**Linear Programming and its Extensions**

**Prof. Prabha Sharma**

**Department of Mathematics and Statistics**

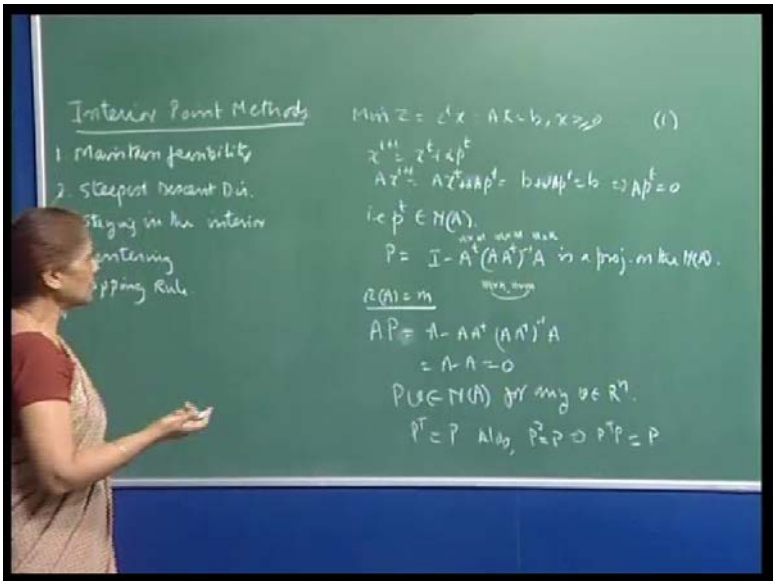
**Indian Institute of Technology, Kanpur**

**Lecture No. # 40**

**Interior Point Methods**

So, to introduce you to interior point methods, I had already started the discussion in my last lecture and I will just take over from there.

(Refer Slide Time: 00:29)




So, we said, that we need to first maintain feasibility. That means, see, remember the problem is minimize Z equal to C transpose x subject to A x equal to b, x greater than or equal to 0. I will call this problem 1, LPP1. So, we said, that we want to maintain feasibility. That means, if I am moving along the direction, so if at any point see the iteration t i have this plus alpha p t, let us say this is the direction along which I am moving and this is my step size, then to remain feasible I require, that A x of t plus 1 should be A x t plus alpha p t, which is because x t is already a feasible point. This is b plus alpha p t should be equal to b because A x t plus 1 should also satisfy the conditions

A x equal to b and therefore, this implies, that your p t, sorry, I should have written here, this is plus alpha A p t. So, this is plus alpha A p t, this implies, that A p t must be 0 because alpha is a step size, the alpha is, obviously, not 0. So, that is and in the, in the terminology of linear algebra, we say that, that is p t belongs to the null space of A.

Because if you take A x equal to 0 and the solutions to this system of homogenous equations is a subspace, which I had discussed with you quiet some time ago when we were discussing the mathematics, required mathematics level, that we want for this course. And so, this will be, this is also called the subspace, the solution space of A x equal to 0 is called the null space of A because every vector gets mapped to 0 by the matrix A. So, P transpose must belong here and this is, so therefore what we say is that we want to project, so that means, and the, and you can show, that P, if you write P as I minus A transpose A A transpose inverse A is a, projection, projection on the null space of A.

And let me just explain because this is the part we did not cover in our math preliminaries. So, the idea here is, see, of course I have assumed, that rank A is m, that means, a is full (( )) and therefore, the matrix A A transpose would be what? This is m by n and this, this will be n by m A transpose. So, the A A transpose would be n by m matrix, this is non-singular you can show; I think, I have put this as an exercise for you to show, that this will be a non-singular matrix and therefore, the inverse will exist.

And you see, that if you take any vector P v and then you apply or just apply A on P, then this will be A minus A A transpose, then you have A A transpose inverse and A. So, this together A A transpose and its inverse gives you identity matrix. So, this is A minus A which is 0. So, AP is a 0 matrix and therefore, P v belongs to null space of A for any. So, here my P is n by m matrix for any v belonging to… So, let us see, this, no sorry, so it is a transpose is n by m, this is n by m matrix and this is m by n. So, p is n by m matrix, so p is n by m matrix for any v belonging to R n. So, therefore, this is what we mean by projection matrix.

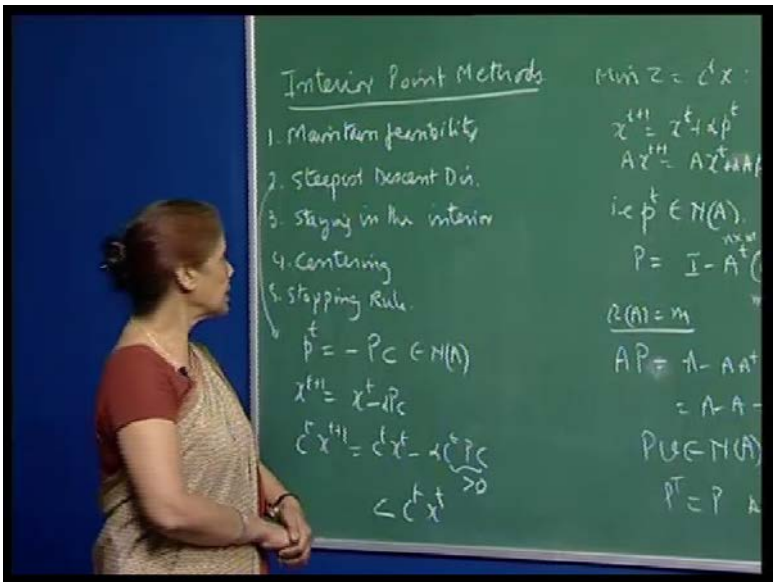
That means, when, if you take any, if I, if I say, that P is a projection on the null space of A and I mean, that the vector Pv for any v and R n, this will belong to the null space of A, this is what we mean by that. And these projection matrices are special matrices, I have put 1 or 2 as an exercise, you can 1st show, that P transpose is P. Also, as a

definition of a projection matrix, you have, that P square is equal to P, this is a definition of a projection matrix, you can verify here and therefore, this will imply, that P transpose P is also P because P is symmetric. And therefore, I have asked to do these simple exercises in the assignment sheet, which I will discuss at the end of the lecture.

So, therefore, you take a direction, which is belonging to the null, that means, you move to remain feasible, to maintain feasibility, you move in the direction, which is a vector in the null space of A, so then, there is no chance. And then, of course, you choose alpha appropriately. So, so, now, what we are saying is that when I look at number 2, so that is what maintains feasibility.

Now, the steepest descent direction because you see, the objective function is C transpose x gradient vector, here would be C and the direction of C would be maximum increase, but we want decrease. So, minus C would be the maximum decrease and that is why we call it the steepest descent direction, maximum decrease in the objective function will happen if you move along the minus C direction.

(Refer Slide Time: 06:51)



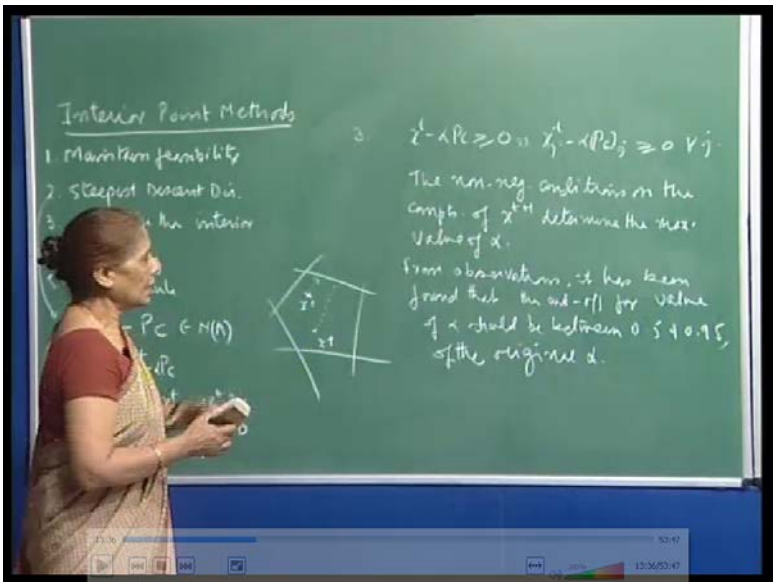

So, therefore, what we say is that and then, we want the direction to be the direction of movement to be in the null space of A. So, this implies that you will take your P the direction of movement or if you want to call it P, P t; that means, the time of t at iteration, this will be minus P C.

So, C is the direction of maximum, minus C is the direction of maximum decrease and then I take the vector minus PC. So, this is now belongs to, so this belongs to null space of A and you have projected the maximum decrease direction of the null space of A, so again, it stays as the maximum improve, (( )) well, we can verify. So, this will be your steepest descent direction and you can check, that if you write x t plus 1 as x t minus alpha PC, then you want to verify the value of the objective function at this point. So, this will be C transpose x transpose minus alpha C transpose PC.

And now, using these properties, I want you to do this an exercise and show that at these quantities always greater than 0, C is a nonzero vector. So, therefore, this is less than C transpose x transpose, so I have shown you, that alpha is a positive number. So, what we have seen is that if I move, if my x t plus 1 is, that means, if I moved in the direction PC, then minus PC, then my objective function value will decrease. So, steepest descent direction gives us decrease in the value of the objective function. So, that is what we do.

And now, staying in the interior, yeah, so staying in the interior, the, the point x t plus 1 satisfies the condition A x equal to b, the conditions I showed you because this is a projection null space.

(Refer Slide Time: 08:50)




Now, the question is, you want also x t minus alpha PC to be greater than or equal to 0 for all, or in other words, this implies, that you want to the jth component minus alpha

PC, jth component to be greater than or equal to 0. If you see, that from, immediately from here, if, if PC, if the jth component of PC is less than 0, then of course, this condition is not going to be violated for any value of alpha because this becomes positive and x j t's are already positive. So, whenever jth component of PC is, less than 0, is greater than 0, then you will get a limited alpha.

So, from here, you see, all these conditions for all j, all these conditions have to be satisfied and so this determines the, the non-negativity conditions, negativity conditions, on the components of, components of x t plus 1 determine the maximum value of alpha. That means, for all alphas for which PC j is positive, you will find out a value of alpha, so that this number remains non-negative and then you will choose the minimum of all those alphas. So, that, all that conditions have simultaneously satisfied with, that is, none of them violated. So, determine the maximum value of alpha.

So, now, the idea here is you want to stay in the interior, as I told you because this is, we do not want to approach the boundary, and because if we, if I reach the boundary, then my next iterate will not be, I will not be able to have a reasonable step size. So, staying in the interior means… So, the, from empirical, people have from observations and from observations it has been found, that a value, that the cut-off, cut-off for, for the cut-off, for a value of, for a value of alpha should be between 0.5 and point 0.95 of the original alpha, of the… So, if, that means, because the value, even the minimum value of alpha, that I will get from these conditions would at least be on the boundary of one of these non-negativity constraints, and I, and as I said, we want to stay in the interior.

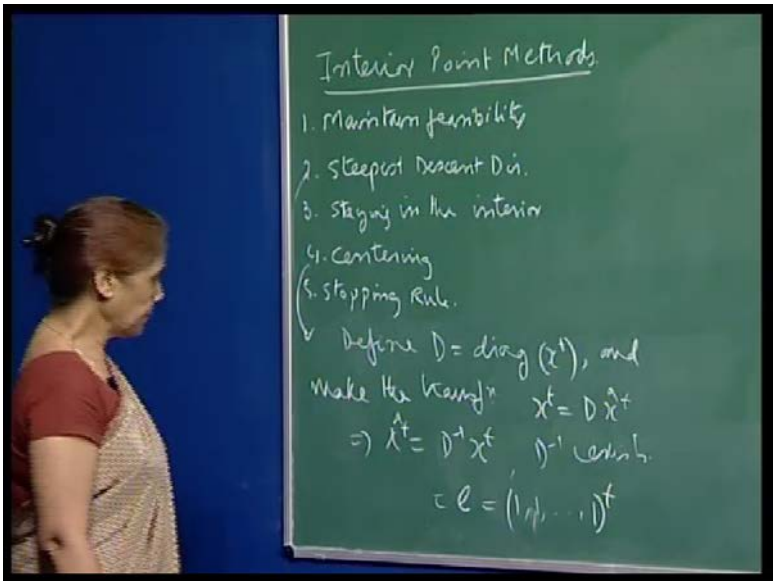
So, therefore, the idea is, this has been after running lot of many problems, you know the algorithmal many problem, it has been found, that if you cut down the size of alpha, the step size, that you want to move of the original alpha to 0, between 0.5 and 0.95, then it gives you good results. So, this is what we are staying in the interior.

And now, idea about centering; see, as I told you and the, you will, I had drawn a figure. So, we had said, that if the point, if your point x is here, then you know, you can only move and you are moving in this direction, suppose if the value of the objective function is improving, decreasing in this direction, then from here, the step size is only this much, but if you are somewhere here, then the step size would be much larger and the idea is

that you want to decrease as much you want to, as much decrease in the value of the objective function as possible. So, centering also helps. That means, you want to...

So, depending on where your point x t is, then we make a transformation and we transform the whole feasible region, so that the current iterate is at the center of the feasible region and that is not difficult. So, this is what we will discuss in the next step. So, the, see what I am trying to do here is discuss the key ideas, that go behind the development of the interior point method and then, we will discuss the actual implementation. So, I, then I will give you the exact algorithm. So, here, yeah…

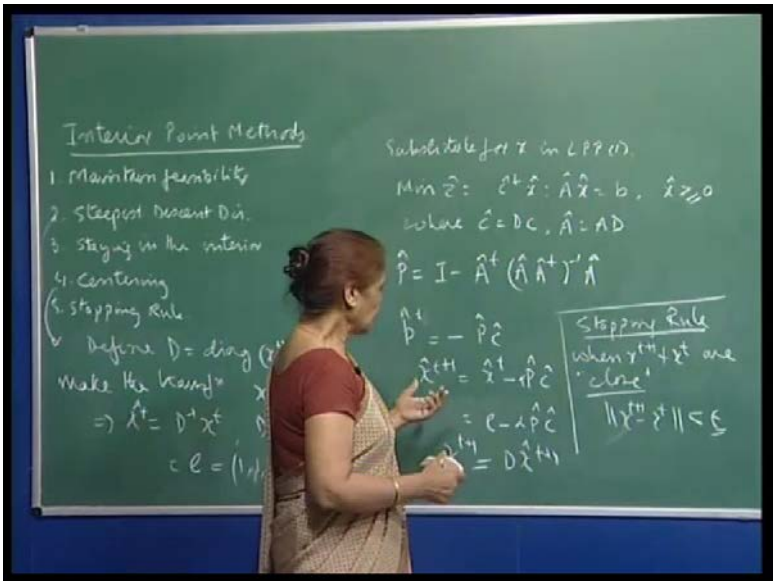
(Refer Slide Time: 13:39)




So, let me now discuss centering. So, here if you have the points at x t, define D as diagonal x t. That means, that the entries, this is the diagonal matrix, where the diagonal entries are x t 1, x t, x t 2 and x t n, all other entries are 0. Define this, and make the transformation, and make the transformation x equal to Dx hat. So, therefore, this would imply, that x hat is D inverse x and you see, and I am writing x t here, so let us make the transformation, this is diagonal, so this is this and this is this. So, then, x hat t would be this and you see, that the, now the inverse and, why that the D inverse exists, why D inverse exists? Because you see, remember, we are at every point, at every iteration in the interior of the feasible region, which means all components of x t are positive.

And therefore, D, when you take the diagonal matrix D, the value of determinant is product of the diagonal entries, which is non-zero and therefore, this is the invertible D inverse exists and when you do this, now when you multiply D inverse with x t, you see, you will get an identity vector. So, this is equal to e, which is equal to 1, 1, 1 transpose. So, all the components of the vector x hat t are 1, 1, 1. So, this is what we mean by, so you have scale the, the variables, the variables vector x t by this, doing this you are making this transformation.

(Refer Slide Time: 15:59)




And now, what we will do is, substitute this in the original problem, substitute for x in LPP define by 1. So, this will give us what? This will give us minimize z hat equal to C hat transpose x hat subject to A hat x hat equal to b and x hat greater than or equal to 0.

Where, where your, where your C hat is, again c you, in original problem you had minimize C transpose x. So, for x, I am making the substitution Dx hat, so then this will be DC. So, this C hat becomes DC and A hat becomes AD because x hat is Dx, oh sorry, i mean x is, x is Dx hat. So, when you have this constraint, Ax equal to b for x, I write down Dx hat, so then it will be AD. So, the matrix now becomes this and with this transform problem.

So, in this case, once you have done the centering, you see, because now x hat has components 1, 1, 1 and in the (( )) because you have these non-negativity constraints. So,

you have brought the point, the current iterate from which you are going to work, look for the step size with the direction in which you want to move. You see, it is equidistant from all the boundaries, which are your xj greater than or equal to 0. So, if you look at this (( )), non-negative (( )), then this point, the current point is now equidistant from all the...

So, then, you see, what you are trying to say, that your point is now not biased by any constraints and your step size should therefore, come out to be a good reasonable size. So, this is what we have here, your projection matrix. So, therefore, your projection matrix P hat will be now, I minus A hat transpose A hat A hat transpose inverse A hat. So, this becomes your A projection matrix. So, then, you find out the direction of A projection, which is b transpose t plus 1, you can call it then minus this will be this will be your P hat C hat, because now you are working with the new problem, with the transform problem.

So, this is your direction of movement and then you will be, you will find the, let us just make sure, that, fine, just keep it as it is because we are at x t right now, and so let us, P hat is minus P hat C hat and then I will move. So, therefore, my x hat t plus 1 is x hat t minus P hat alpha P hat C hat, where x hat t is actually your e, remember. After the transformation this becomes 1, 1, 1 minus alpha P hat C hat and so we can determine the size alpha, as I told you by making sure, that non-negativity conditions are satisfied. And then, we will cut down this alpha to between, number between 0.5 and 0.95. So, once you have obtained this, then you go back to the original space because remember, your D gets defined by the original problem.

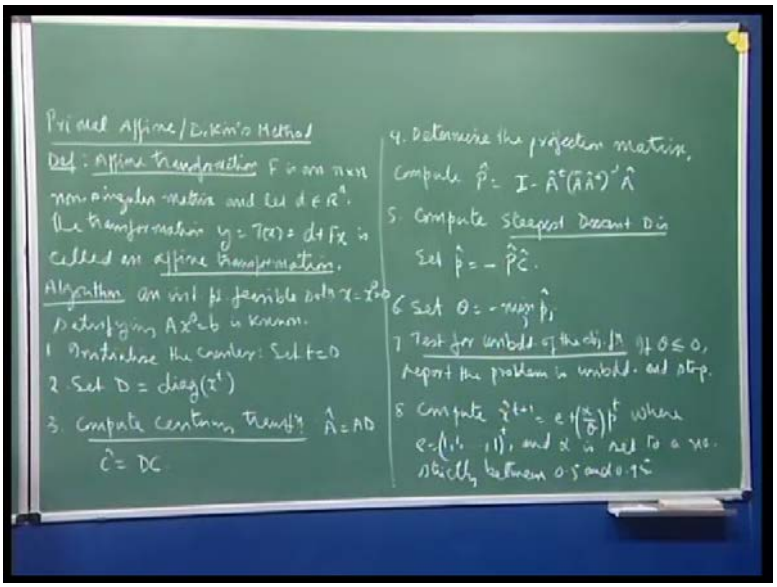
So, here, what do we need to do here? Once you have this, then you will have to define x t plus 1 as D of x hat t plus 1. So, you, you would do the descaling or the rescaling, whatever you want to call it. So, they, once I have obtained this point because this, the point x hat was centered in the feasible region, so my step size was, comes out to be a reasonable step size and using that I get the next point. Then, I go back to the original space x t plus 1 and then, you know, from x C plus 1, again I will define my nu. So, x t plus 1 will help you to define my nu D and I will then continue with the problem. So, this is what, wherever centering is.

And stopping rule is yeah, so stopping rule is simply, stopping rule is when x t plus 1 and x t, remember we will work with x t plus 1 because x hat t plus 1 and… So, by, after scaling all become 1, 1, 1. So, when x t plus 1 and x t are close, so here you can put, you, this thing for example, most of the people say, that the distance between x t plus 1, that means, nor, nor x t plus 1 minus x t is less than some number epsilon. You can stop because then you say, that the current point, the current iterate is close to the optimal solution and so this is the basic idea.

And let us now look at the implementation part, so we will discuss some more aspects there and then I will try to show you, where these algorithms, where these methods stand. So, before i actually, so what I have given you here are the ideas behind. So, these are the basic ideas behind any interior point method and right now, I will discuss the very basic one, which is Dikin's method and it also, it is also know as a Primal Affine method.

Let me explain the word Affine here. So, affine transformation, I think I used it while defining polyhedral, while discussing polyhedral theory, but let us quickly review it. So, what we say is that in affine transformation, is an n by n non-singular matrix and if d is some vector in R n and the transformation y, which transforms x by this rule d plus F x is called an affine transformation and you see that everywhere in this, these are affine methods.

(Refer Slide Time: 22:55)



Dikin's method is an affine transformation method because x hat t plus 1 is x hat t minus alpha P hat C hat. So, the transformation is from x hat t to x hat t plus 1 and this is the direction. So, here, actually your transformation matrix is the identity matrix. So, this, the affine transformation is given by this, that is what I wanted to find out, that this is the direction, which you are moving and this is the point, which gets transformed to x hat t plus 1.

Now, let us look at the steps of the algorithm and I will try to discuss the details as we go along. So, the first, the 1st, the starting point is that you have an interior point feasible solution x equal to x naught t and the point x naught satisfies the condition A x naught b, it is an interior point solution x naught is strictly greater than 0, that means, all components of x naught are positive. Then, we initialize the counter, we said t equal to 0, it was, the starting point is x equal to 0 is x naught and then you define the diagonal matrix D, which is made up from the current iterate, which is x t, whatever the counter and then you compute these settling transformation A hat AD, just explain to you here and this C hat is DC, the objective function prospector and the new coefficient matrix is A t. So, once you have this, then we will determine the projection matrix.

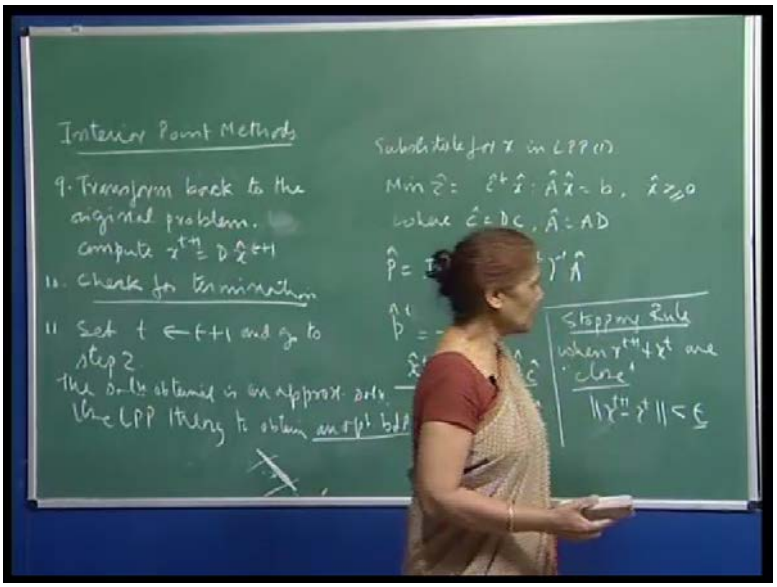
So, the projection matrix is give by this after the transformation. Then, we compute the steepest descent direction, which is minus P hat C hat, P hat is equal to here and this gives you the direction along which will move and then we put theta as minus min j P j hat. So, here is the vector, take the components of this direction and then you take the minimum of these, minus of that is theta. Now, test for unboundedness, this is where, again you have your have your linear programming theory coming to your help. This is, we said, that if because the direction of movement, if all components are less than or equal to 0, then what will happen?

If all components are less than or equal to 0 for theta, that means, your P j hat, the components would be, see, if this is less than or equal to 0, it means, that P j hats are all non-negative, remember. And your direction, if you, when, whenever you say, found, that your feasible region had a direction you stop and you said that the problem is unbounded right.

So, therefore, here if theta is less than or equal to 0 will report, that the problem is unbounded and stop, and again I think, I have asked you to sit down and verify, that yes

this is the unboundedness criteria, if theta is less than or equal to 0. So, this is the part of an assignment sheet and then, if, if this is not satisfied, then of course, you will go on step 8 and you will compute x hat t plus 1 is e plus alpha by theta P raise to t because you are normalizing, yeah, alpha by theta with respect to t, where e is 1, 1, 1 transpose and alpha is set to a number strictly between 0.5, point, 0.5 and 0.95. So, this is, this, once you have this and then of course, you will check for, so let me now continue with the...

(Refer Slide Time: 26:45)




So, yeah, now then, 9, we went up to 8, so you will translate, transform back to the original problem by using, by the transformation. That means, compute, let us compute x t plus 1 as Dx t, x t hat, x t plus 1 hat because we were discussing that here, so have to, you want to get the original thing, the vector x t plus 1 through D x hat t plus 1.

So, it transforms and then it checks for termination. So, here, one can have more than different criteria also, you might just look at the norm of x t plus 1 itself, but normally you want to see, that when you become getting closer and closer to your last iterate, then you want to stop because then, you know, that you are not going to get much improvement and the improvements in the objective function may be insignificant according to your this thing. So, check for termination, whatever criteria you want to set.

And then, 11 set. So, in computing, the knowledge you say, that t goes to t plus 1 and go to step 2. So, this is what you will do. Basically, I have discussed the algorithm and now just gave you the step wise description of the algorithm. So, this is what it is, yes, as I

told you, that when you stop, because you will be stopping when the norm of x t plus 1 minus x t is less than epsilon, so I was discussing the termination criteria.

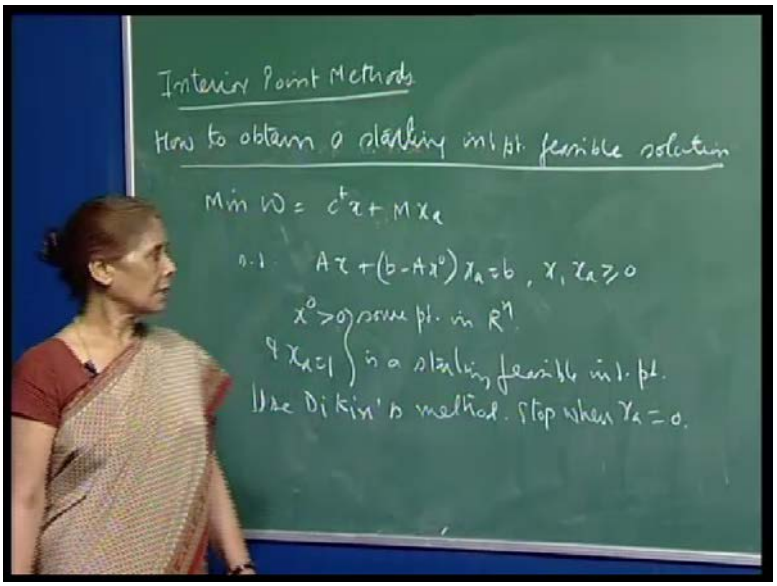
So, you, whenever you get a new solution, try to check for termination and of course, the, I had pointed out, this is, this can be one criteria, but there can be many more. In fact, (( )) has a different stopping criteria and so there could be many, but the whole idea is that, see what will happen is that, if there is a unique optimal solution to your original linear programming problem, then an interior point method will also terminate at that extreme point, at the unique extreme point.

But problem arises when you have a non-unique optimal solution, in the case of a non-unique, oh I should have said non-unique optimal solution. That means, the dual problem has degeneracy then, because you see, your one of the c j minus z j's will be more than, more than, m of them will be 0. So, you can have alternate optimal solutions and so on. So, in the case of a non-unique optimal solution an interior point method will typically not terminate at an extreme point, that is, at a basic, basic feasible solution, instead it will terminate at a point somewhere on the optimal phase of the feasible region. See, I told you, that if in case you have 2 optimal solutions, then every point on this edge of the feasible region, of the feasible polytop will have optimal solutions, all of them. And in fact, the, in more than 2 dimensions, it will be a phase because it will be a supporting hyper plane, so you, or we call it the optimal phase. So, every point on that optimal phase is an optimal solution.

So, your interior point method may end somewhere here, anywhere on the optimal phase of the feasible region. In that case, you want to still get to an extreme point because what happens in some applications, it may be, to have a non-basic feasible solution or a non-extreme point solution, but sometimes in, especially in commercial applications and so on, you want minimum number of non-zeros, you do not want to have too many positive variables in your solutions. So, then, because then in that way you have less choices to consider.

So, therefore, at many applications, that is desirable to have a basic feasible solution and therefore, you would want to be able to go to the closest basic feasible solution, which is the optimal, which is an optimal solution.

(Refer Slide Time: 33:29)




And in the, again, somewhere in the course of the, while developing the linear programming theory, I had shown you how to obtain a basic feasible solution from a feasible solution. But now, we will also, we will have to just extend the technique and which is possible, simple, you can just read it up that if you want to go from feasible solution to a basic feasible solution, such that the 2 x, 2 objective function values are the same because remember, this is an optimal solution and you want to go to basic feasible solution, which is optimal. So, the values of the, and essentially, you will be moving on that phase.

So, because from here, you either go here or you go here, then in the process you see, you just have to maintain, that the value of the objective function remains the same and once you do that, then he will be able to reduce feasible solution, which is an output of interior point method to basic feasible solution, so little more work will be required. So, you can see that, of course, in a linear programming problem you will always be able to end the algorithm at a basic feasible solution. Here, you will be required to do a little work to get to the extreme point optimal solution.

2nd (( )) of this set, of this class of method is that you need a starting interior feasible point and that can be a problem and… So, Dikin, of course, gives you a method for obtaining a starting feasible solution, which means, that you again apply the interior point method to get a starting feasible solution and then of course, there is no problem in

that, that you do, you can stop the moment, you have the feasibly satisfied and this will again be done through the big-M method, the same technique of using an artificial variable.

(Refer Slide Time: 33: 29)

Insert Slide here

So, let me just give you the, spell out the method for obtaining a starting feasible solution. How to obtain a starting interior point feasible solution and says, that you minimize equal to C transpose x plus Mx a. So, I am adding an artificial variable here and the corresponding constraint is subject to A x plus b minus A x naught, and I will explain why x a equal to b, where x and x a are greater than or equal to 0 and x naught, x naught greater than 0 is 0, some, some step, some point in R n.

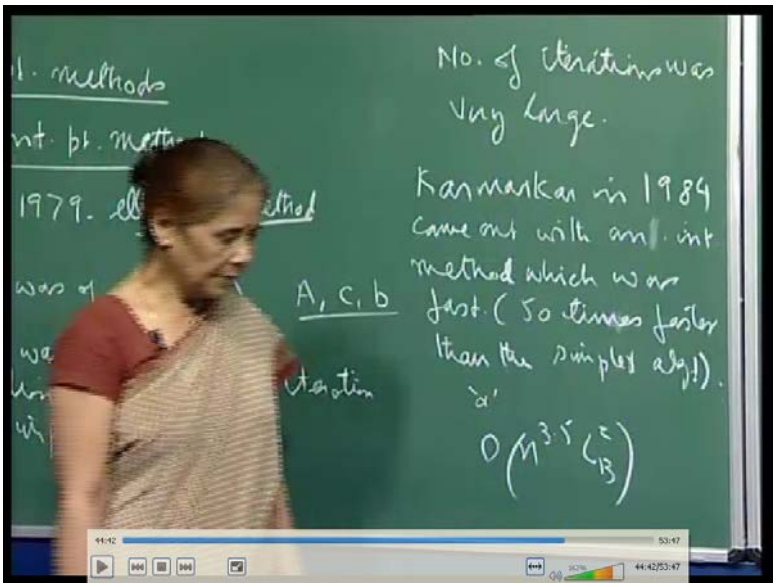
So, you see, here ambiguities are there and lot of experiences you need to really, you know, depending on how you choose your x naught, it will, but anyway first you look, look at the problem. So, here, you see, I have added an artificial variable and a x plus b minus a x naught times x a is b and here if you choose, so x naught greater than 0 and x a equal to 1.

So, this is, is a starting feasible interior point for this transform problem because right now, here, even to solve this problem by the interior point method, I need the starting feasible solution, which is an integer point. So, x a equal to 1 and x naught strictly positive, all components positive, is a feasible solution. Here, just see, because yeah, when I take x equal to x naught a x naught and there this is 1, so A x naught minus A x naught cancels, then you have b equal to b. So, this is a starting feasible point for the transform problem and now you use, again use, use Dikin's method and stop when x A is 0. The moment you get the artificial variable in x A is 0, you would have a feasible solution for A x is equal to b and you want a interior point. So, that is it.

So, therefore, once you get an interior starting feasible solution, you can continue go back to the original problem and continue working with the interior point. So, this was one of the primal methods. That means, you started with the primal feasible solution and then maintain feasibility and then stop when you thought that you had come close to the optimal solution.

(Refer Slide Time: 36:52)

Take slide grab at 42:08




Now, there are also dual methods, dual interior point methods, which we have not (( )) here. So, what they do is that they start with the dual feasible solution, again you see as in LPP, you, you may start with that dual problem if you have easily access to a dual feasible solution. And if we add an access to a primal feasible solution, we use the primal simplex algorithm and you saw the variation. So, same here, if you can find dual interior point feasible solution, then you may start with dual methods and of course, almost the same ideas persist and you can follow the algorithm.

Then, there are some infeasible, infeasible interior point methods and this again is, the idea here is that you take care of the, the problem of a not being able to find starting interior point feasible solution. So, you may start with the infeasible point, there is no problem. Then, again the question comes, if the, how far away from the feasible region you can start. So, all these tons of question are there.

But even then, sometimes (( )) shown, that some of these, some of these methods work and so you may start with the infeasible. So, then, interior point in the sense that infeasible methods, they are class of interior point methods, so, but you try to then move towards the boundary of the feasible region and the moment you hit the boundary, you have obtained an optimal solution because you keep improving the value of the objective function.

Now, let me come back to some recent developments, of course interior point methods have been there much before, Khachian's, see Khachian, Russian, Russian mathematician. In 1979, he gave his ellipsoid method, the famous ellipsoid method, where he kept shrinking the, the size of the ellipsoid in which the optimal solution, of course, he worked with the feasibility problem in which the feasible solution will lie and once the size of the ellipse became, you know, within a certain limit, then you could conclude, that you have an optimal solution because the volume of the ellipsoid was so small, so that it could only contain 1 feasible point.

And so, he, this is, this is a remarkable breakthrough in the sense, that, so far linear programming problems were not known to be, feasible, to be polynomial time, I will try to explain to you in the next lecture what all I mean by that. And so, his, he could actually show, that the linear programming problems can be solved in polynomial time, but the size of the complexity, so this algorithm, the complexity, complexity was of order n raise to 6 L B square, where you see, this is the number of bits, L B is the number of bits, that you require in the computer to store all the data. That means, they, this is your matrix A, c and b, so these information will, this is how you define an instance of a linear programming problem.

So, to store all these, this is number of bits square, now this was the polynomial size is high. So, the algorithm of course, was a breakthrough in the sense, that it showed that LPP's are polynomially solvable, but his method was the algorithm, was slow algorithm, was slow. In fact, the one iteration took longer than the, longer than the iteration of, of the simplex algorithm and the number of iterations was very large, number of iterations was very large; so, in fact, not at all a practical algorithm.

In fact, it is not implementable easily. So, as a theory, it was a beautiful theory because it showed, that a linear programming problems could be solved polynomially, which was not, which had not been shown earlier. So, and of course, the method, that separation method then so on, have given rise to so many other ideas of solving hard problems, that in any case, this was not implementable and even the earlier interior point methods, which were there, were not again very implementable because they, they would require all computations.

And Karmarkar, then Karmarkar in 1984 came out with an interior point, with an interior point method, which was fast. Actually, he claims, that it is 50 times faster than, which may be for some problems, not all, 50 times faster than the simplex method, than the simplex algorithm. This will be faster than the simplex algorithm; this is an exclamation because obviously, not for all problems you can show that the Karmarkar's will be faster than the simplex algorithm because simplex method has a very good record in any case. Karmarkar in 1984 came out with an interior point method, which again, yes, the same ideas that Hashing used, but he, he made it the, see, see the thing is that you have the, when you are computing the ellipsoidal, so on for the Hashing's algorithm, lot of computations are required. So, he simplified some of those things and then he also made the computation, for example, of alpha very simple and fast. So, he certainly made lot of very important improvements in the interior point method, existing interior point method and gave the algorithm, which did work.

And of course, the complexity of the algorithm order n 3.5, so from 6 he boarded down to 3.5 and the same, this thing because you require that many bits to store the instance of the problem. So, (( )) complexity was order n 3.5 and also, he could see at some point, when he came out with this algorithm, there was some large size linear programming problems, which could not be solved by the simplex algorithm because as you have seen, that the storage requirement for simplex is high and the, the existing in 1984, whatever computing facilities one had, the, you could not solve because you could not store the, the space and then space requirement, that simplex algorithm has was much higher than what the existing computing facilities could provide, and therefore, they were not, they could not be solved. But once the Karmarkar came with this algorithm, they could be solved very fast and so this was another breakthrough in the, in the, in the, in solving linear programming problems.

And so interior point methods have now come to stay and thousands of people research papers have come out and he also inspired lot of new ideas and therefore, they have been now used for solving difficult problems. So, this was the stage, that Karmarkar and started with Khachian, and then Karmarkar made significant improvements and contributions and the interior point algorithms have become a, considered to be now very efficient and it managed to solve large problems in reasonable amount of time.

So, I think this is what we want to say about the interior point methods and the, the exposition here is not been in anyway comprehensive or even you, just to give you an idea as to what goes behind the construction of interior point method and of course, the whole literature is around and one can hopefully, with whatever knowledge you have obtained about the linear programming theory and so on, and have some knowledge of linear algebra, you should be able to read the new interior point method and get a feeling for them, and may hopefully, software are all available.

So, once you, you had to need to know the theory and little bit for you can use the software for existing time. So, this is what basically I wanted to say because I thought, that a linear programming course must bring you up to date and as to, and to show you and to give you the latest developments. So, here I have again just limited myself to very basic presentation of the linear programming methods. So, let me discuss assignment 9, which is based on your, you know, Pert (( )) CPM.

So, the 1st problem, I have given you activities, the normal duration and the crash duration and then, the normal cost, which is in hundreds of rupees, that means, for activity 1 to 2, the normal cost, is 1200 rupees and when you want to crash the activity, it is 1500 rupees. So, you have to compute here in order to get the rate of crashing for each activity, you will compute it by that formula that for, for example, for the activity 2 to, it will be 15 minus 12 divided by 1.

(Refer Slide Time: 48:18)

Insert Slide here

So, which means, that it is 300 rupees right because activity 1, 2 can be crashed only by 1 day and so the cost of crashing, the rate of crashing is 300 rupees per day. Similarly, you will do it for the other activities. Then, I want you to, I am saying that 30th day is the due date for the competition, see the day, I am not giving you a due date.

So, I do not know whether 30th is the actual completion day of the project or it may be a due date. That means, it may be further than that and then, you want, I want you to compute the early and late start, and finish off each activity. See, idea through this example is that when I discuss the problem in the lecture, I took the due date as the day of the actual completion of the project.

So, here, I think I have made 30th little different from the actual completion time and so the early and the late start. So, for example, if 30th, if, if the project is completing before the 30th day, then you see, the late and early starts for critical activities will not be 0s, they will, there will be some slack there.

But that, because you are delaying the, you are saying the due date is beyond the actual completion of the project date, anyway just check for yourself. Then, I want, you find the optimum duration of the project, which is, you know, you are reducing the cost by crash or you are reducing the duration of the project by crashing it. And as I told you, that this is a trade-off between the direct cost in the, indirect cost, the indirect cost I have mentioned in the beginning is 1200 rupees per day.

So, if you are saying, in by crashing the activities is less than 1200 in a day, then you will go on crashing till the cost of crashing goes beyond 1200. In that case, you will have to, but then again you can, so you will stop, you will say, that the optimum duration is when the cost of crashing is less than or equal to the overhead cost of 1200 rupees per day. So, this is what we had discussed in the lecture.

(Refer Slide Time: 50:18)

Insert Slide here

So, let us look at the 2nd problem, this is a project, consists of 10 independent activities, this is Pert problem as shown in the table below.

So, the 1st time is your 6.5 days, which is the optimistic time. Then, M denotes your most likely time and these, the pessimistic time when things can go wrong and so for that project, the activity completion gets delayed. So, this one is a small network and I want you to compute the expected duration and variance for the project, as we had discussed in the lecture. So, for each, that means, you will find out the longest path respect to the expected duration of each activity and then variance, you add up the variances because independents holds.

Then the probability, that the project will get completed by the 34th day, 34th day should be, so the probability, remember and we said, that by central limit theorem you can reduce this probability to a standard normal variant and then compute it from using the

tables. What due date D the manager should suggest if you want the project to be completed by the Dth day with 0.95 probability. So, here, you will use the standard normal tables for computing the, whatever required probabilities you have, 2 in the 2 and 3.

4th, I am asking you to probability of completing the project on 33rd day, that is on the expected completion day is 0.5 always. If, if I, my due date is the same as t e, then it will be, see probability of z less than or equal to 0, which is 0.5 for a standard normal variant and the probability of completing the project on the 34th day is also close to 0.5.

So, now I want you to answer why is that, so that even on the 34th day the probabilities close to 0.5, that is because of high variance of one of the jobs in the critical path, but I want you to you know, for your, for yourself to work it out.

(Refer Slide Time: 52:25)

Insert Slide here

Yes, problem 3 now, these 3, 4 and 5 are related with the interior point methods and I already told you in the lecture, that I want you to show, that the projection matrix P has the property, that P squared is P and that P transpose is P because they are symmetric, then why does A A transpose inverse exist? I have already answered it, but anyway work it out for yourself.

Then, in the 5th problem, in the primal affine algorithm discussed in the lecture, show that if theta is less than or equal to zero, the LPP is unbounded. So, you have to show me, that there is a direction along which if you move, continue to move your, remain feasible and therefore, and the objective function value will decrease. And so, you will have the unboundedness situation being satisfied. So, I hope you enjoy doing this assignment sheet.