Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan

Department of Mathematics

Institute of IIT Kanpur

Lecture: 43

Hello everyone. This is Lecture 3 of Week 9. Recall that in the previous two lectures, we learned about the simplex method. We first started with the linear programming problem and discussed some of its properties. We said that the constraint set will always be a polytope and that at least one of the vertices of this polytope will be an optimal solution.

We began the simplex method by first converting a given problem into its standard form. The standard form is:

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

where A is an $m \times n$ matrix with rank m.

After that, we saw how to split the matrix A into a set of basic and non-basic variables. We form a matrix B from m linearly independent columns of A.

The solution for the basic variables is then $B^{-1}b$, and the solution for the non-basic variables is 0. A solution of this form, $(x_B, x_N) = (B^{-1}b, 0)$, is a vertex of the feasible set, provided that $B^{-1}b \geq 0$. Conversely, any vertex can be written in this form.

Let us work with an example to understand how we choose basic and non-basic variables. We will use the triangle example with vertices at (0,0), (1,0), and (½,½). After conversion to standard form, the constraint matrix A is:

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

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

This matrix corresponds to the variables $[x_1^+, x_1^-, x_2, x_3, x_4]^T$.

The right-hand side vector b is $[0, 1]^T$, and all variables are non-negative.

This is a 2×5 matrix (m=2, n=5).

We need to choose 2 linearly independent columns from the 5 to form our basis B. Note that the first and second columns are linearly dependent (column 2 is -1 times column 1).

However, most other pairs of columns are linearly independent, giving us several possible choices for B.

For each choice of B, we compute the vertex $(B^{-1}b, 0)$ and check if $B^{-1}b \geq 0$.

Let's examine a few:

* B from columns 1 & 3 (x_1^+ and x_2):

  $B = [\,-1, 1; 1, 1\,]$, $b = [0; 1]$

  Solving $B * [x_1^+; x_2] = b$ gives $x_1^+ = ½$, $x_2 = ½$.

  The full solution is $x = (½, 0, ½, 0, 0)$.

  Since $x_1 = x_1^+ - x_1^- = ½ - 0 = ½$, this corresponds to the vertex (½, ½).

* B from columns 1 & 4 (x_1^+ and x_3):

  $B = [\,-1, 1; 1, 0\,]$, $b = [0; 1]$

  Solving gives $x_1^+ = 1$, $x_3 = 1$.

  The full solution is $x = (1, 0, 0, 1, 0)$.

  This corresponds to the vertex (1, 0).

* B from columns 1 & 5 (x_1^+ and x_4):

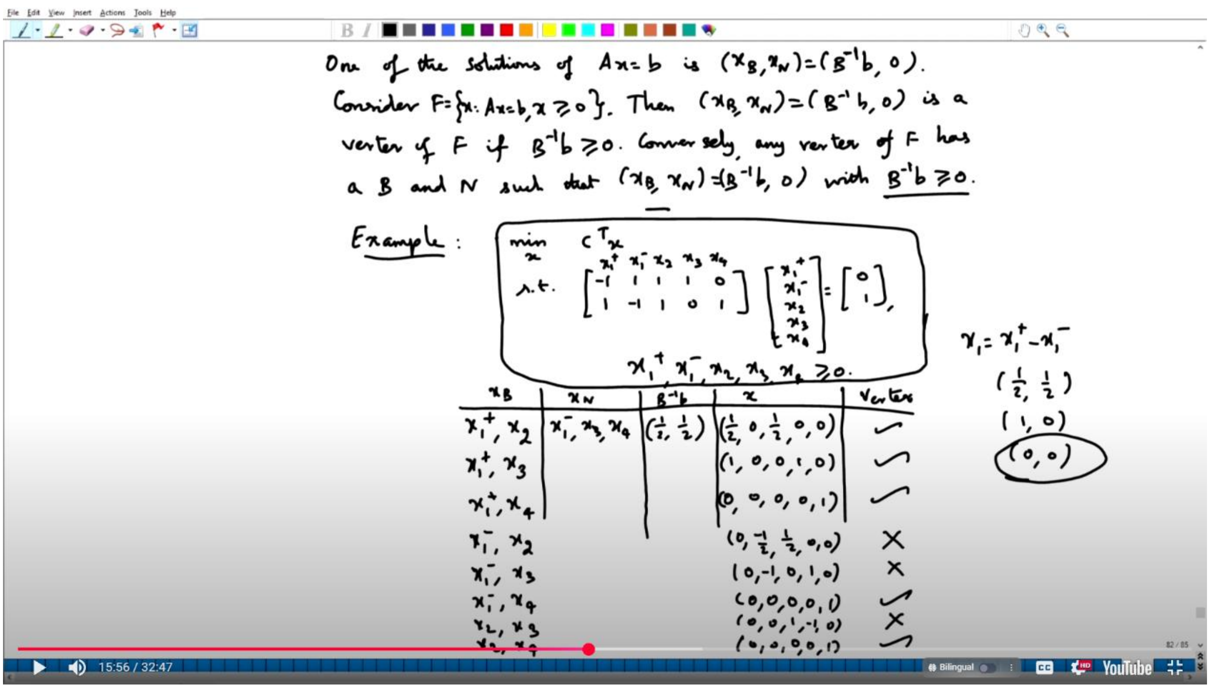
  $B = [\,-1, 0; 1, 1\,]$, $b = [0; 1]$

  Solving gives $x_1^+ = 0$, $x_4 = 1$.

  The full solution is $x = (0, 0, 0, 0, 1)$.

  This corresponds to the vertex (0, 0).

(Refer Slide Time 15:58)



Other choices for B either yield one of these three vertices again or result in $B^{-1}b$ having a negative component, which is an invalid solution for our non-negative constraints. This exercise shows that by choosing different sets of basic variables (different bases B), we can find all the vertices of the feasible polytope.

We now have a method to find all vertices. A brute-force solution would be to evaluate the objective function $c^T x$ at every vertex and choose the one with the minimum value. However, this is computationally infeasible for large problems.

For example, if n=100 and m=50, the number of possible bases (and thus potential vertices) is "100 choose 50", which is an astronomically large number (~10^{29}).

The simplex method provides a smarter approach. It starts at one vertex and moves to a "neighboring" vertex that improves the objective function, rather than checking all vertices.

The next step is to determine if a given vertex is the minimizer. We start with a basis B, giving us a vertex $x^* = (x_B, x_N) = (B^{-1}b, 0)$.

We rewrite the constraints and the objective function in terms of the non-basic variables:

From $Ax = b$, we have $B x_B + N x_N = b$. This can be rearranged as:

$x_B = B^{-1}b - B^{-1}N x_N$

The objective function is $c^T x = c_B^T x_B + c_N^T x_N$.

Substituting the expression for x_B, we get:

$c^T x = c_B^T (B^{-1}b - B^{-1}N x_N) + c_N^T x_N$

$= c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}N) x_N$

Let us denote the vector $(c_N^T - c_B^T B^{-1}N)$ as θ. This is a vector of length n-m, with each component corresponding to a non-basic variable.

The key insight is this: If every component of the vector θ is non-negative ($\theta \geq 0$), then the current vertex $(B^{-1}b, 0)$ is the minimizer.

This is because increasing any non-basic variable x_N from zero would only add a non-negative term to the objective function value $c_B^T B^{-1}b$, making it worse (larger).

The minimum is achieved by keeping $x_N = 0$.

Conversely, if any component of θ is negative, the current vertex is not optimal. We can improve (decrease) the objective function by increasing the corresponding non-basic variable from zero.

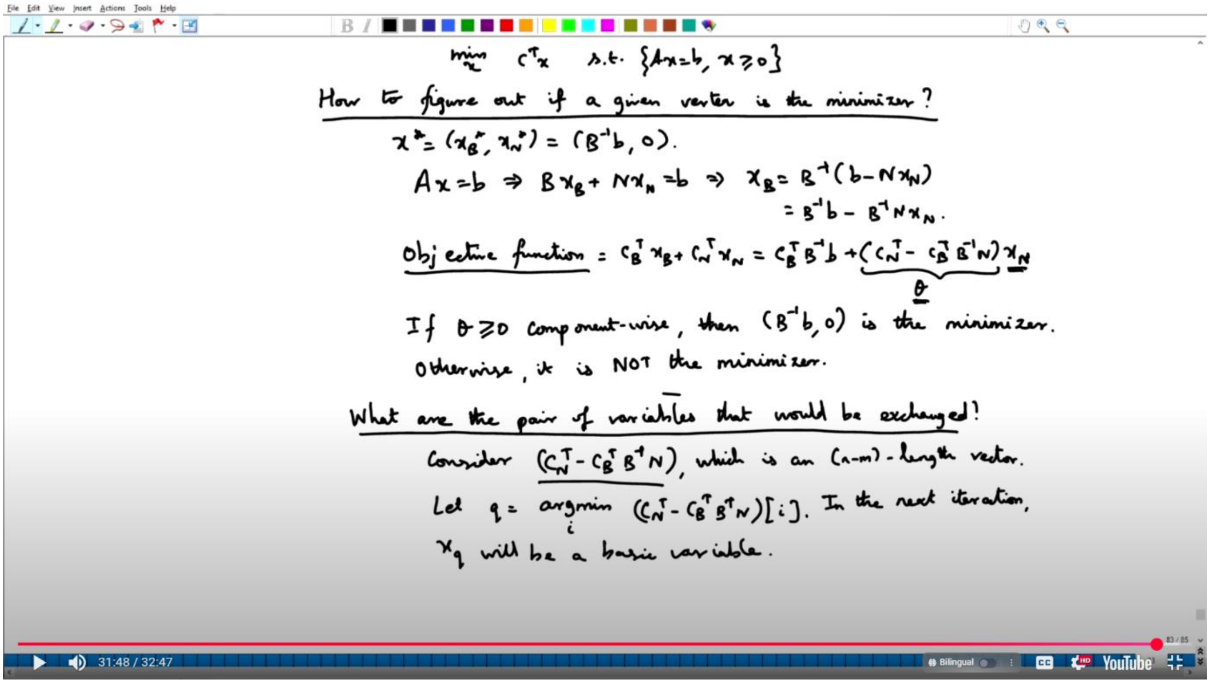
The next question is: which non-basic variable should we bring into the basis? The simplex method heuristic is to choose the variable that corresponds to the most negative component of θ. Let q be the index such that:

$q = \text{argmin over } i \text{ of } (\theta[i])$

In other words, we choose the non-basic variable x_q that gives the greatest rate of decrease in the objective function per unit increase.

We have now chosen a variable, x_q, to enter the set of basic variables. In the next iteration, it will become a basic variable.

(Refer Slide Time 31:48)



The final step, which we will cover in the next lecture, is to determine which current basic variable must leave the basis to make room for x_q, ensuring we move to a new, valid vertex.

Thank you.