CO327 Deterministic OR Models (2021-Spring)

Introduction / Quick review of Linear Programming

Instructor: Andersen Ang

Combinatorics and Optimization, U.Waterloo, Canada

msxang@uwaterloo.ca, where $\mathbf{x} = \lfloor \pi \rfloor$

Homepage: angms.science

First draft: December 26, 2020 Last update: May 26, 2021

What is optimization

- A general optimization problem

$$\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \mathcal{C}.$$

- Short notation

$$\min_{\mathbf{x}}\ f(\mathbf{x}) \ \text{ s.t. } \ \mathbf{x} \in \mathcal{C}.$$

  - f: objective function / cost function
  - $\mathbf{x}$: optimization variable / decision variable / unknown
  - $f(\mathbf{x})$: objective function value at $\mathbf{x}$ / cost at $\mathbf{x}$
  - $\mathbf{x} \in \mathcal{C}$: constraint description
  - $\mathcal{C}$: the constraint set / feasible set
  - $\min$: you want $f(\mathbf{x})$ to be as small as possible

- About min and max: they are the opposite.

Narrow down the scope: we consider

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathcal{C}.$$

- $\mathbf{x}$ is a vector in $\mathbb{R}^n$, $n < \infty$
  - i.e. we consider problem with *finitely* many variables.
  - What's the opposite: $n = \infty$, $\mathbf{x}$ is a function.
- f is a deterministic function of $\mathbb{R}^n \to \mathbb{R}$.
  - No randomness in f
  - What's the opposite: stochastic programming
- $\mathcal{C} \subseteq \mathbb{R}^n$
  - $\mathcal{C}$ is a subset of $\mathbb{R}^n$.
  - In integer programming $\mathcal{C}$ is a subset of $\mathbb{Z}^n$.

What is Linear Optimization (LO)

- In LO, f and $\mathcal{C}$ are linear.
- f is linear:

$$
\begin{aligned}
f(\mathbf{x}) &= c_1x_1 + c_2x_2 + \cdots + c_nx_n && \text{linear combination of variables} \\
&= \sum_{i=1}^{n} c_ix_i && \text{compact notation} \\
&= \mathbf{c}^\top \mathbf{x} && \text{vector notation 1} \\
&= \langle \mathbf{c}, \mathbf{x} \rangle && \text{vector notation 2}
\end{aligned}
$$

- Example: 2-variable: $n = 2$, $\mathbf{x} = [x_1, x_2]^\top$ and $\mathbf{c} = [\pi, e]^\top$

$$
f(\mathbf{x}) = \pi x_1 + e x_2 = \sum_{i=1}^{2} c_ix_i = \begin{bmatrix} \pi \\ e \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \langle \mathbf{c}, \mathbf{x} \rangle.
$$

- Question: what about affine function?

What is Linear Optimization (LO)

- In LO, f and $\mathcal{C}$ are linear.

- $\mathcal{C}$ is linear. Let

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \circ_1 b_1. \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \circ_2 b_2. \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \circ_m b_m. \end{aligned} \iff \begin{aligned} \mathbf{a}_1^\top \mathbf{x} \circ_1 b_1. \\ \mathbf{a}_2^\top \mathbf{x} \circ_2 b_2. \\ \vdots \\ \mathbf{a}_m^\top \mathbf{x} \circ_m b_m. \end{aligned}$$

  where $\circ \in \{\geq, =, \leq\}$.

- Compact notation: let $\mathbf{a}_i$ be the ith row of a matrix $\mathbf{A}$ and $\mathbf{b} = [b_1, b_2, \ldots, b_m]$ and $\circ = [\circ_1, \circ_2, \ldots, \circ_m]$ then $\mathbf{A}\mathbf{x} \circ \mathbf{b}$.

- Canonical form $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{A}\mathbf{x} \leq \mathbf{b} \right\}$.

- Standard form $\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0 \right\}$.

What does “linear” means and why linear

- Proportionality
  If you double the amount of the decision variable x_1, its effect on the constraint and cost are scaled twice.

- Easiest modeling technique by human.

- Non-linear model: you have $\cdot^2$, $\cdot^3$ or a polynomial.

- Changing from linear to non-linear, things can become “crazily difficult” to solve: NP-hard!

- In fact, Linear program can be solved in polynomial time[1].

---

[1] N. Karmarkar A New Polynomial Time Algorithm for Linear Programming. Combinatorica, 1984.

Linear geometry

- In $\mathbf{A}\mathbf{x} \le \mathbf{b}$, each row is either $\mathbf{a}^\top \mathbf{x} = b_i$ or $\mathbf{a}^\top \mathbf{x} \le b_i$.
- $\mathbf{a}_i^\top \mathbf{x} = b_i$ is an affine hyperplane.
- $\mathbf{a}_i^\top \mathbf{x} \ge b_i$ is a half-space.
- $\mathcal{C}$ is described by a finitely many linear hyperplanes and half-spaces.
  - $\mathcal{C}$ can be unbounded (not so interesting).
  - $\mathcal{C}$ can be bounded: polyhedron (more interesting).
- The cost $\mathbf{c}^\top \mathbf{x}$ is a linear functional and $\mathbf{c}^\top \mathbf{x} = f^*$ is an affine hyperplane with a particular cost value f^*.
- Gradient of $\mathbf{c}^\top \mathbf{x}$ is $\mathbf{c}$.
- So geometrically, LP $=$ sliding a hyperplane on a polyhedron to find extreme value.

An example

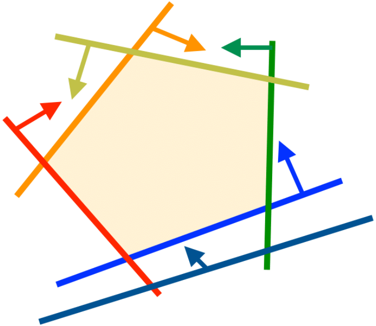

Figure: A polygon described by 5 half-spaces. Source: internet.

Type of LP

- Infeasible LP
  The LP is unsolvable: there is no solution that satisfies all the constraints.

- Feasible LP (Feasible LP with bounded optimal cost)
  The LP is solvable: there is (at least one) finite optimal sol. that satisfies all the constraints.
  - Type 1. The optimal sol. is unique (one and only one).
    Geometry: optimal sol. at a corner of the feasible region.
  - Type 2. The optimal sol. is non-unique (many).
    Geometry: optimal sol. at an edge (facet) of the feasible region.

- Unbounded LP (Feasible LP with unbounded optimal cost)
  The LP is solvable: but the optimal sol. is at ∞.

- Generalization: Fundamental Theorem of Linear Programming (later).

A long sentence to describe what is LP

A (deterministic) linear program is an (deterministic) optimization problem with finitely many variables on a (deterministic) linear objective function with a constraint set defined by a finitely many (deterministic) hyperplanes and/or half-spaces.

Real-life vs textbook

- The "deterministic" in the long sentences are important.
  - Real-life: problems are *uncertain*.
  - Textbook: perfect toys with no uncertainty.

  There is a research area called "uncertainty quantification"

- Also
  - textbook: most probably stick with $n = 2$ (2D cases) with only x_1, x_2, which is easier to explain, easier to visualize and easier to understand.
  - Real-life: $\mathbf{x}$ can be in $\mathbb{R}^{99\cdots 9}$.

  In fact, even with $n = 3$, if m (number of constraints) is big, the polyhedron becomes difficult to visualize.

- We don't deal with stochastic linear programming in this course, but a way to deal with uncertainty is to perform *Sensitivity analysis*: study how the (optimal) value and sol. change as the problem changes (a little bit).

Remark and connection to linear algebra

- LP in standard form:
$$\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x} \ \text{ s.t. } \ \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0.$$

- Important note: $\mathbf{A}$ is singular / non-invertible.
$\mathbf{A}$ has more columns than rows / $\mathbf{A}$ is a short-fat matrix.

- Why: otherwise $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ (unique sol.) and we solved the problem.

- Linear algebra: for short-fat matrix, the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a vector space of solutions (many sol. / no unique sol.).

- So LP = try all the candidate sol. from such vector space of solutions, and pick the one/those that minimize(s) the cost.

How to solve LP

- Graphical method: only for toy problems.
- Simplex method and dual simplex method.
- Ellipsoid method
  - Mathematically an interesting method for solving LP.
  - Poor performance in practice.
- Interior-point method
  - Theoretically and practically the fastest method for large LP.
  - $\mathcal{O}(n^{3.5})$ vs $\mathcal{O}(2^n)$ for worst case for simplex method.

*CO327 focuses on models $\implies$ not focus too much on methods.

History

- 1947 Dantzig: Simplex method
- 1972 Victor Klee and George Minty: "Worst problem for simplex method". In $\mathbb{R}^n$,

$$\begin{array}{ll} \min & x_n \\ \text{s.t.} & 0 \leq x_1 \leq 1 \\ & \epsilon x_{j-1} \leq x_j \leq 1 - \epsilon x_{j-1}, \quad j \in \{2, \dots, n\}, \; 0 < \epsilon < \dfrac{1}{2} \end{array}$$

  In star with $[0, 0, \dots, 0, 1]^\top$, simplex method will visits all those 2^n vertices $\implies$ simplex method in the worst case has non-polynomial complexity $\mathcal{O}(2^n)$.
- 1979 Leonid Khachiyan: Ellipsoid algorithm $\mathcal{O}(n^6)$.
- 1984 Karmarkar: Interior Point Method $\mathcal{O}(n^{3.5})$
- 2019 Cohen-Lee-Song: $\mathcal{O}\big(n^\omega \log(n/\delta)\big), \omega \sim 2.37$

Summary

- Review and overview on what is LP.

Next : go to tutorial 1.

End of document