MAT 3007 – Optimization

Linear Optimization

Lecture 03

June 11th

Andre Milzarek

iDDA / CUHK-SZ

Repetition and Content

Recap: Linear Problems

Linear Optimization Problem:

$$
\begin{aligned}
\text{minimize/maximize}_{x\in\mathbb{R}^n} \quad & c^\top x \\
\text{subject to} \quad & A_1 x \geq b \\
& A_2 x \leq d \\
& A_3 x = e \\
& x_i \geq 0 \qquad \forall\ i \in N_1 \\
& x_i \leq 0 \qquad \forall\ i \in N_2 \\
& x_i \text{ free} \qquad \forall\ i \in N_3
\end{aligned}
$$

Standard form of a LP:

$$
\begin{aligned}
\text{minimize}_{x\in\mathbb{R}^n} \quad & c^\top x \\
\text{subject to} \quad & Ax = b \\
& x \geq 0
\end{aligned}
$$

- Homework 1 will be posted on Thursday, June 11th. It is due on Friday, June 19th, 11am.

Today's topics:

- Standard form for LPs (Continued).
- First implementations in MATLAB.
- Further examples and modeling techniques.
- Graphical solutions.

Linear Optimization: Standard Form

If the objective was maximization:

- Use $-c$ instead of c and change it to minimization.

Eliminating inequality constraints $Ax \le b$ or $Ax \ge b$:

- Write it as $Ax + s = b, s \ge 0$ or $Ax - s = b, s \ge 0$.
- We call s the slack variables.

If one has $x_i \le 0$:

- Define $y_i = -x_i$.

Eliminating "free" variables x_i (no constraints on x_i):

- Define $x_i = x_i^+ - x_i^-$, with $x_i^+ \ge 0$, $x_i^- \ge 0$.

$$\begin{array}{rllll} \text{maximize} & x_1 & +2x_2 & & \\ \text{subject to} & x_1 & & \leq 100 \\ & & 2x_2 & \leq 200 \\ & x_1 & +x_2 & \leq 150 \\ & x_1, & x_2 & \geq 0 \end{array}$$

Standard form:

$$\begin{array}{rllllll} \text{minimize} & -x_1 & -2x_2 & & & & \\ \text{subject to} & x_1 & & +s_1 & & & = 100 \\ & & 2x_2 & & +s_2 & & = 200 \\ & x_1 & +x_2 & & & +s_3 & = 150 \\ & x_1, & x_2, & s_1, & s_2, & s_3 & \geq 0. \end{array}$$

$$
\begin{array}{rll}
\text{minimize}_{w,b,t} & \sum_i t_i & \\
\text{subject to} & y_i(x_i^\top w + b) + t_i \geq 1, & \forall\ i \\
& t_i \geq 0 & \forall\ i.
\end{array}
$$

- Define $w = w^+ - w^-$, $b = b^+ - b^-$, with w^+, w^-, b^+,$b^- \geq 0$.
- Add slack variables to eliminate inequality constraints.

Standard Form for SVMs

$$
\begin{aligned}
\min_{w^+,w^-,b^+,b^-,t,s} \quad & \textstyle\sum_i t_i \\
\text{subject to} \quad & y_i(x_i^\top w^+ - x_i^\top w^- + b^+ - b^-) + t_i - s_i = 1 \quad \forall i \\
& w^+, w^-, b^+, b^- \geq 0 \\
& t_i, s_i \geq 0 \quad \forall\ i.
\end{aligned}
$$

Standard Form: Discussion

- Standard form is mainly used for analysis purposes. We do not need to write a problem in standard form unless necessary.
- Usually we just represent the problem in a way that makes it easy to understand.

- Transforming an LP into the standard form is an important skill. It is helpful for analyzing LP problems as well as when using software to solve it.

Linear Optimization: Implementation

In this course, we will mainly work with MATLAB to solve LPs and other optimization problems:

- Download a package called CVX.
- Read the instruction documents.
- `http://cvxr.com/cvx/`.

You may also use Python (cvxpy) or Julia (cvx.jl).

Example: Production Planning Problem

$$
\begin{array}{rrrl}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & x_1 & & \leq 100 \\
 & & 2x_2 & \leq 200 \\
 & x_1 & +x_2 & \leq 150 \\
 & x_1, & x_2 & \geq 0.
\end{array}
$$

Example: Support Vector Machine Problem

$$
\begin{array}{lll}
\text{minimize}_{w,b,t} & \sum_i t_i & \\
\text{subject to} & y_i(x_i^\top w + b) + t_i \geq 1, & \forall\ i \\
& t_i \geq 0 & \forall\ i.
\end{array}
$$

Example: Shortest Path Problem

Let $G = (V, E)$ be a graph where $V = \{1, ..., n\}$ is the set of nodes and E is the set of edges.

- We denote the source node by 1 and the terminal node by n.
- We use w_{ij} to denote the distance from i to j. In general, w_{ij} does not necessarily equal w_{ji} (it is a directed graph).
- We assume E contains all pairs of (directed) nodes: If there was no edge for (i, j), we can just set w_{ij} extremely large (e.g., larger than n times the maximum of the rest of w_{ij}).

We want to write a general shortest path solver using LPs:

- Input: A weight matrix $W = \{w_{ij}\}_{i,j=1,...,n}$.
- Output: The shortest path from 1 to n and its distance.

We have derived the optimization formulation for this problem:

$$\begin{array}{rl} \text{minimize} & \sum_{(i,j)\in E} w_{ij}x_{ij} \\ \text{subject to} & \sum_j x_{1j} = 1 \\ & \sum_j x_{jn} = 1 \\ & \sum_j x_{ij} = \sum_j x_{ji}, \quad \forall\ i \neq 1, n \\ & x_{ij} \in \{0,1\}, \qquad \forall\ (i,j) \in E. \end{array}$$

Example: Shortest Path Problem

For simplicity of implementation, we further include x_{ii} as decision variables and set w_{ii} to be very large.

Decision variables: a matrix $X = \{x_{ij}\}_{i,j=1,\dots,n}$

Objective function: $\sum_{i,j} w_{ij} x_{ij}$

- MATLAB representation: `sum(sum(W.*X))`.

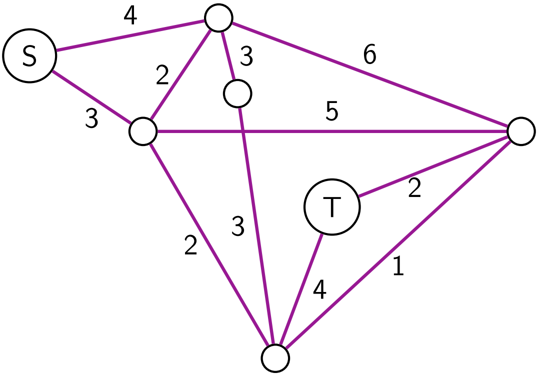
Constraints:

- `sum(X(1,:)) = 1`.
- `sum(X(:,n)) = 1`.
- `sum(X(i,:)) - sum(X(:,i)) = 0` for $i \neq 1, n$.

Integer constraints:

- We relax $x_{ij} \in \{0, 1\}$ to $0 \leq x_{ij} \leq 1$. For this problem, this will not change the solution ($\rightsquigarrow$ more later).

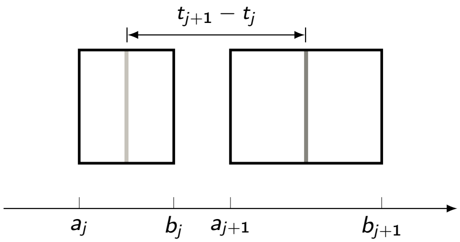
After discussing the general setup, we can solve a specific problem:



Modeling: Minimax Problems

An air traffic controller needs to control the landing times of n aircrafts:

- Flights must land in the order $1, \ldots, n$.
- Flight j must land in time interval $[a_j, b_j]$.
- The objective is to maximize the minimum separation time, which is the interval between two landings.



An Optimization Formulation

Decision variables:

- Let t_j be the landing time of the flight j.

Optimization problem:

$$\begin{array}{lll} \max & \min_{j=1,\ldots,n-1}\{t_{j+1} - t_j\} & \\ \text{s.t.} & a_j \le t_j \le b_j, & j = 1, \ldots, n \\ & t_j \le t_{j+1}, & j = 1, \ldots, n-1. \end{array}$$

Observation:

- The objective function is not a linear function. We call it a maximin objective.

We define

$$\Delta := \min_{j=1,\dots,n-1} \{t_{j+1} - t_j\}.$$

- Then by definition: $t_{j+1} - t_j \geq \Delta$ for all $j \rightsquigarrow$ use this for reformulation!

Write an LP:

$$\begin{array}{lll} \text{maximize}_{\Delta,t} & \Delta & \\ \text{subject to} & t_{j+1} - t_j - \Delta \geq 0, & j = 1, \dots, n-1 \\ & a_j \leq t_j \leq b_j, & j = 1, \dots, n \\ & t_j \leq t_{j+1}, & j = 1, \dots, n-1. \end{array}$$

- The optimal Δ must equal the minimal separation.
- This is called a maximin problem.

Minimax Objective

Similar to the air traffic control problem, we are also interested in a minimax objective:

$$\begin{aligned} \text{minimize}_x \quad & \max_{i=1,\dots,n}\{c_i^\top x + d_i\} \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

We can deal with it in a similar manner:

- Define $y = \max_{i=1,\dots,n}\{c_i^\top x + d_i\}$ and consider:

$$\begin{aligned} \text{minimize}_{x,y} \quad & y \\ \text{subject to} \quad & y \geq c_i^\top x + d_i \quad \forall i \\ & Ax = b \\ & x \geq 0. \end{aligned}$$

Modeling: Absolute Values

Dealing with Absolute Values

Problems with absolute values might be handled as well by LPs:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i=1}^n |x_i| \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

This can be equivalently written as

$$\begin{aligned} \text{minimize}_{x,t} \quad & \textstyle\sum_{i=1}^n t_i \\ \text{s.t.} \quad & t_i \geq x_i \\ & t_i \geq -x_i \\ & Ax = b. \end{aligned}$$

- Similar ideas can be applied for constraints like $|a^\top x + b| \leq c$.
- The splitting $x_i = x_i^+ - x_i^-$ for $x_i^+, x_i^- \geq 0$ can also be used for a different reformulation.

Consider the optimization problems:

$$\text{minimize}_x \; \textstyle\sum_{i=1}^n f_i(x) \quad \text{s.t.} \quad x \in \Omega \tag{1}$$

and

$$\text{minimize}_{x,t} \; \textstyle\sum_{i=1}^n t_i \quad \text{s.t.} \quad x \in \Omega, \quad f_i(x) \le t_i, \quad \forall\; i, \tag{2}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ are given and $\Omega \subset \mathbb{R}^n$ is the feasible set.

Lemma: Modeling Tool

The problems (1) and (2) are equivalent in the following way:

- If x^* is a sol. of (1), then $(x^*, f(x^*))$ is an opt. sol. of (2).
- If (x^*, t^*) is an opt. sol. of (2), then x^* is an opt. sol. of (1).

In this case, both problems have the same optimal value.

Consider a similar problem:

$$\begin{array}{rl} \text{maximize} & \sum_{i=1}^{n} |x_i| \\ \text{s.t.} & Ax = b. \end{array}$$

Can we use the similar idea and transform it into:

$$\begin{array}{rl} \text{maximize} & \sum_{i=1}^{n} t_i \\ \text{s.t.} & t_i \geq x_i \\ & t_i \geq -x_i \\ & Ax = b. \end{array}$$

- Answer: No. There is some intrinsic property that prevents us from formulating it as an LP (non-convexity). We will talk about it later in this course.

Modeling: Fractional Programming

Consider the problem:

$$\begin{aligned} \text{minimize}_x \quad & \frac{c^\top x + d}{e^\top x + f} \\ \text{s.t.} \quad & Ax \le b. \end{aligned}$$

- We assume that $e^\top x + f > 0$ for any x satisfying $Ax \le b$.
- Production Planning: The objective of the company might be based on maximizing the ratio: (*total profit*)/(*total production costs*).

Any idea to transform it to LP?

- Define:

$$y = \frac{x}{e^\top x + f}, \qquad z = \frac{1}{e^\top x + f}.$$

We can write the problem as:

$$\begin{aligned} \text{minimize}_{y,z} \quad & c^\top y + dz \\ \text{s.t.} \quad & Ay - bz \le 0 \\ & e^\top y + fz = 1 \\ & z \ge 0 \end{aligned}$$

- This is an LP!
- Why are they equivalent?

$\rightsquigarrow$ See Boyd and Vandenberghe for details (supplemental reading: page 151).

Recap

Up to now, we have learned how to:

- Formulate linear optimization problems.
- Transform an LP into a standard form.
- Use MATLAB to solve linear optimization problems.

Next:

- How to solve linear optimization problem?
- We will start with some basic properties of LPs.

Solving LPs Graphically

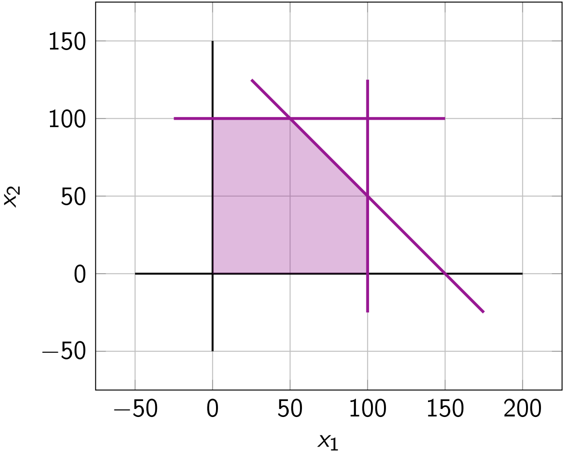
It is very helpful to study a small LP from a graphical point of view.

Recall the production problem:

$$\begin{array}{rrrl}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & x_1 & & \leq 100 \\
& & 2x_2 & \leq 200 \\
& x_1 & +x_2 & \leq 150 \\
& x_1, & x_2 & \geq 0.
\end{array}$$

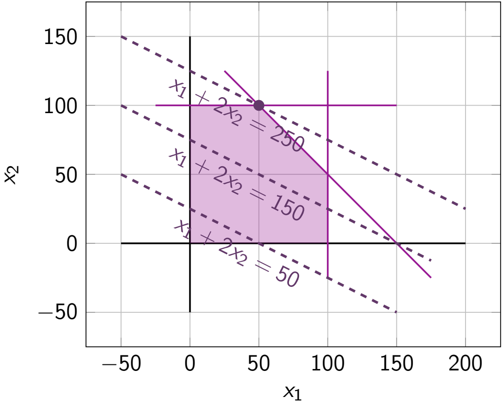
How can we solve this using a graph?

We first draw the feasible region.



To Maximize $x_1 + 2x_2$ ...

We then draw the function $x_1 + 2x_2 = c$ for different values of c.



- The optimal solution is the highest one among these lines that touches the feasible region.
- Optimal solution: $(50, 100)$. Objective value: 250.

- The feasible region of an LP is a polyhedron.
- The optimal solution tends to be a corner of the feasible region.
- Some constraints are active at the optimal solution ($x_2 \leq 100$, $x_1 + x_2 \leq 150$), some are not ($x_1 < 100$).

We will formalize these observations and study algorithms for solving LPs that can:

- Guarantee to find the optimal solution.
- Run within a certain (reasonable) amount of time.

Some Definitions: Polyhedron

Polyhedron

A polyhedron is a set that can be written in the form:

$$\{x \in \mathbb{R}^n : Ax \geq b\},$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$.

- Recall that in the standard form of LP, the feasible set is

$$Ax = b, \quad x \geq 0.$$

- Is this a polyhedron? Why?
- Yes, we can write it as $Ax \geq b$, $Ax \leq b$, $I \cdot x \geq 0$ where I is the identity matrix.

Convex Sets and Convex Combinations

Definition: Convex Set

A set $S \subseteq \mathbb{R}^n$ is convex if for any x, $y \in S$, and any $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in S$.

Convex Combination

For any $x_1, ..., x_n$ and $\lambda_1, ..., \lambda_n \geq 0$ satisfying $\lambda_1 + \cdots + \lambda_n = 1$, we call $\sum_{i=1}^n \lambda_i x_i$ a convex combination of $x_1, ..., x_n$.

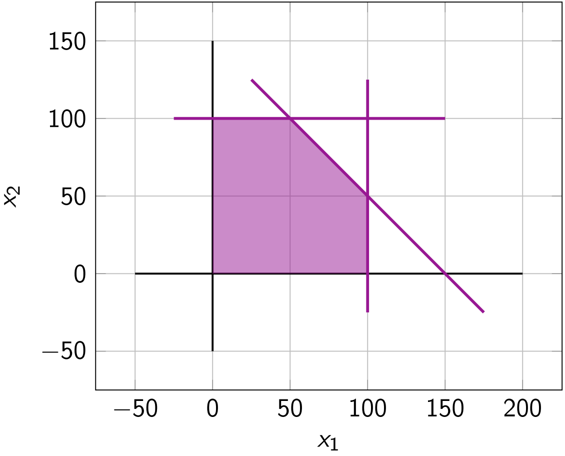
- We noticed that in an LP, the optimal solution tends to be in one of the corners of the feasible region. We first formalize this notion.

Definition: Extreme Point

Let P be a polyhedron. A point $x \in P$ is said to be an extreme point of P if we can not find two vectors $y, z \in P$ with $y, z \neq x$ and a scalar $\lambda \in [0, 1]$, such that $x = \lambda y + (1 - \lambda) z$.

- That is, x cannot be represented as a convex combination of other points in P.
- We sometimes call the extreme point the vertex or corner of the polyhedron.

Example: Extreme Points



How many extreme points are there in this feasible region?

- Answer: 5

Extreme Points

We just introduced the definition of the extreme points/vertices.

However, this does not tell us how to find those points. We want to have a good way for finding extreme points.

Preview of the next lectures:

- We will introduce an algebraic way to represent extreme points.
- We will show that it is sufficient to look at extreme points to solve a linear optimization problem.
- Finally, this will lead to the construction of the simplex algorithm for solving LPs.

Questions?