0. So, please do it and we can check it? In the next lecture, I can give you these and because most of the time I will be I just took the general form to demonstrate to you how the dual problem can be formulated in the most general case.

Now, let us just take up an example here; we will just demonstrate to so consider this problem; consider the LPP minimize Z equal to x 1 plus x 3 subject to x 1 plus 2 x 2 less than or equal to 5, then x 2 plus x 3 or 2 x 3 2 x 3 is equal to 6 x 1 x 2 greater than or equal to 0 x 3 unrestricted.

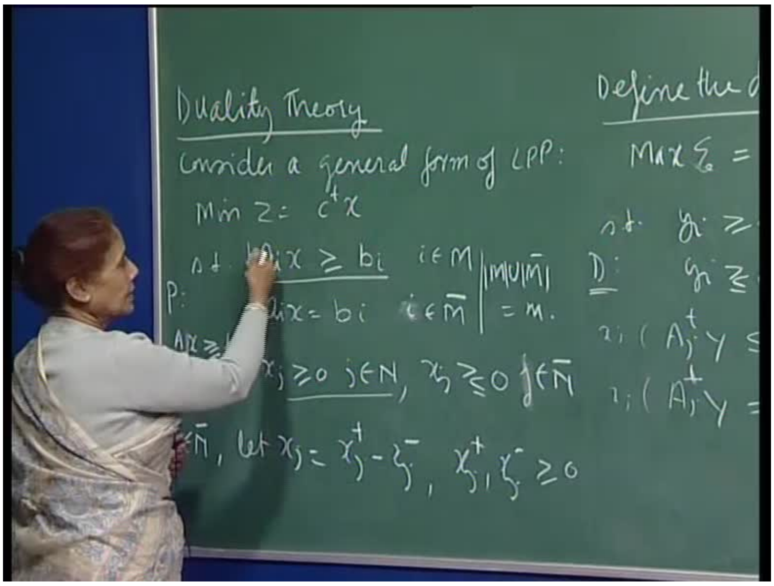
(Refer Slide Time: 46:35)




Suppose, you have linear programming problem in this form, so I have deliberately chosen it will different from the form I stated; let us see does not matter know if you have a greater kind inequality, if you have a less kind, then you can multiply the whole thing by minus sign and you can get rid of the greater kind.

So all these things are easily manipulated no problem; so, anyway the best thing would be, see we can do it two ways; so, as I told you, we convert it to convert the problem to the standard form, which would mean that, for x 3 I will have x 3 plus and x 3 minus, so this will become minimize z hat equal to x 1 plus x 3 plus minus x 3 minus, for x 3 it sub restrictive, so I will substitute, I mean, substitute it by this expression; then here we will have to added slack variable, so this will be subject to x 1 plus 2 x 2 plus x s 1 equal to 5; and then you will have x 2 plus twice x 3 plus minus twice x 3 minus equal to 6; and now, all the variables are non-negative, x 1, x 2, x 3 plus, x 3 minus, x s 1, all the variables are non-negative.

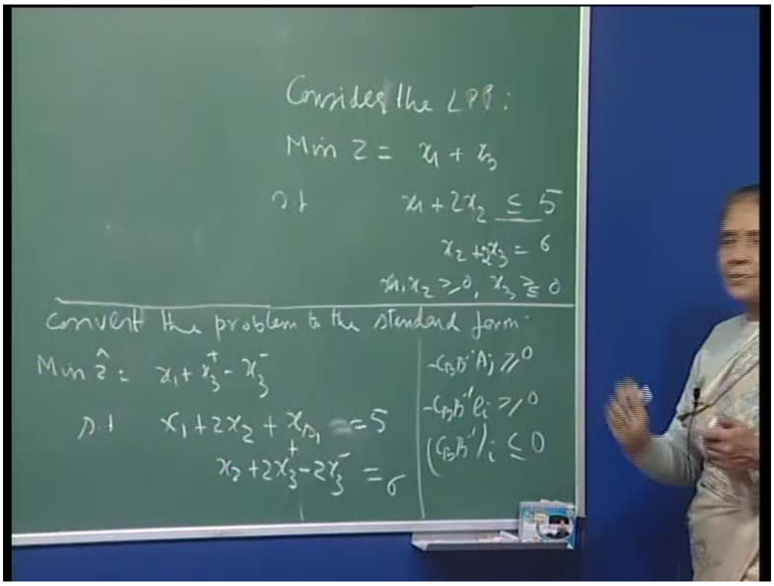
(Refer Slide Time: 48:07)




But the you want to follow the rule here, and then write the dual, let us do that simply, yes or if you want I can derive it from see the…; only thing is that, here remember this constraint is less kind, I gave you the rule for writing the dual problem, then your constraints or either greater or less.

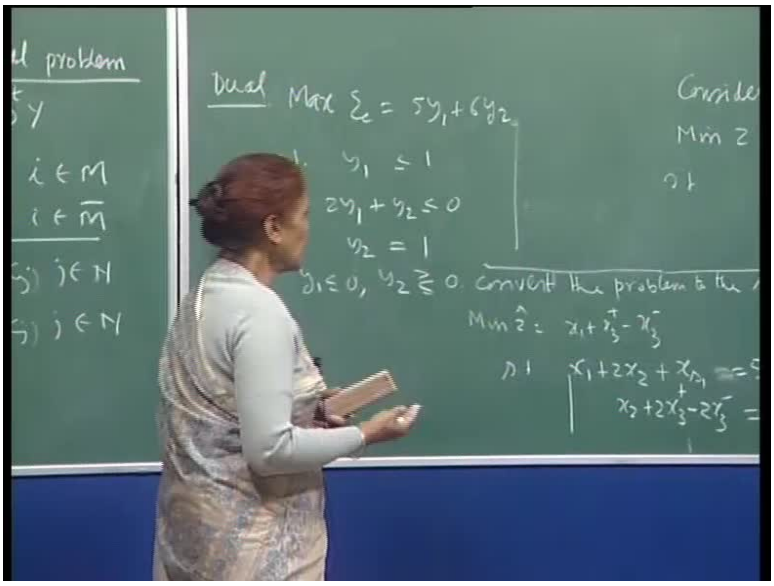
So, what would you want to do? See, if I can convert this 2 minus type also minus a i x less than or equal to minus b i, but how will that change; one can guess and say that the corresponding variable y i will be less than or equal to 0, why because remember see the simple how did I derive this condition y I greater than or equal to 0; the condition I derived was from because for the surplus variable the corresponding coefficient in the objective function was 0 and the column was minus e i.

(Refer Slide Time: 49:22)




Now, when I add slack variable the column is plus e i; so, and the coefficient is 0, so you have minus C B B inverse A j greater than or equal to 0, and A j is you also minus C B B inverse, and A j is your e I, so this is greater than or equal to 0, which means that the corresponding component ith component is less than 0.

(Refer Slide Time: 50:05)




So, when you have for a minimization problem, if you have less than or equal to constraint the corresponding dual variable will have a negative sign less than or equal to 0, so that is it, so that is all you have to remember; so, in that case, now you can write the dual quickly, so this will be maximize psi equal to…, and so I will write it and then I want you to sit down in so this will be 5, the right hand side becomes 5 y 1 plus 6 y 2 subject to remember the columns become the rows transpose.

So, particular column here this will become…, so this will be y 1 this only entry here, and y 1 would be see this is less than or equal to less than or equal to 1; then the second column 2 y 1 plus y 2 1 plus y 2 this will be less than or equal to less than or equal to the corresponding coefficient 0 here, y 2 is not appearing here; and then for the for the unrestricted thing I had equality constraint.

So, the corresponding column for this thing, for x 3, the corresponding column for x 3 is simply 0 2, so 0 2, so that means, y 2 is equal to in this case it will be 1, and y 1 less than or equal to 0, y 2 unrestricted; for equality constraint your corresponding variable is unrestricted, which you have so a second constraint is unrestricted; so, actually now you see that the correspondence is the dual variable corresponds to a primal constraint. So, y 2 corresponds to this constraint, y 1 corresponds to this constraint, since this is of the wrong kind, so the wrong kind in the sense that I have formulated it for greater, so there the y corresponding variable was non-negative here, it is less so the corresponding variable is less than or equal to 0, so this is your pole.

So, in the exercise, in the assignment sheet and also while you are going through this material you should sit down and write down the dual learn to be very familiar, how you immediate.., how you write down the dual problem corresponding to any form of the LPP you can either converted to the standard form, and then as I told you do it as an exercise write the dual for the standard problem, we will revisit it in the next lecture, and try to show you how in the standard form also you can immediately write the dual.

So once you have this, now we will have very interesting interpretations for the dual problem, for the dual variables, and especially they are lot of economic interpretations of the dual variables also, so this adds the richness of the linear programming theory. So, I will continue with this in the next two lectures.