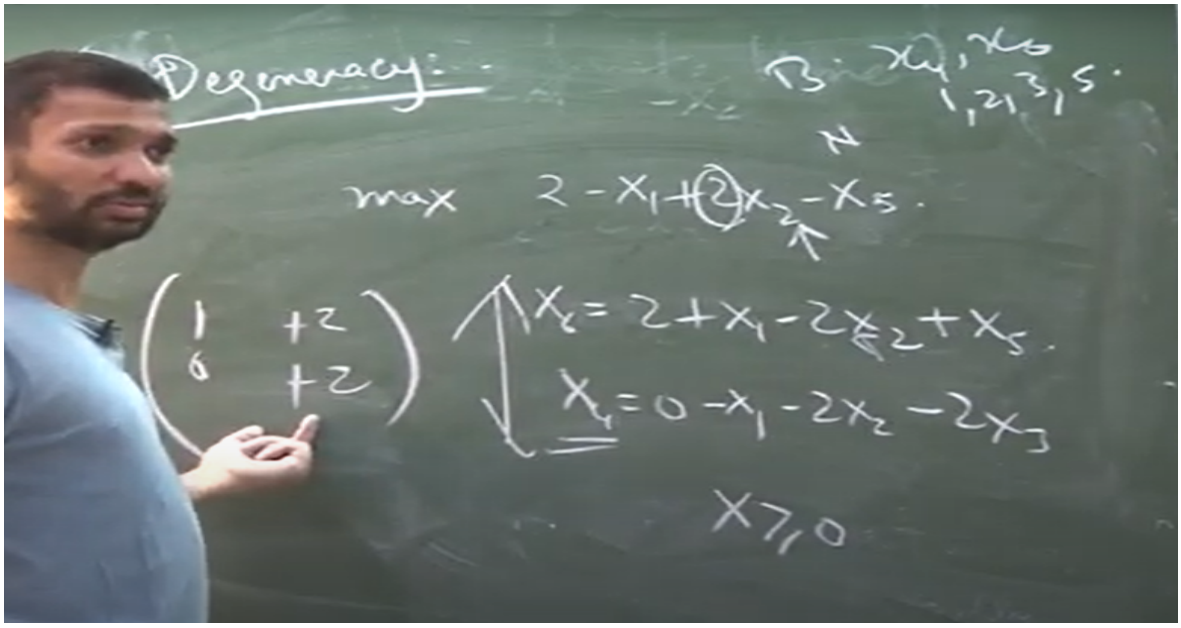
Linear Programming and its Applications to Computer Science
Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute Of Technology, Kanpur

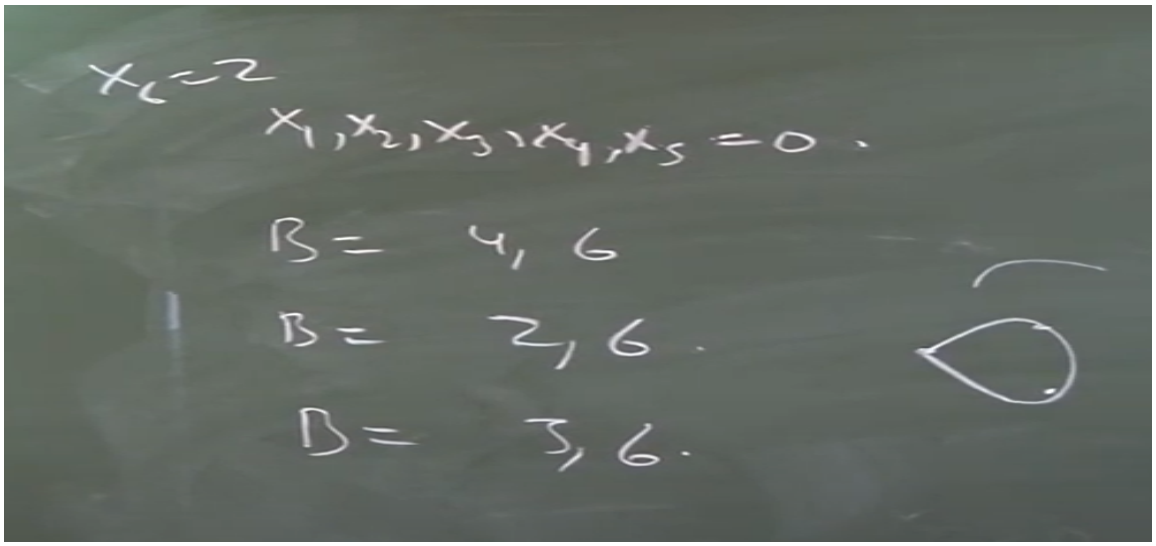
Lecture – 22
Degeneracy

So, just to again tell you that while you do all this. So, this is at some point I am stuck with this kind of an equation right. So, can you tell me what is the BFS? Exactly the one which is listed here. This is my B or 4 and 6 are my B 1 2 3 5 are my non basic right. At this point my objective value is 2. Now, when I look at the objective function this is clearly not locally optimal, because the coefficient of X 2 is positive.

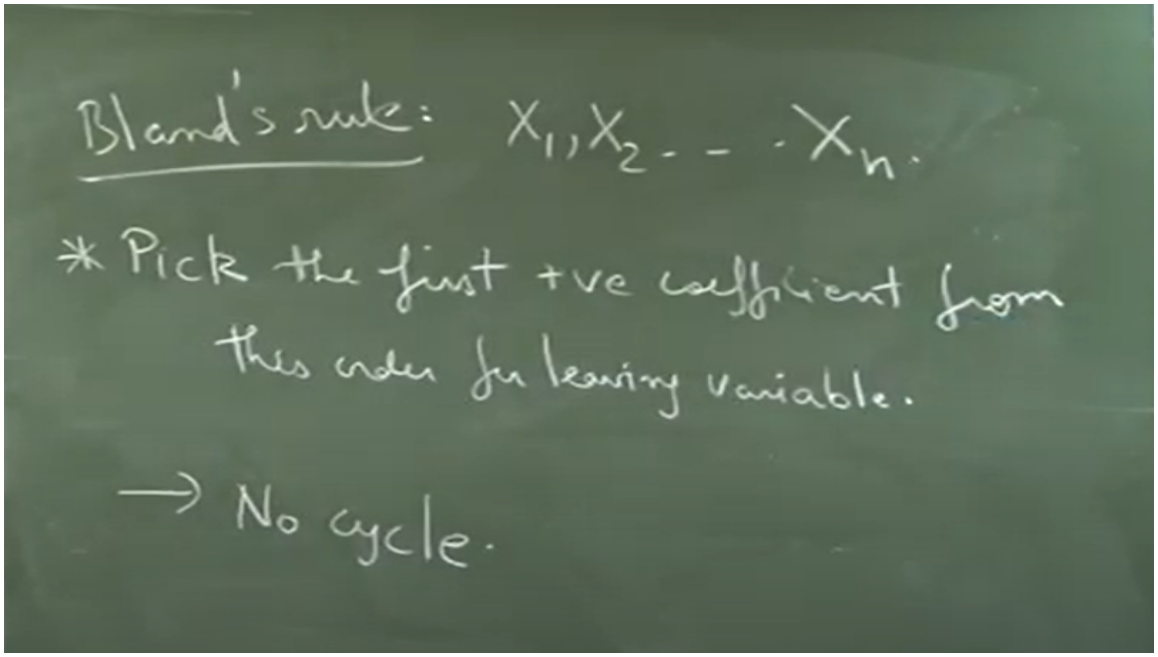


So, but then that means X 2 is the only possibility for me to be the leaving variable. But now yes here it is great I can increase X 2 to be equal to 1, but X 4 cannot be increased. So, if you actually look at this solution this solution is right. And then clearly this X 6 X 4 this is a diagonal matrix 1 and 0 this is invertible, but even the column correspond to X 2 X 6 and the column corresponding to X 2 this is also full rank.

This is like I think 1 0 minus 2 minus 2 or plus 2 plus 2 this is also full rank right. And not just that I think X 5 cannot come, but probably yeah even X 3 and X 6 are also basic feasible. So, the same vertex here the same solution here you can have 4 6 you can have 2 6 all those are invertible right. And all for those all those invertible I know that everything else is 0. So, it is a basic feasible solution right.



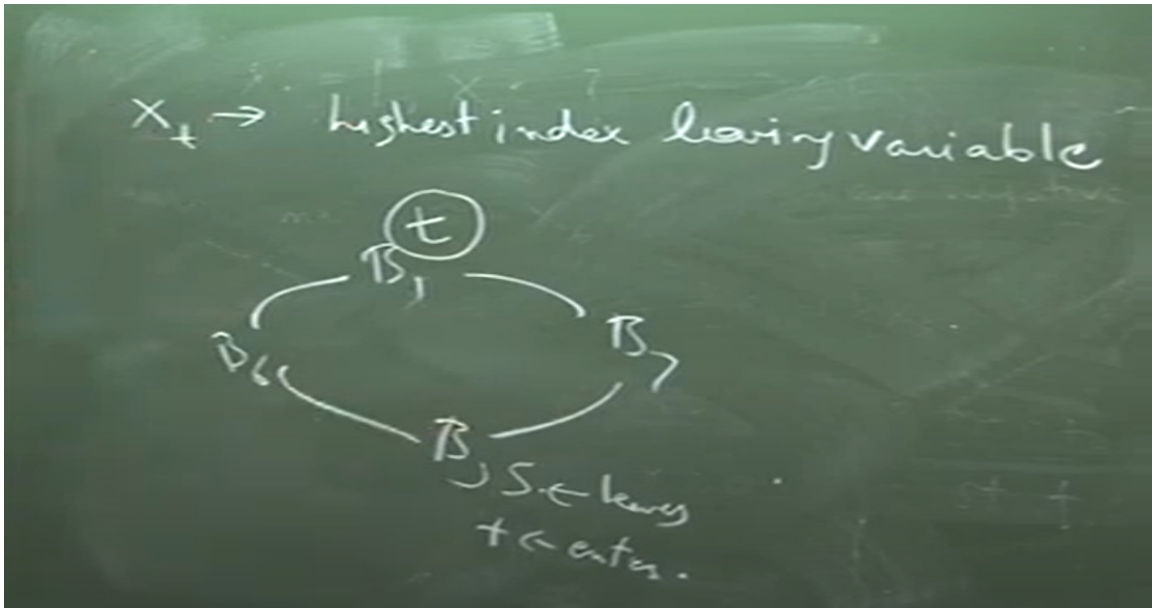
So, it is a serious concern what if we keep circulating between these 3 basic feasible solutions. So, it turns out that actually the implementation is not very difficult there is a simple solution which takes care of it. And so, I have given you an example where it can potentially get stuck if you are not smart enough if you just take a random choice of the leaving variables. But what you know is something called Blands rule it is artificial I am just going to tell you the Blanch's rule the proof is there in the notes it is technical and it does not give us any extra advantage you will just forget about it after this you will never encounter it. So, I do not want to go into the proof, but if you are interested you can see the proof it is like 2 paragraphs it is just.



So, I will tell you the idea of the proof also, but yeah what Blands rule does is. So, one thing could be you might say oh whatever is has the highest coefficient right in some sense that is the highest increase direction you might want to pick that it turns out it does

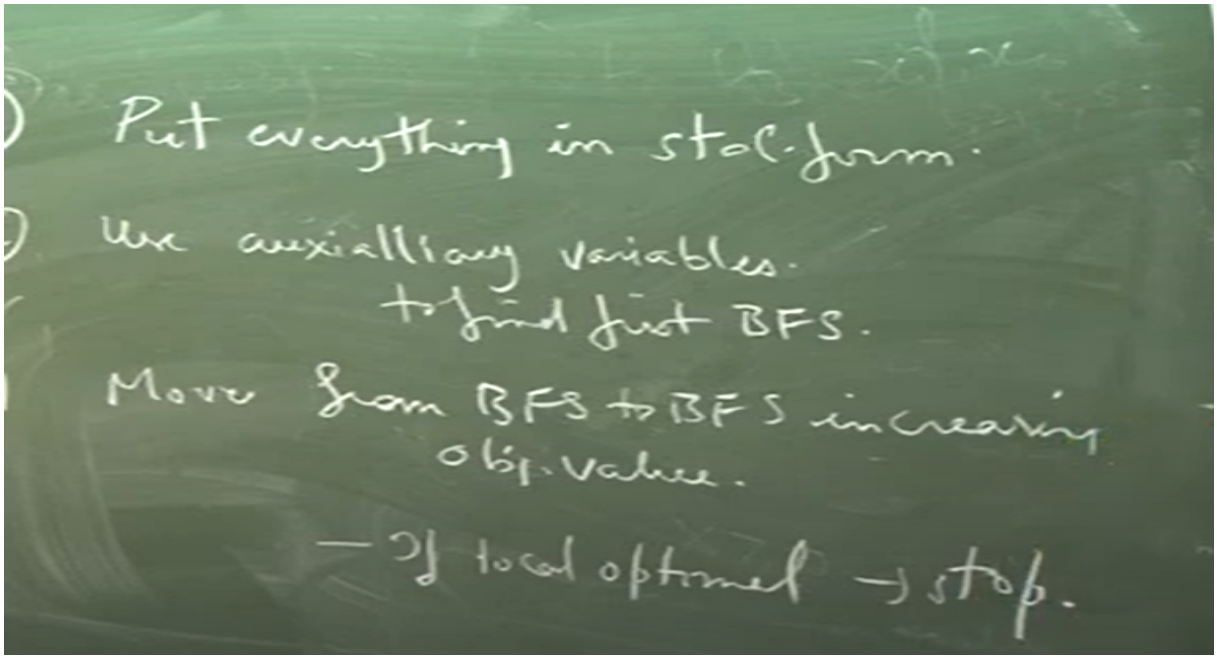
not work. So, what they say is you first decide on an ordering your favorite ordering whichever one you want right. Let us say this is the favorite ordering this is the 1 2 3 n then whichever look at all the coefficients which are positive for the non basic variables look at the first variable from that list which is a positive coefficient pick that. The amount of coefficient the value of the coefficient does not matter it is basically saying smallest variable here like smallest is a bad word, but smallest index variable whose coefficient is positive I am going to pick that why it works proof is there in the notes, but this is the.

So, if you fix that strategy always it is not a random choice of how you are deciding the leaving variable. So, what is the rule pick the first positive coefficient from this order leaving variable this is called Blands rule and if you do that you have a guarantee that you will not get stuck into any cycle you might be wondering why. So, what is happening? So, basically the way things work is that if now you have picked x 2 here next time you will not you will only see variables as a positive coefficient which are after that. So, if you have picked x 2 you will not circle back in some sense. So, if you look at the proof of this what will.



So, it will start by saying let us say x t is the highest leaving variable in your cycle, but how does it matter x t is the highest next leaving variable. That means, it is the highest index entering variable also right this is the cycle. That means, all variables if they leave they will enter once again. So, you have a cycle of b 1 b 2 b 3 b 6 or whatever and 1 variable leaves here at some point it has to enter it will just analyze this is where leaves this is where it enters let us say s is the leaving variable at b 1 b 2 b 3 b 6 whatever I do not know what I have written here, but this is where t enters again. So, you look at the 2 standard form equations here and here and you will basically argue that you know the coefficient of x s had to be positive here why did not you pick x s why did you pick x t

that is the kind of like analyzing all the coefficients.

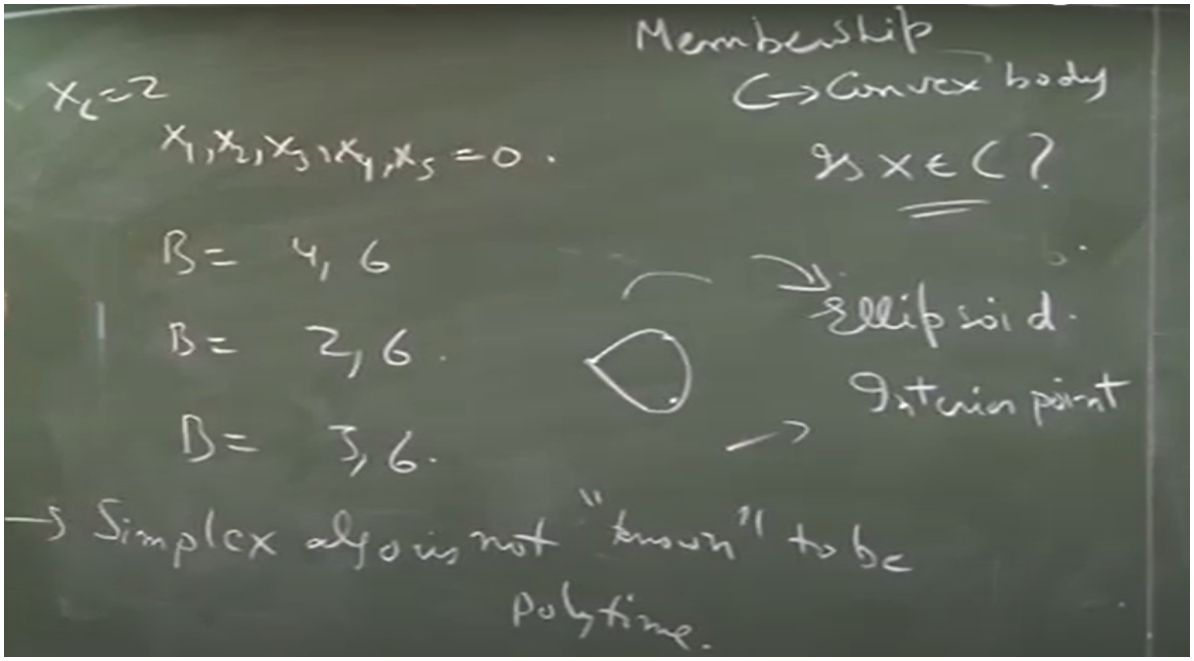


So, it was going in the right direction this is positive, but you know you have this multiple dependencies that for the leaving variable this coefficient has to be positive this variable has to be. So, for with all that you can do that, but again it does not give lot of things for future, but it is a possibility how do you avoid it sounds good. So, this finishes the idea of simplex algorithm. So, let me just write this implementation actually you can help me out the first is put everything in standard form I think I called auxiliary here. So, let me use auxiliary variables to find the first bfs if you want if you do not want to be very efficient or something you can just introduce auxiliary variables in all the equations who cares right.

And then just write minus x 1 minus x 2 minus y 1 minus y 2 up to minus y completely fine it does not matter. So, you use auxiliary variables for let us say each equation change your objective function solve the different LP to find the first bfs you are convinced that this works right. Then move from increasing objective only if at any point my BFS is locally optimal stop right and you know how to check that this is basically looking at the objective functions what are the coefficients. Any point how to move from bfs to bfs you know pick the positive constant probably with the ordering with blend rule in mind that is my leaving variable decide the leaving variable look at the equations see which variable is going to 0 first that is going to be your leaving variable. So, sorry you have entering variable and leaving variable.

So, find an entering variable depending on entering variable you decide the leaving variable you get to the new bfs you might go from a bfs to bfs where your vertex is not

changing, but will never be a cycle by blends rule. Since you keep increasing your value



infinite number of steps this will finish, but what we know is that for all the implementing there are programs there are linear programs which can circulate over exponential number of vertices. So, this is not known to be polynomial time can anyone tell me why I put codes on known because there is no 1 simplex algorithm right when I when we say this it means from this class of algorithm which utilizes this convexity properties is there an implementation like something like blends rule can I have 1 rule there depending on how I choose the leaving variable how I choose the entering variable probably that is how you become you know suddenly from exponential to polynomial time. We do not know whatever people have suggested have been short term that is the status, but thankfully we have 2 ellipsoid and interior point method. And ellipsoid method is polynomial time, but not very feasible because of numerical accuracy and things like that the idea there is that you it turns out not just for linear programs for any convex program you can solve them if you can solve the membership problem which is saying whether a point is inside my feasible region or not.

So, if you are given oracle for that you use it many times and then you can find an optimal solution that, but that is the idea behind ellipsoid method, but that membership question is something is going to be important for us from the next class itself. How do we find out whether a point is inside a convex area or not or how do we prove that a point is not inside a convex region. This question shift C is some convex body how do I prove it is not element of C right that is going to be the discussion from the next class. What people use is interior point method and I do not know if you are in a patriotic mood you can be proud because this was one of the main person who came up with this was karmarkar like an Indian dude he is now in puny I think, but he was the inventor of

one of those interior point methods which is used heavily which made him very very rich right. So, you are at the right class.