Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 42

Hello everyone, this is Lecture 2 of Week 9. Recall that in the last lecture, we learned certain characteristics of a linear programming problem. We learned that the constraint set of a linear programming problem will always be an n-dimensional polytope. We also learned that an optimal solution of a linear programming problem will always be a vertex of the feasible polytope.

Given these characteristics, George Dantzig proposed the simplex method, which involves mathematically characterizing the vertices of the linear programming problem. How do we do this?

The first step is to convert a given problem in the form:

Minimize $c^T x$ subject to $Ax \leq b$

into the standard form:

Minimize $c^T x$ subject to $A'x = b'$, $x \geq 0$

where A' is an $m \times n$ matrix with a rank equal to m.

I will illustrate this conversion with an example, and then we will understand how to do it in general.

Let us take our usual triangle example with the constraints:

1. $-x_1 + x_2 \leq 0$
2. $x_1 + x_2 - 1 \leq 0$
3. $-x_2 \leq 0$ (which is equivalent to $x_2 \geq 0$)

We want to convert these inequality constraints into equality constraints. We do this by introducing slack variables .

* For the first constraint, $-x_1 + x_2 \leq 0$, we add a non-negative slack variable x_3 to make it an equality:

  $-x_1 + x_2 + x_3 = 0$, where $x_3 \geq 0$.

This is equivalent because if $-x_1 + x_2$ is less than or equal to zero, we can add a positive number (x_3) to make it zero.

* For the second constraint, $x_1 + x_2 \leq 1$, we add a non-negative slack variable x_4:

  $x_1 + x_2 + x_4 = 1$, where $x_4 \geq 0$.

* The third constraint is already $x_2 \geq 0$.

We now have two equality constraints and three non-negativity constraints ($x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$). However, we are not done because the variable x_1 is unconstrained (it can be positive or negative). The standard form requires all variables to be non-negative.

To handle an unconstrained variable, we express it as the difference of two non-negative variables. We write:

$x_1 = x_1^+ - x_1^-$, where $x_1^+ \geq 0$ and $x_1^- \geq 0$.

* If x_1 is positive, x_1^+ equals x_1 and x_1^- equals 0.

* If x_1 is negative, x_1^+ equals 0 and x_1^- equals $-x_1$.

We now substitute x_1 with $(x_1^+ - x_1^-)$ in our equations:

1. $-(x_1^+ - x_1^-) + x_2 + x_3 = 0 \Rightarrow -x_1^+ + x_1^- + x_2 + x_3 = 0$

2. $(x_1^+ - x_1^-) + x_2 + x_4 = 1 \Rightarrow x_1^+ - x_1^- + x_2 + x_4 = 1$

Our variables are now: x_1^+, x_1^-, x_2, x_3, x_4, and all are non-negative .

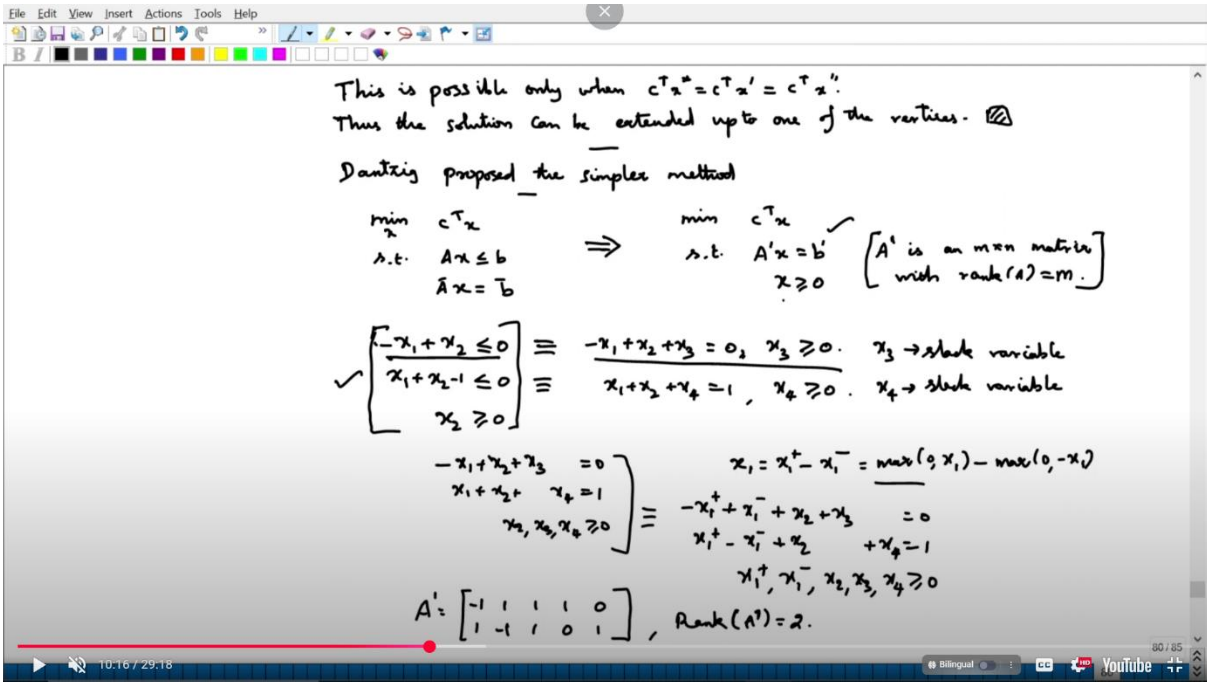
The constraint matrix A' is:

[-1, 1, 1, 1, 0]

[1, -1, 1, 0, 1]

This is a 2 × 5 matrix. The two rows are not multiples of each other, so the rank of A' is 2, which is equal to the number of constraints (m=2). We have successfully converted the problem to standard form.

(Refer Slide Time 10:16)



General Steps to Convert an LP to Standard Form

1. Convert maximization to minimization: If the problem is to maximize $c^T x$, convert it to minimize $-c^T x$.

2. Convert inequalities: All constraints must be "$\leq$". If you have a constraint $a_i^T x \geq b_i$, multiply by -1 to get $-a_i^T x \leq -b_i$.

3. Introduce slack variables: For each inequality constraint $a_i^T x \leq b_i$, add a slack variable $s_i \geq 0$ to create an equality: $a_i^T x + s_i = b_i$.

4. Handle unconstrained variables: For any variable x_i that is not constrained to be non-negative, replace it with $x_i = x_i^+ - x_i^-$, where $x_i^+ \geq 0$ and $x_i^- \geq 0$.

5. Ensure full row rank: The final matrix A' in A'x = b' must have full row rank (rank = m). If the rows are linearly dependent, eliminate redundant constraints until the remaining rows are linearly independent.

Characterizing Vertices in the Standard Form

We now have the standard form:

Minimize $c^T x$ subject to $Ax = b$, $x \geq 0$

where A is an $m \times n$ matrix with rank(A) = m. Since $m \leq n$, A is a "wide" matrix.

Because A has rank m, we can select m linearly independent columns from A. Let us form a matrix B from these m columns. B is an $m \times m$ invertible matrix. The remaining n-m columns form the matrix N .

We can partition the variable vector x accordingly:

* x_B: The "basic variables" corresponding to the columns in B.

* x_N: The "non-basic variables" corresponding to the columns in N.

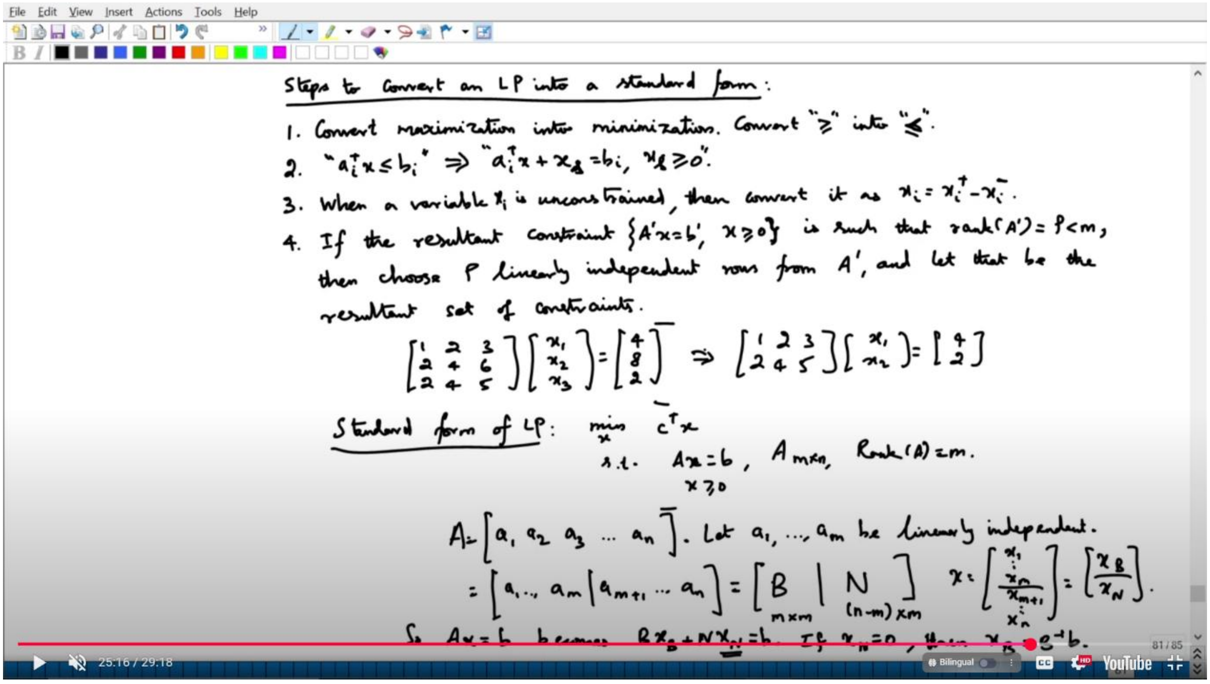
The equation $Ax = b$ can be rewritten as:

$B x_B + N x_N = b$

If we choose to set all the non-basic variables to zero ($x_N = 0$), we can solve for the basic variables:

$x_B = B^{-1}b$

(Refer Slide Time 25:16)

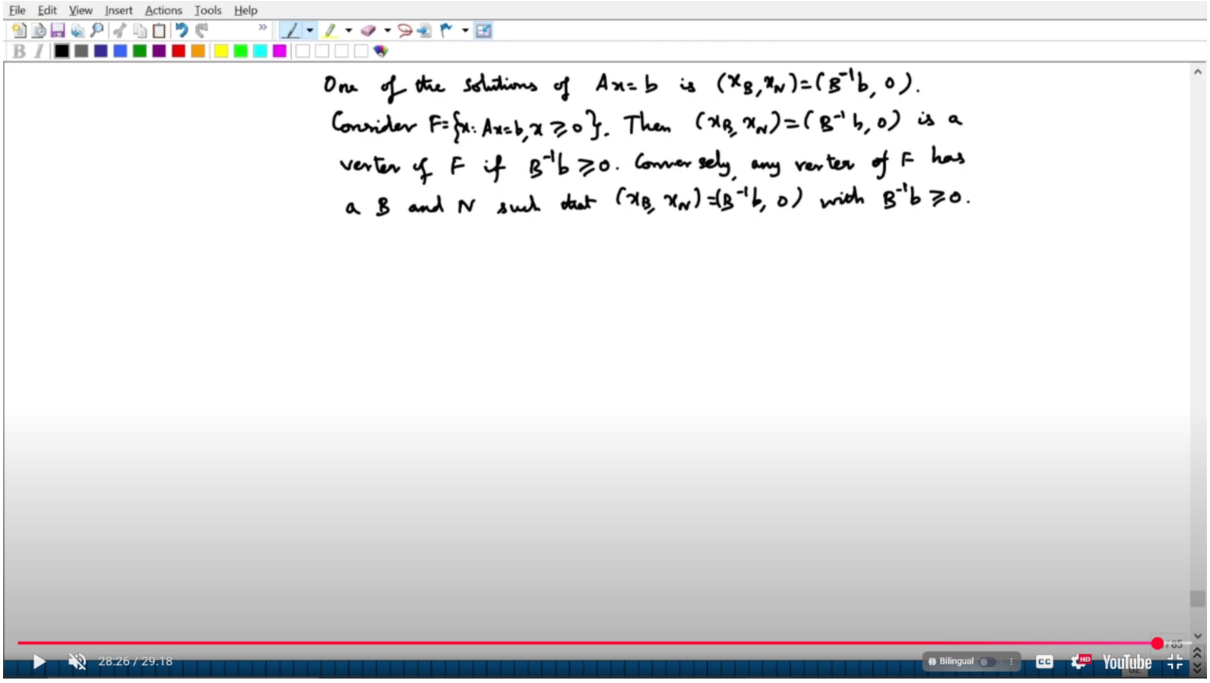


This gives us a solution of the form $(x_B, x_N) = (B^{-1}b, 0)$.

Why is this solution interesting? It is because any solution of this form is a vertex of the feasible set $F = \{x \mid Ax = b, x \geq 0\}$ (provided that $B^{-1}b \geq 0$).

Conversely, any vertex of the feasible set F can be represented in this form for some choice of basis B.

(Refer Slide Time 28:26)



This is the key insight of the simplex method: it systematically searches through these "basic feasible solutions" (the vertices) to find the one that minimizes the objective function.

We will continue this discussion and begin explaining the algorithm itself in the next class. Thank you.