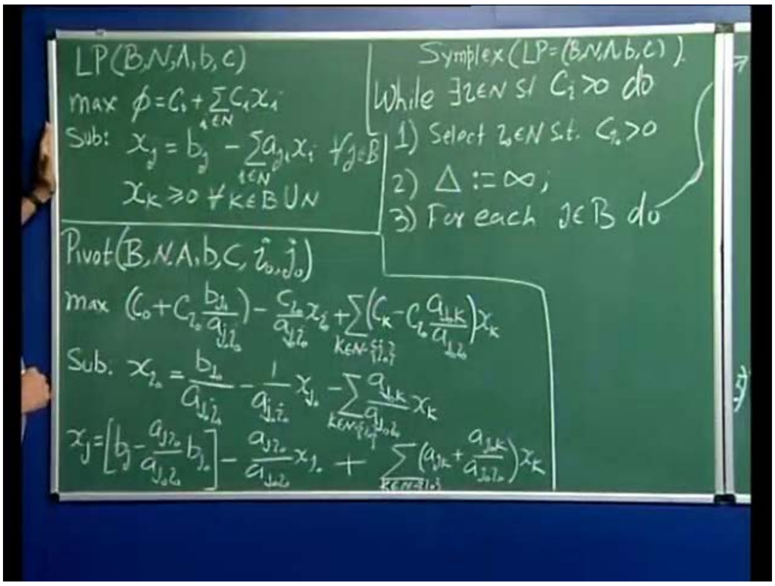
Computer Algorithms-2

Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Science, Kanpur

Lecture - 28
Linear Programming II

(Refer Slide Time: 00:28)



Hello. So, we will continue with our discussion on Linear Programming to remind ourselves, let me just quickly run through what we had discussed. The definition of linear program in was 2 index sets, one is the set of variables in the, and the other is the set of equations or inequalities, this is a matrix of a coefficients of conditions. And these are true vectors, the expression that we want to maximize, which is devoted by phi here, is this linear expression where this C 0 and C i's are given in this vector. Subject 2, these equations where x j is suppose to the b j minus a j i x i for each j in the, every variable those in B as well as in N are suppose to be non negative in the solution.

In the last lecture we had described pivoting, which is transferring one variables from N to B, and one variable from B to N. So, this is a variable from N this is a variable for from B, and what we did, is we choose the j 0'th equation from the equation set and this equation should be size that a j 0 i 0 is non-zero. So, the coefficient a j naught i naught is non-zero, we take x i naught on this side and we get x i naught which is same equation looks like this.

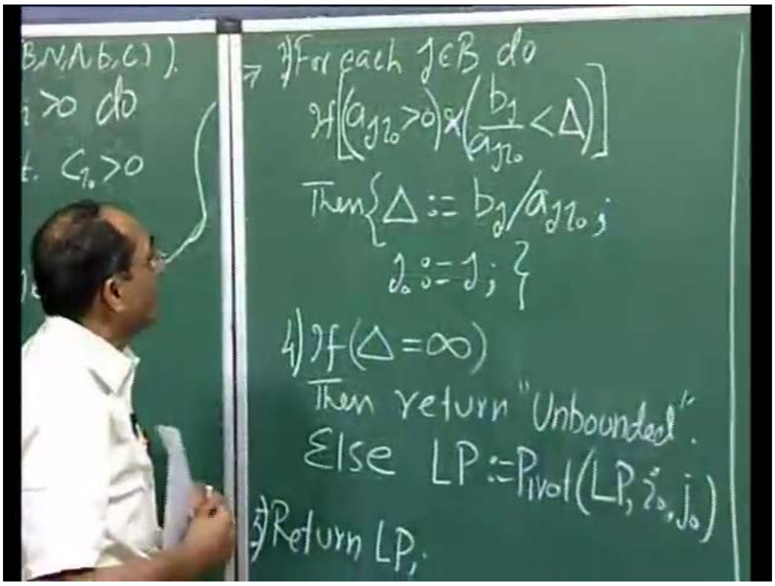
And since, there was a j naught i naught here we had to divide the equation by a j naught i naught. So, that one equation corresponding to j naught is rewritten here, the remaining equations, we wrote this way all we have done there is that occurrence of x i naught was replaced by the right hand side. So, we get a new appearance of the same equation and the same thing we did for the object function the x i naught was substituted by this expression and we get this expression for the object function.

So, we got to remember that this is a new linear program, but it is completely equivalent to the old one, in this sense that every solution or rather any variable assignment that satisfies these conditions, will satisfy these conditions and vice versa. Any set assignments to the variables that satisfies these conditions, will do the same here. Now, based on our discussion from the last lecture, we have a an algorithm that we are going to describe now, and that called simplest algorithm.

Now, here I am without mentioning I am assuming, that this algorithm is meaningful only when the basic solution of the linear program is feasible, recollect that the basic solution is one in which we set all the N variables to 0. So, these are all 0, these are all 0 and the d variables even the x j are equal to b j and basic solution is feasible, if all these b j's are non negative because that is our requirement here. And in the basic solution the object function takes value c naught because this is gone.

Now, subject to this condition, that the input linear program has feasible basic solution. We get into this while loop, what we do is we check whether there is some C i which is positive. If it is positive, if we have at least one such i and we select such i lets say it is i naught, so we have selected an i naught that C i naught is positive. Now, the next steps are essentially to select the corresponding j naught subject to a condition that we had discussed last time, so let us just go through that. We initialize a variable delta to infinity then the third step says the following.

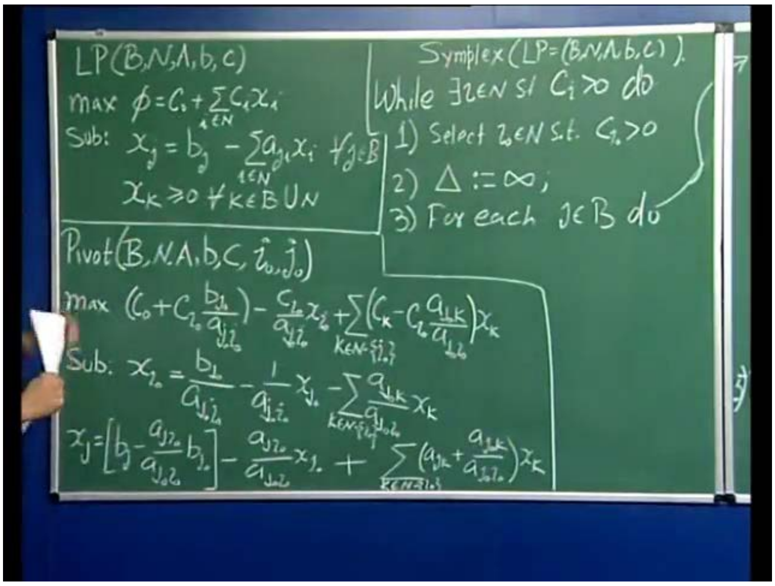
(Refer Slide Time: 05:31)



If a j i naught, so for every j we are checking this condition, if this coefficient is positive and if b j by a j i naught is less than delta then is set delta to that value and j naught to be j. Now, what exactly this dealt us for us, this step checks out whether there is at least one j says that a j i naught is positive, if all of them are 0 or negative then this variable delta remains infinite because we never executive this way.

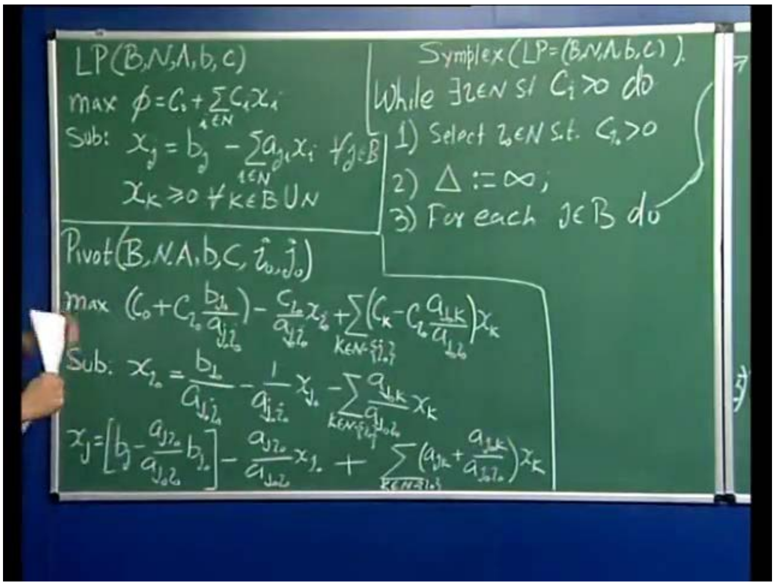
If at least one of them is positive then j naught will be set to that value, subsequently of there are several j variable have such that a j i naught is positive, then we are going to select that j naught for which this quantity is the least and that least value has been set, into this delta. Finally, all we are doing is we checking whether delta is infinity that is to say, we never found any j says that a j i naught is positive in that case we return unbounded, else we perform the pivot operation. So, we find the new value of LP to be the old value of LP and we perform pivot with respect to i naught and j. Now, let us try to understand what this algorithm is doing.

(Refer Slide Time: 07:31)



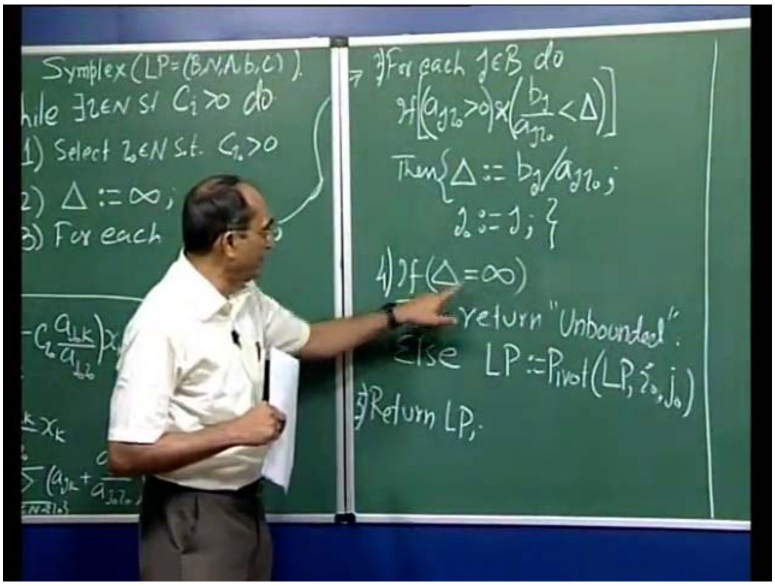
Now, if this while look terminates, then it will terminate only when all C i’s are less than or equal to 0 notice that, when all C i’s are less than or equal to 0 then the basic solution of that LP in which all the x i’s are set to 0 will give me the optimum value of phi the reason is this is a negative term, by raising the value of x i we will never contribute anything positive. So, the best value of phi can take is C naught.

(Refer Slide Time: 08:17)



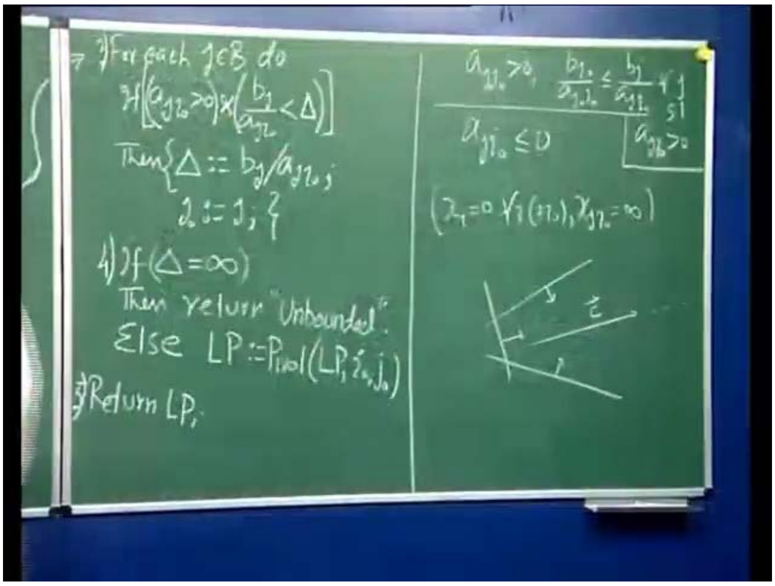
So, we will output that LP and the basic solution of that LP will be our optimum solution. So, that is one way to terminate here.

(Refer Slide Time: 08:33)



In case we are in this loop and we come to this stage, we find that delta is infinity. Now, that as we notice happens when for that particular choice of i naught.

(Refer Slide Time: 08:49)

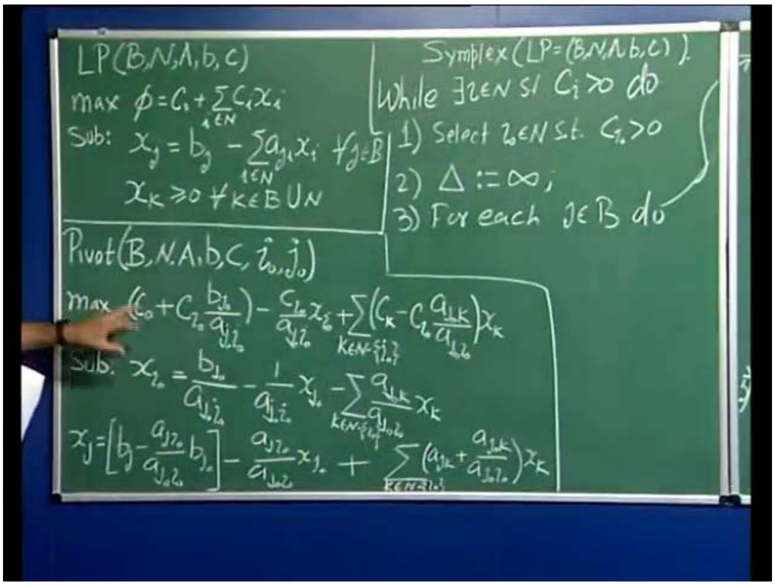


All j i naught are less than equal to 0 yesterday we noticed that, when this happens then x i equal to 0 for all i other than i naught x i naught at equal to infinity, if this assignment we make to the variables of n, then we will still be satisfying the conditions. And the corresponding value of the object function, will be also unbounded hence, what we notice is that there is a solution, with arbitrarily large value of phi object function.

Hence, this solution is unbounded and once again you can remember, that we have seen such a feasible visibility space, in which the solution was suppose to be on this side and lets say c is pointing in this direction we can go arbitrarily far, continue to say in the feasible region and increase the value of the object function. So, that validates the fact that we are, outputting unbounded in this case.

Hence, the last situation is, when neither of these two are happening then come to this step. This step say the following that the C i naught is positive, the j naught the j naught is. So, selected that two things are happening that a j naught i naught is positive and b j naught by a j naught i naught is less than or equal to b j over a j i naught for all j such that a j i naught is positive, among those j for which this is positive, this is the j naught is, so chosen that this is the least of point.

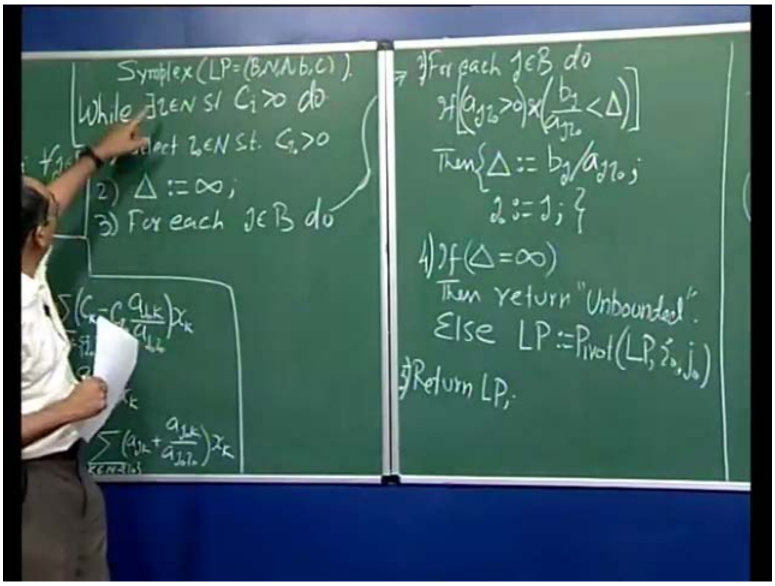
(Refer Slide Time: 11:30)



Now, lets take a look again into this expression, that condition guarantees that these are all non negative, the last condition of the choice of j naught guarantees that this is non negative. It also guarantees that this is non negative, remember that b j naught was all b j’s were non negative to start with, because we are assuming the input linear program is, such that its basic solution is feasible. So, all b’s the given b’s were positive non negative. So, this is also non negative.

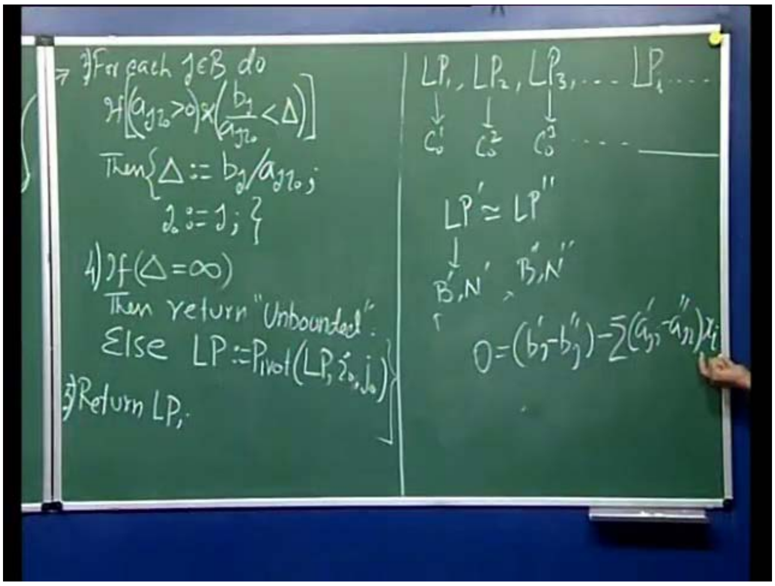
Hence, the new program that we find has these coefficients non negative hence it is basic solution is also feasible and that allows us to continue with this loop because once again the new LP is such that it is basic solution is feasible. One more thing we noticed is, that the object functions value for the basic solution can only increase, can never decrease because this is positive, this is positive this is positive sorry this is positive this is non negative and this is positive. So, it may be 0, but otherwise it can only increase.

(Refer Slide Time: 13:05)



These are the two ways actually for termination, one is that all these are negative and second that the solution turns out to be unbounded, but then there is a third possibility. The third possibility is, that this while loop which terminates here, remember the while loop terminates here, some how while loop does not end and this goes into an infinite loop, if it goes into an infinite loop, then we will have a series of linear programs.

(Refer Slide Time: 13:49)



Lets say we have, if there C values, that is the value of the object function in there basic solutions are lets say C 0 1 C 0 2 the constraint part of the object function, are we know that the maximum value of the object function is bounded, because if it is unbounded it will be captured in this ways, ultimately will freeze and after that this will never change. Now, we have at some point then that this increases and it remains the same throughout, if it remains the same, we will have to look at what happens to the corresponding LP from that point onwards.

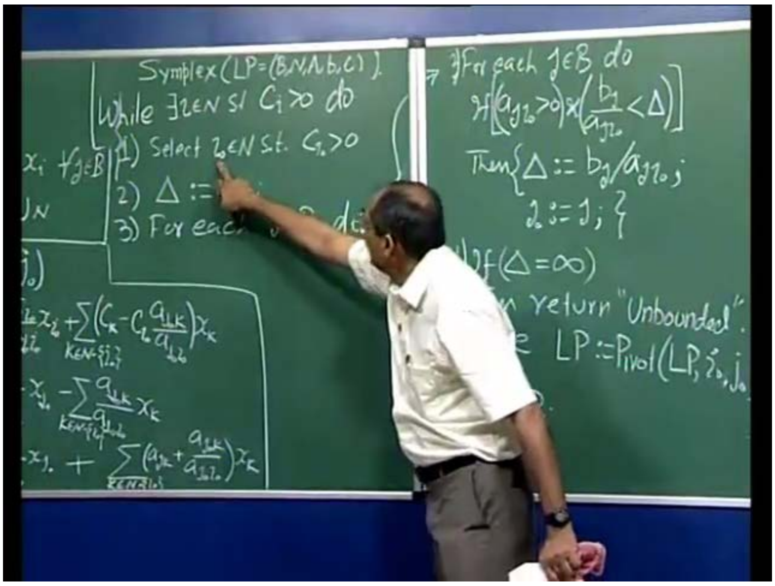
Now, incase this thing is fixed and if we keep record of each LP by simply storing there the B i and N i set the index sets, when we notice that after a while this set must repeat because of this is an infinite change, the set of variables the union of B is N is same, it is a same sort of variables, in each step one variable migrates from here to here and one from here back to N. So, ultimately because there are only mod B plus mod N C mod B possible ways you can partisan those variables into these two sets and there sizes are always same.

So, ultimately this pair must reoccur somewhere. Now, we are claiming that, if the pair is the same then the linear program has to be identical. Now, that is not very difficult to see the reason is, that these are all equivalent linear programs and suppose, we have 2 LP prime and LP double prime which are equivalent, there sets are also same B and N is same as B and N double prime, prime here if they are same and these are equivalent.

Then it is not too difficult to see that x j equal to b j prime minus some a prime j i x i and x j equal to b j double prime minus some a double prime j i x i we will have these two equations, one in this case and one in the second case one can take the difference of the 2 and you end up with the equation, if you take the difference you are going to get equation to be keep 0 equal to b prime j minus b double prime j minus some a prime j i minus a double prime j i x i this equation must be satisfied by every value of x i in the feasible region.

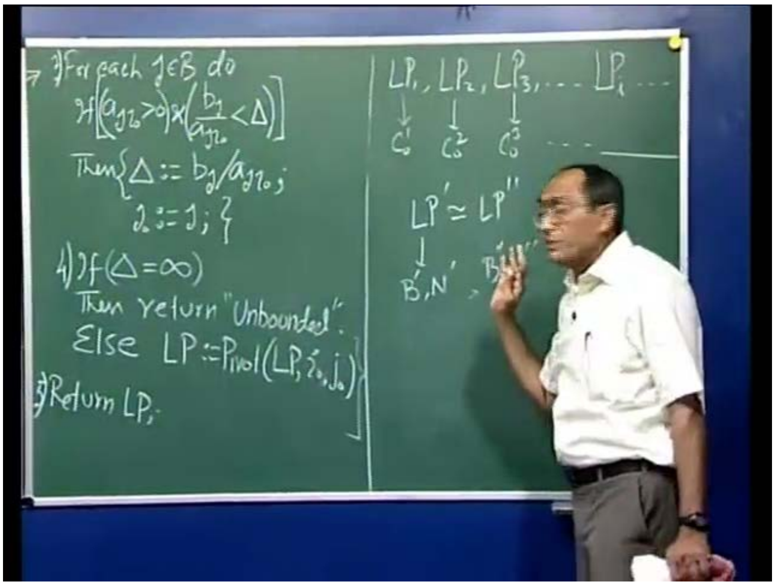
Because, they are equal in for every value of the x i set because this is an equation of half plane and this equation can be satisfied, if it is non trivial that is say if not all these coefficients are 0 if they are not all 0 then this is the half plane and that can be 0 only on the plane, once you get out of the plane. So, if you are out of the plane, if you are here or here this cannot add up to 0. Hence, we know that this is 0 and this is 0 for every value of j well, in that case we conclude that the 2 linear programs are identical and that just reoccurring and of the reoccurring, then we are going to gain anything. Hence, we have to look at a point when, there are no new programs are occurring, at that time we can stop and that linear program all of these, in fact all of these, give you the same object function all of them can be given as a solution, we written the linear program as solution.

(Refer Slide Time: 18:56)



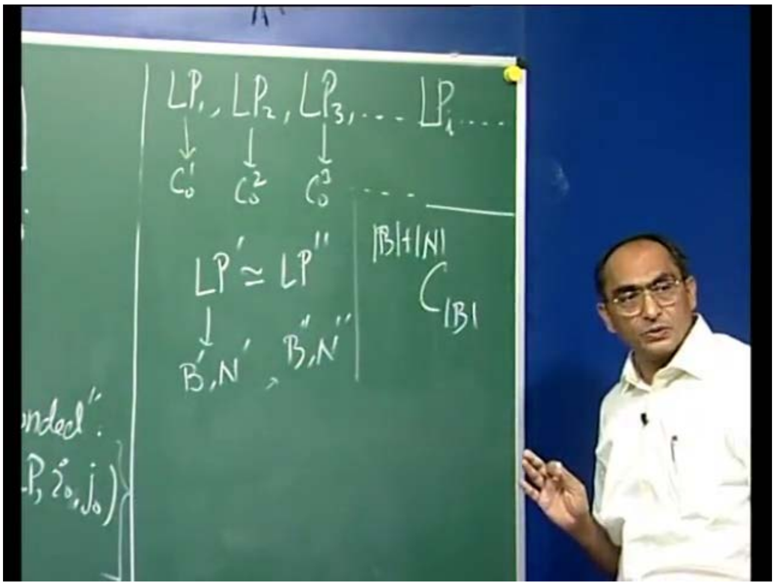
So, any one of these are as good as the other and notice that the two points and we to selection here, for i naught.

(Refer Slide Time: 19:03)



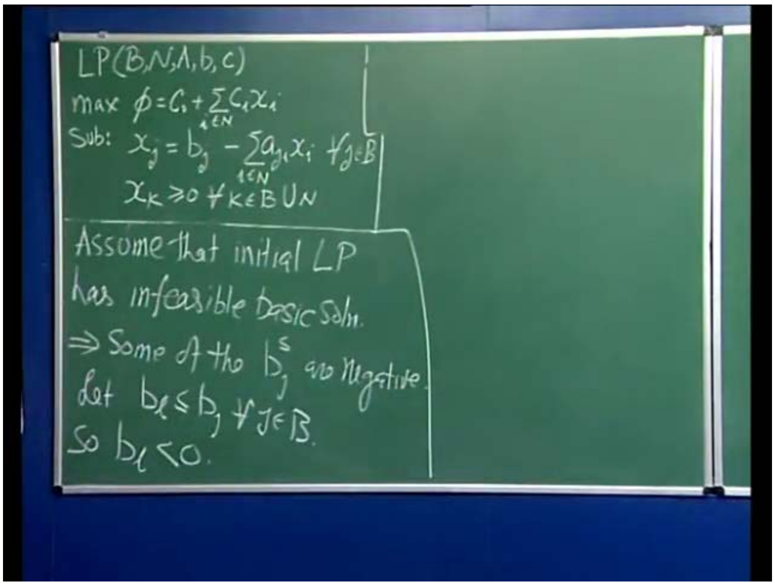
And here for j naught, if we have several j for which delta is same we can choose any one of them, we must try all possibilities. So, that every combination is created eventually when you capture that this is not going to change it is has it is entered into an infinite loop we can stop and we will get the solution as any one of these.

(Refer Slide Time: 19:19)



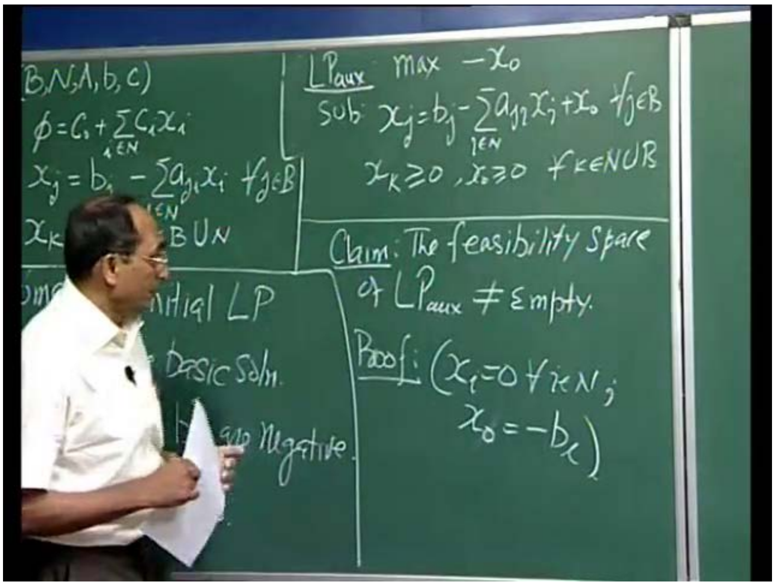
The point to remember is that the total number of linear programs is only, so much. So, this loop, this series cannot be longer than this. So, we can actually stop after we have found that these many steps are occurred and we have not discovered any new linear programs. So, now, we are ready to go to the last step of the program. And notice that we had. So, far discussed the algorithm which works when the initial in a program has a feasible basic solution, but when you are given a liner program it is possible that the basic solution for that program is not feasible, that does not mean that it is a solution space is empty or its visibility space is empty that is not in glide.

(Refer Slide Time: 20:54)



So, now we are going to show how to deal with the case when the initial linear program infeasible basic solution infeasible now let me first. So, let me just write down now assume that initial LP has infeasible basic solution, that simply means that some of the b j's are negative, because if all b j's were positive then we have a feasible solution a basic solution. So, let now, we will assume that b l is less than or equal to b j for all j in B. So, b l better be negative that is this is another assumption.

(Refer Slide Time: 22:09)

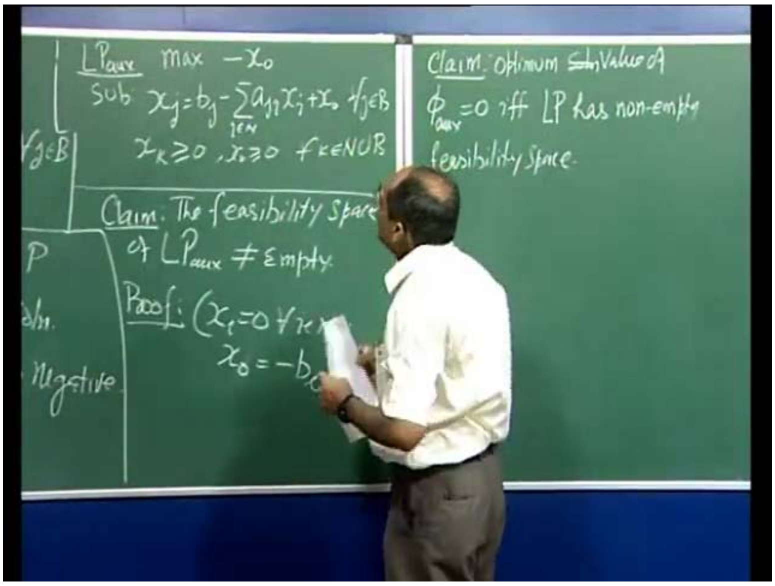


So, now, first I am going to introduce a variant of our linear program and I am going to call it LP auxiliary LP aux in this I am going to maximize I am introducing a new variable x naught, which was not present earlier and I am maximizing minus x naught this is the only expression, subject to this almost the same expression, we have x j equal to b j minus a j i in n x i plus x naught, for all j in b and x k must be non negative x naught must be non negative, these are k r in N union B.

So, I have just read it in the same program, only thing is I have introduced an x naught in the condition part, I have just added x naught here, but I completely replace the old object function by this expression namely minus x naught. And now, lets look at this program, first thing I am claiming is that the solution space of auxiliary program is feasible, that is the feasibility space of LP auxiliary is naught empty. And that we can see as follows, all we have to do is give one solution to these conditions and that is not too difficult if you take x naught to be minus b l.

So, the proof is take x i to be 0 for all i in a and take x l to be minus b l notice that b l is negative. So, minus b l is positive x l is positive here, all x i's are 0 and then if you notice, these terms vanish b j's are all greater than minus are greater than b l. So, b j minus b l is non negative here right yes, this is for x naught yes x naught is minus b l and when I plug in minus b l then this is b j minus b l which is non negative by the choice of index l. Hence, all of these are non negative, it satisfies all the conditions hence, this is a solution point inside the feasibility space of LP auxiliary.

(Refer Slide Time: 25:44)

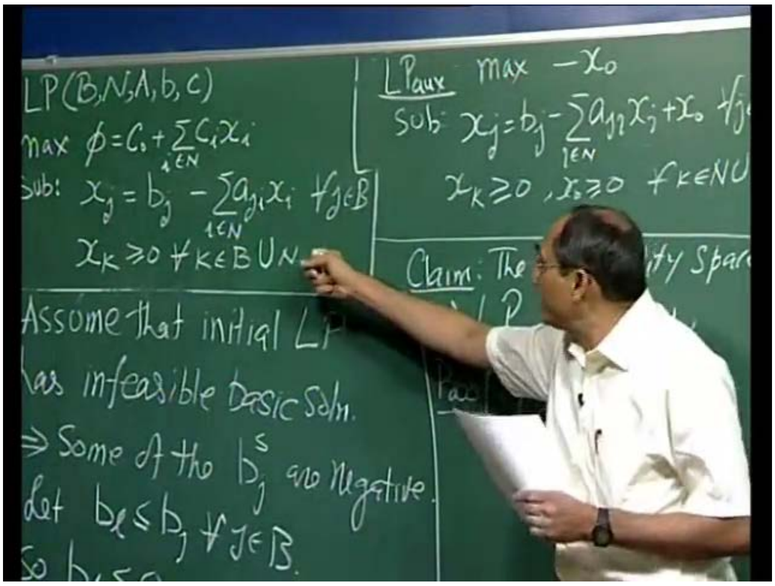


second claim we can make is that, the optimum solution or optimum value of phi auxiliary is 0 if and only if the original LP has non empty feasibility space, that is to say, there exist at least one solution which satisfies the condition of LP if and only if, the optimum value of this program is 0 way to see this stream is, that notice this is minus x naught. So, the highest value and the fact that x naught has to be non negative, it cannot reach above 0 it can be otherwise it can either negative or 0.

Now, if this LP has a solution, if any solution, any point which satisfies the condition of this LP then that will satisfy this condition with x naught equal to 0 will satisfy these conditions, by taking x naught equal to 0 you append the solution of this with x naught equal to 0 we see that these conditions are satisfied and the value of this is 0 and that has to be optimum solution of this, conversely if the optimum solution of this gives me this value to be 0; that means, it is giving me x naught equal to 0.

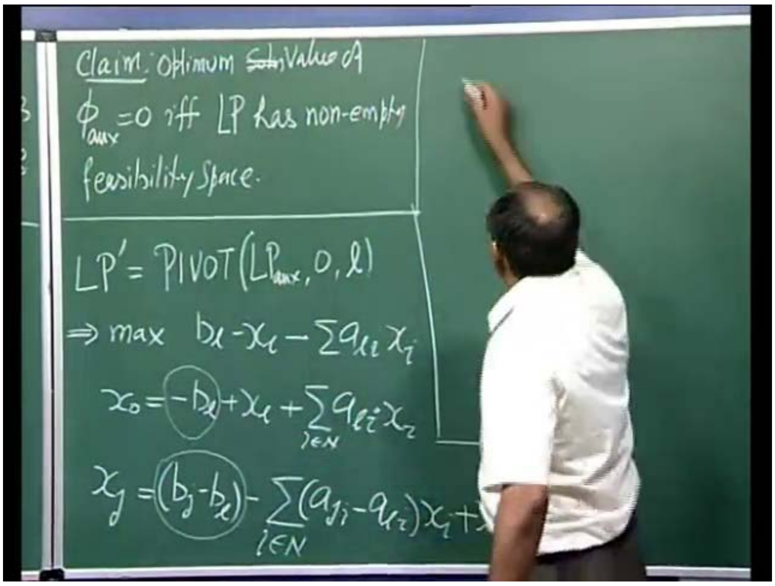
Hence, we have values of x j’s which are non-negative while x naught is 0 because the solution of this, will automatically become a solution of this. We if we just drop x naught term because that is taking value 0 in the optimum solution. Hence, what we have found is there is a relation between, the optimum solutions value, of the auxiliary program and the linear programs feasibility space.

(Refer Slide Time: 28:46)



We are going to exploit this, our goal is now to figure out whether the feasibility space is empty and if not then we have to find equivalent program for the original LP in which the basic solution is feasible. Once, we get one equivalent form in which basic solution is feasible, we can run the simplest program as described earlier.

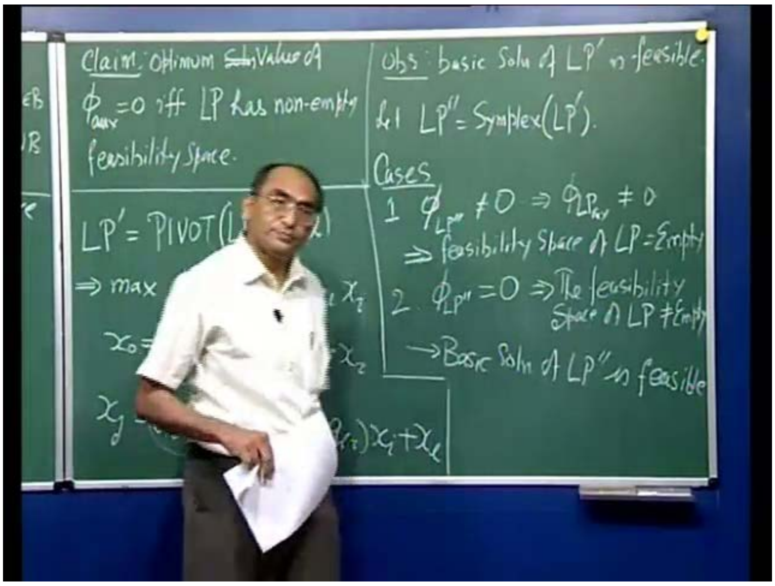
(Refer Slide Time: 29:05)



So, now I am going to consider the following transformation lets now, take LP prime which is the pivot of LP of which 0 the index 0 is the index of this variable and l, l is that in index. So, it be for which b l goes the least, we are going to compute this and lets write down the expression for this, we are getting max first let me write down the condition, we find that x naught looks like minus b l plus x l plus some change l i x i.

So, this is the equation for x l we have transformed and brought x naught on this side, the other equations look like x j equal to b j minus b l minus some a j i minus a l i x i plus x l for i e N and our object function was minus x naught. So, it should be the minus of this will become b l minus x l minus some a l i x i this is our pivot that a new program a pivoting about these two coordinates. Now, first of all let us look at the whether the basic solution is feasible or not. So, notice that b l was negative and now this term is positive this is; obviously, non negative because b l is the smallest among all the b j's. So, both these are non negative.

(Refer Slide Time: 31:54)



So, we observe that the basic solution of LP prime is feasible. Hence, we can actually run the program, we can run the simplest program starting from LP prime alright. Now, let LP double prime be the result of running simplex on LP prime, what we had observed earlier is that LP auxiliary has a non empty feasibility space, LP auxiliary is equivalent to LP prime, because we have performed one pivoting subsequently we have, perform simplest algorithm on that. So, this is also equivalent to LP auxiliary.

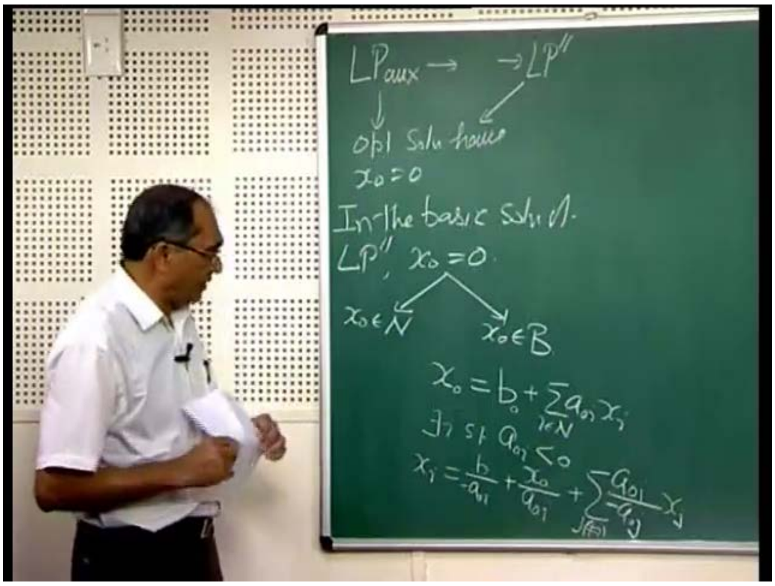
Since, the feasibility space is non-empty; we have of course, solution and further more the maximum value is bounded by 0. So, it has to provide a solution to us, this is the final LP and it is basic solution is what we have to look at to decide whether, it is value is

0 or not. So, now, we have two possibilities, cases 1 the phi value LP double prime is not 0. If we find that the object function has non 0 value in LP double prime this implies phi value of LP auxiliary is not 0 because they are same programs, which implies that feasibility space of LP is empty.

So, we can start, this is one of those cases where we have to be on this side of this, half page this side on this half page this side on this half page there is no region, where they coincide all of them. So, we can stop here and say that the feasibility space is empty. Now, we have to see the case where phi of LP double prime is 0. So, now, the 2 nd case is when the optimum value of the auxiliary program is indeed 0 that is the object function evaluates to 0.

Now, in this case, we know that our original LP the feasibility space of original LP is not empty, that is what we just saw. So, I have to know give you a program equivalent to LP in which the basic solution is feasible, in hence you can then run simplex program, starting from that LP. Now, let us look at LP double prime, what we notice is that LP double prime is the last form, when we run simplex program on this. And we notice that, every in this case when we start with a program which has feasible basic solution every subsequent program has feasible basic solution. Hence, LP double prime has. So, we notice that the basic solution of LP double prime is feasible because it is coming from there. Now, this is feasible.

(Refer Slide Time: 37:09)



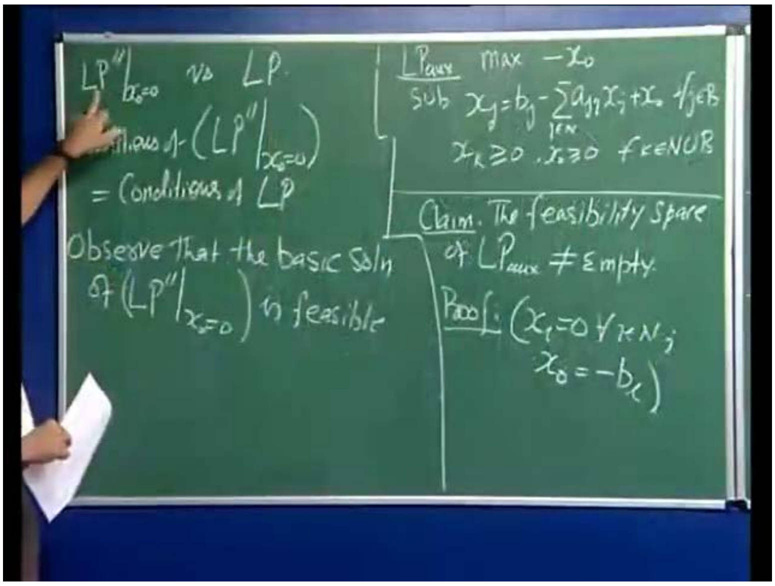
Now, let us just look at for a moment what is the relationship between the LP double prime and LP auxiliary what we noticed is that from LP auxiliary a series of a transformations let to the pivoting transformations let to LP double prime. Hence, the two problems are equivalent and hence their solutions are same now, we know that the only solutions which are optimum have at 0 equal to 0. So, the optimum solutions have x naught equal to 0 and hence, the same true for LP double prime.

Now, the optimum solution of LP double prime is it is basic solution. So, in the basic solution of LP double prime. x naught must be 0 now there are two possibilities, one is that x naught belongs to N or it belongs to the set B of variables. So, this is fine with us lets just take a look at this case, in this case there must be an equation in LP double prime which looks like, x naught equal to b plus some a 0 i x I, i belonging to N of course, this in the basic solution turns out to be 0.

So, and b is non negative or we can say b naught. So, there must be some there exist some i says that a 0 i is negative. Now, in this situation I can perform one transformation where we swap x i and x naught and that will be x i equal to b over minus a naught i plus x 0 divided by a 0 i plus some a 0 j j naught other than i rather divided by minus a naught i x i a naught j this solution, has one property that since in the optimum x naught was 0 then again the basic solution after this transformation is still the optimum solution of LP double prime.

Lets and in this case a it is not is actually, a member of m say because it come under other side of the quality. So, in uniform situation we will assume, that LP double prime has x naught in N because if it is not in n I can always make this transformation and we end up having x naught in n and the basic solution which is feasible gives you the optimum solution of LP aux.

(Refer Slide Time: 41:14)

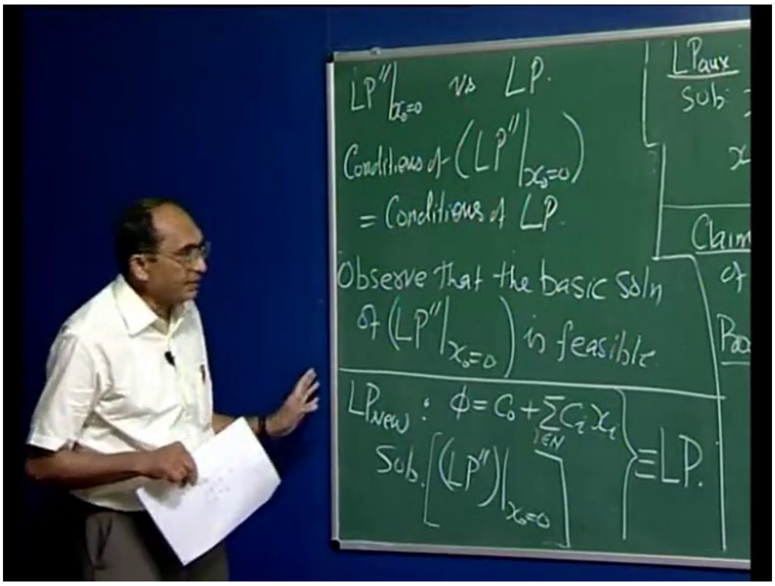


L P double prime by forcing x naught to 0 with LP notice that, in the basic solution x naught is 0 we can now, set x naught to 0 and leave all the other variables as it is. In our, auxiliary program if I force x naught to 0 this program reduces to the original LP as far as the condition part is concerned. So, what we notice is, that the conditions of LP double prime with x naught force to 0 is equal to condition of LP they are same, but the advantage is that LP double prime had feasible basic solution, then this also must have feasible basic solution because the basic solution of this had x naught equal to 0.

So, all the other variables values will constitute the basic solution of this program. So, observe, that the basic solution of LP double prime with x naught restricted to 0 is feasible, notice that the basic solution we can actually give, by taking the basic solution of LP double prime and just dropping x naught because there was x naught 0 there. So, we are anyway removing that positive.

So, what we have, as far as the conditions are concerned they are equivalent to LP and the basic solution of this is feasible, that is all that matters the object function does not play any role in this part.

(Refer Slide Time: 43:34)



So, now, we are ready with a new program L P new, in which we will start with our original expression phi to be C 0 plus some C i x i x belonging to N and the conditions will be replaced by LP double prime, restricted to x naught equal to 0 this only one thing you may have to do is N and d sets of this may no longer be the same as the original one. So, for all those variables of N old N which are in d set of this, we can plug in those values for this. Just plug in those values, because you will have suppose, i a particular i naught in N is in b set of this will there will be an equation for x i naught here, we plug in those values.

So, we will suitably transform this with respect to the new set of variables and now we are ready with a new program, which is completely equivalent to the original LP, but the advantage of this program is that it is basic solution is feasible. And since, that is feasible now at this point we will, begin with simplex program using this program and not this program, this ends the entire algorithm this is where we stop, because now we have completely described all possible the outs for starting from any linear program, we have seen all possibilities and every possibility has been taking care of in the complete program.