Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 41

Hello everyone, this is the first lecture of week 9. Until now, we have learned various algorithms to solve unconstrained optimization problems and various algorithms to solve non-linear programs in constrained optimization. We will now get started with a few algorithms for linear programming problems in constrained optimization. Linear programming is a special case in constrained optimization, and since it is a special case, certain special techniques can be applied.

If you look at the chronology of when these algorithms were designed, you will see that the algorithm we are going to study, called the simplex method, was possibly the first algorithm invented for constrained optimization. All the penalty methods and augmented Lagrangian methods came much later. The simplex method was designed by Danzig in the 1940s, during World War II, for solving general linear programming problems.

Before going into the algorithm, I will explain in some detail the structure of linear programming problems and their solutions. Let us start with the general structure of a linear programming problem. It is of the following form:

minimize $c^T x$

subject to

$Ax \leq b$ and $\bar{A}x = \bar{b}$.

Here, A is a $p \times n$ matrix and $\bar{A}$ is an $m \times n$ matrix.

This means there are p inequality constraints and m equality constraints, all of which are linear. We will first examine the feasible set defined by linear constraints. This analysis holds even if the objective function is non-linear, as we are focusing solely on the constraint set. The constraint set is characterized by linear inequalities and linear equalities.

If a constraint set is characterized only by linear inequalities and linear equalities, how does it look? We have already seen some examples.

For instance, consider the set of constraints: $-x_1 + x_2 \leq 0$, $x_1 + x_2 - 1 \leq 0$, and $-x_2 \leq 0$.

This defines a triangle with vertices at (0,0), (1,0), and (½,½).

Let me give a few more examples.

Consider this example: $x_1 + x_2 - 1 \leq 0$, $2x_1 + 4x_2 - 3 \leq 0$, and $x_1, x_2 \geq 0$ or $-x_1 \leq 0$, $-x_2 \leq 0$.

There are four constraints in two dimensions.

I am using two dimensions so you can visualize the constraint set, as higher dimensions cannot be visualized. For $x_1 + x_2 - 1 \leq 0$, the x-intercept and y-intercept are 1.

For $2x_1 + 4x_2 - 3 \leq 0$, the x_2-intercept is ¾ and the x_1-intercept is 3⁄2.

The intersection of these constraints, along with $x_1 \geq 0$ and $x_2 \geq 0$, forms a quadrilateral.

The intersection point is (½,½), which satisfies both constraints: ½ + ½ - 1 = 0 and 2×(½) + 4×(½) - 3 = 1 + 2 - 3 = 0.

In a similar way, more examples can yield five-sided polygons (pentagons), six-sided polygons (hexagons), or any n-sided polygon. In two dimensions, the constraint set of a linear programming problem is always a polygon. For a three-dimensional problem with variables x_1, x_2, and x_3, the constraint set is called a polyhedron.

You might be familiar with tetrahedrons, which are a type of polyhedron with four vertices. In general, for n-dimensional linear programming problems, the constraint set is called a polytope. A polygon is the two-dimensional case, a polyhedron is the three-dimensional case, and a polytope is the general term for any dimension.

Why is this relevant? It is relevant because polytopes have vertices, and the solution of a linear programming problem is always at a vertex.

For example, in the triangle defined by the constraints $-x_1 + x_2 \leq 0$, $x_1 + x_2 - 1 \leq 0$, and $-x_2 \leq 0$, the solution must be one of the vertices: (0,0), (1,0), or (½,½), regardless of the coefficients in the objective function $c^T x = c_1 x_1 + c_2 x_2$.

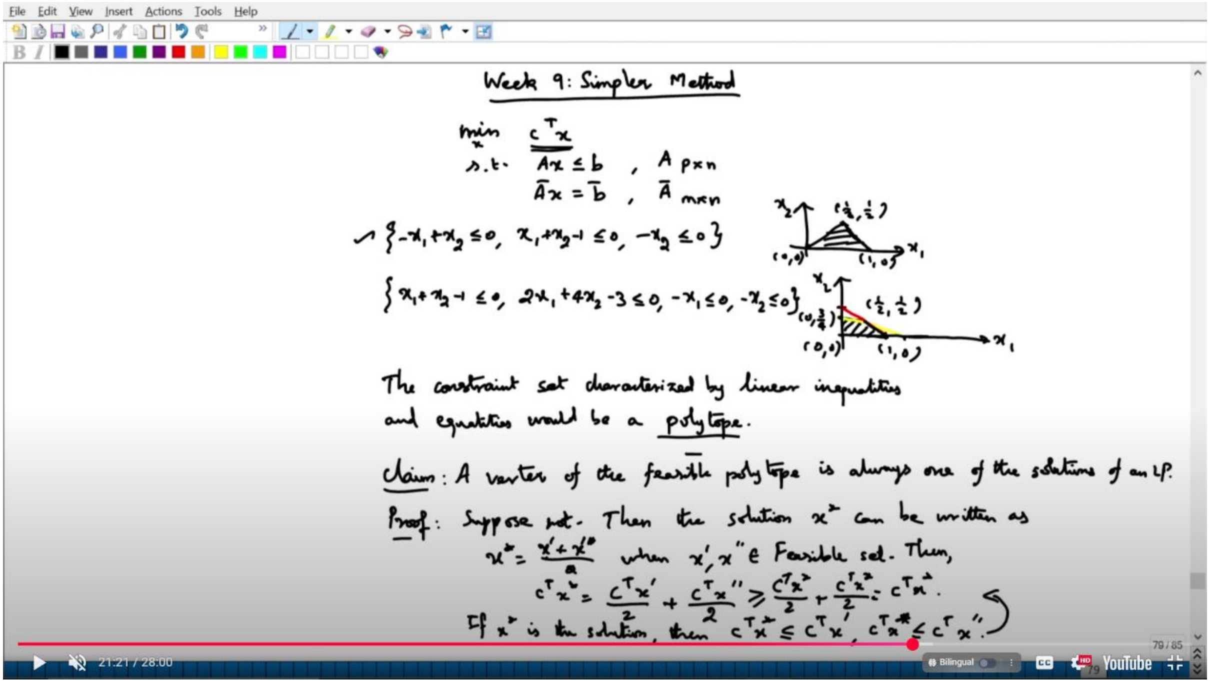
Let us prove this claim. Suppose the solution x^* is not a vertex. Then it can be written as $x^* = (x' + x'')/2$, where x' and x'' belong to the feasible set.

Since x^* is the solution, $c^T x^*$ is the minimum value.

Then, $c^T x^* = (c^T x' + c^T x'')/2$. However, since x^* is the minimum, $c^T x^* \leq c^T x'$ and $c^T x^* \leq c^T x''$. This implies that $c^T x^*$ must be equal to both $c^T x'$ and $c^T x''$.

Therefore, if the solution is unique, it must be a vertex. If there are multiple solutions, then at least one vertex is always a solution.

(Refer Slide Time 21:21)



This means we can find the solution by searching among the vertices. The vertex that gives the minimum value of the objective function is the solution. Now, the question is how to find the vertices, especially in higher dimensions where geometric visualization is impossible. For example, in problems with thousands of dimensions, we need a mathematical method to identify the vertices.

This is where Danzig's simplex method comes in. The first step in the simplex method is to convert the general linear programming problem

minimize $c^T x$

subject to

$Ax \leq b$ and $\bar{A}x = \bar{b}$ into a standard form.

The standard form requires that all constraints are equalities, and the only inequalities are the non-negativity constraints $x \geq 0$.

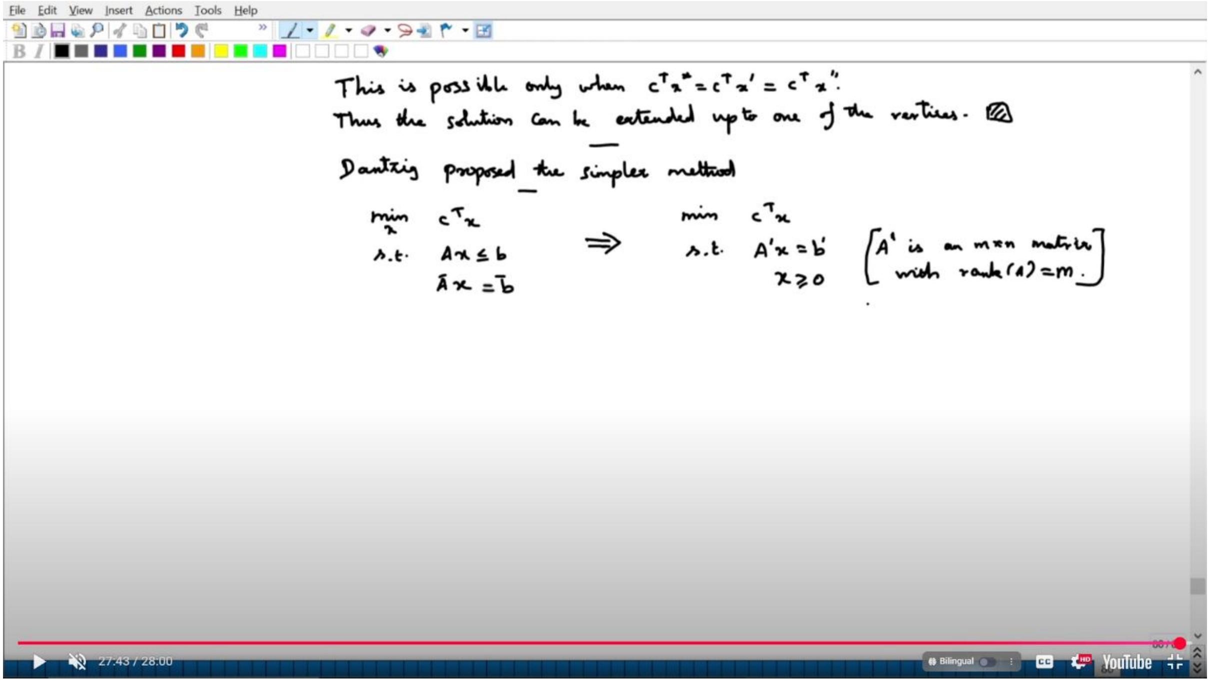
Specifically, the problem is transformed to

minimize $c^T x$

subject to

$A'x = b'$ and $x \geq 0$, where A' is an $m \times n$ matrix with rank m.

(Refer Slide Time 27:43)



The conversion process is the first step in the simplex method, and we will see how to do it in the next lecture. Thank you.