Spring 2023

Linear programming I

Modeling mathematical programs

Cathy Wu

1.041/1.200 Transportation: Foundations and Methods

References

1. Readings: AMP - Chapter 1 Mathematical Programming: an overview
   Companion slides of Applied Mathematical Programming by Bradley, Hax, and Magnanti (Addison-Wesley, 1977)
   prepared by José Fernando Oliveira Maria Antónia Carravilla
2. Lecture slides from Prof Carolina Osorio (MIT 1.041)
3. Introduction to Linear Optimization (Bertsimas and Tsitsiklis) available in library
4. Formulating an MP: An Overview (Nathaniel Grier) available through Piazza
5. Introduction to Operations Research (Hillier and Lierberman)

Unit 4: The Big Picture

Units 1 & 2: Modeling transportation systems

Units 3 & 4: Optimizing transportation systems

- Unit 3: Making **sequential** decisions (dynamic, control)
  - Stochastic or deterministic
- Unit 4: Making a **single** decision (static, resource allocation)
  - Deterministic
  1. Modeling and solving linear programs (LPs)
  2. Modeling and solving integer programs (IPs)

Outline

1. Mathematical programming
2. Linear programming
3. Linearizing non-linear problems
4. Modeling caveats and standard form

Outline

1. **Mathematical programming**
2. Linear programming
3. Linearizing non-linear problems
4. Modeling caveats and standard form

Mathematical programming (Bradley et al., 1977, Chapter 1)

Mathematical programming, and especially linear programming, is one of the best developed and most used branches of management science.

It concerns the **optimum allocation of limited resources** among competing activities, **under a set of constraints** imposed by the nature of the problem being studied.

Linear program

$$\max_x c^T x$$
$$s.t.\ Ax = b$$
$$x \geq 0$$

where $x \in \mathbb{R}^n, c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$, and $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$

Mixed integer linear program

$$\begin{aligned} \max_{x,y} c^T x + d^T y \\ s.t.\ Ax = b \\ Cy = e \\ x, y \geq 0 \\ y \in \mathbb{Z}_+^{n_1} \end{aligned}$$

where A, b, c are as before,

and d $= \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n_1} \end{bmatrix}$, $C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n_1} \\ c_{21} & c_{22} & \cdots & c_{2n_1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{m_1 n_1} \end{bmatrix}$, and $e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{m_1} \end{bmatrix}$

Note that the constraints and objective remain linear.

General mathematical programs

A general (possibly nonlinear) mathematical program is of the form:

$$
\begin{aligned}
\max_{x} \ & f_0(x) \\
s.t. \ & f_1(x) \leq b_1 \\
& f_2(x) \leq b_2 \\
& \vdots \\
& f_m(x) \leq b_m \\
& x \in X
\end{aligned}
$$

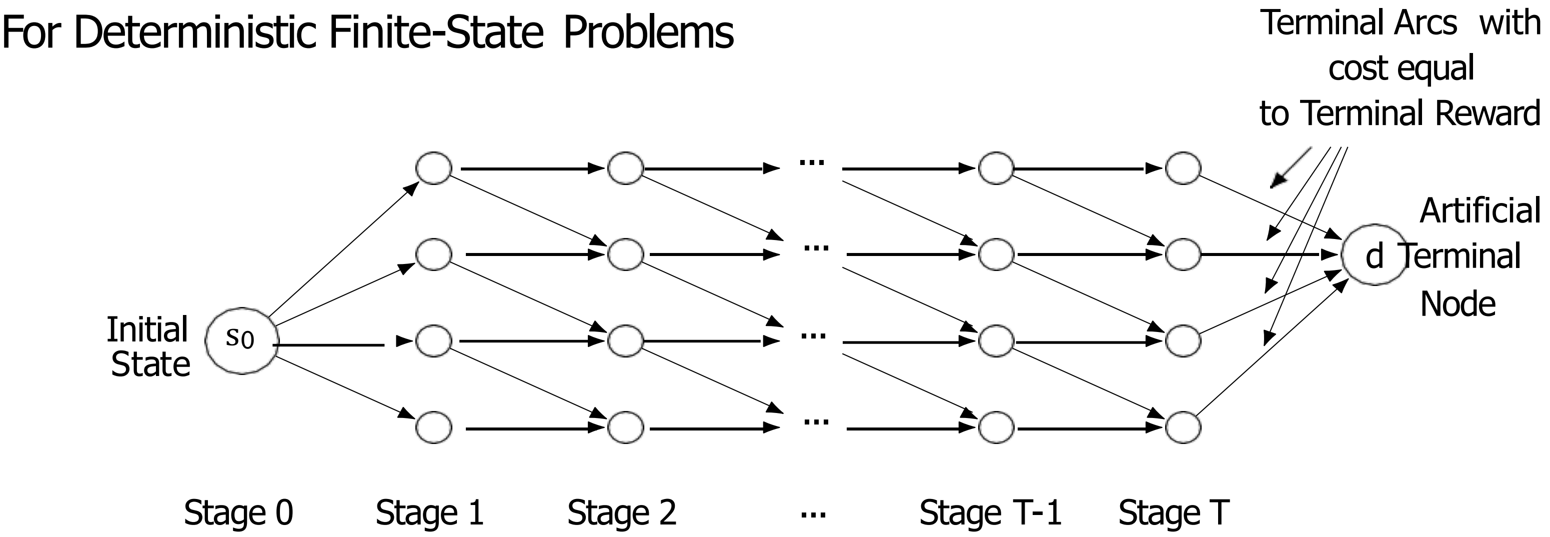
where X denotes a (possibly infinite) set of valid variable scopes.

Classifications of mathematical programs

Also called *optimization models* or *mathematical programming models*.

- According to the number of time periods considered in the model
  - **Static**: single time period (Unit 4)
  - **Sequential** or multi-stage: multiple time periods (Unit 3)
- Based on the type of variables in the optimal solution
  - **Continuous**: variables are allowed to take any value that satisfies the constraints (L19-21)
  - **Integer** or discrete: variables are allowed to take on only discrete values (L22)
  - **Mixed**: there are some integer variables and some continuous variables in the problem
- Based on the parameters of the model $(A, b, c, f_0, f_1, \text{etc})$
  - **Deterministic**: parameters are known and fixed constants (Unit 4)
  - **Stochastic**: parameters are specified as uncertain quantities, whose values are characterized by probability distributions
  - **Parametric**: some of the parameters are allowed to vary systematically, and the changes in the optimal solution corresponding to changes in those parameters are determined

Recall (Unit 3): Sequential decision making as shortest path

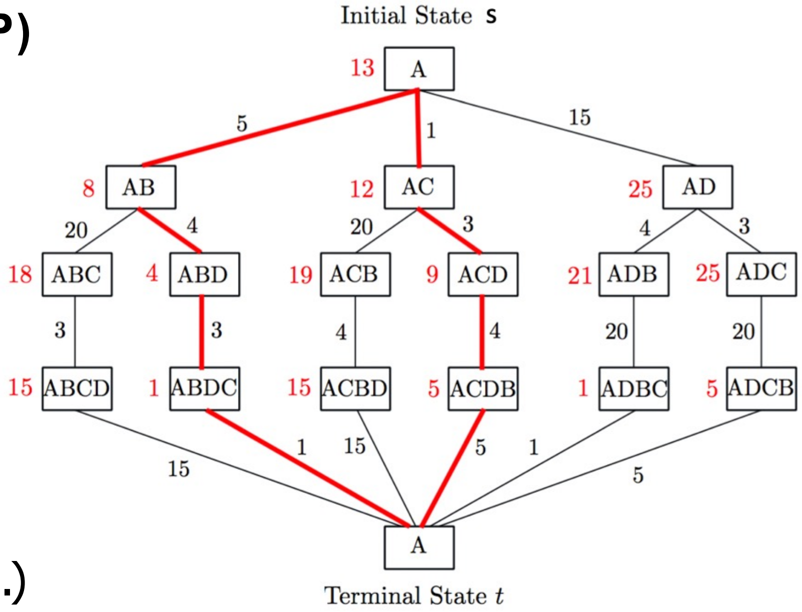


Example: Integer programming (combinatorial optimization) [We'll revisit in Unit 4.]

$$\begin{aligned} \max \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \in \{0,1\}^T \end{aligned}$$

Recall (Unit 3): Sequential decision making can get hairy

- **Example: traveling salesman problem (TSP)**
  - N cities.
  - **Goal**: Find the shortest tour (visit every city exactly once and return home).
  - In this case, can't get around exponential. (why?)
  - $|S| = O(N!)$, $|A| = N$, $T = N$, so
  - $O(|S||A|T) = O(N!)$.
  - (Actually, DP *is* slightly better: $|S| = O(2^N N^2)$.)
  - This is called the **curse of dimensionality**.



Matrix of Intercity Travel Costs

	5	1	15
5		20	4
1	20		3
15	4	3	

Example integer linear program

$$
\begin{aligned}
\max_{x,y} z &= 5x_1 + 8x_2 \\
s.t.\ x_1 + x_2 &\le 6 \\
5x_1 + 9x_2 &\le 45 \\
x_1, x_2 &\ge 0 \\
x_1, x_2 &\in \mathbb{N}
\end{aligned}
$$

- This problem can be formulated as a N=2 stage sequential decision problem.
  - Decision #1 is to choose a value for x_1
  - Decision #2 is to choose a value for x_2
- Then, the principle of optimality and dynamic programming applies. If we already had sub-solutions for each valid choice of x_2 then selecting an optimal x_1 would be easy.
- A very helpful element is that each valid choice of x_2 constrains the valid choices of x_1

- If shortest path isn't hard, why are these problems hard?
  - The number of edges may be exponential!
  - The number of edges at each step could grow by a factor of $|\mathcal{A}_k|$
  - So the number of edges overall could be $\approx \prod_{k=0}^{N-1} |\mathcal{A}_k|$
- Example
  - Take $|\mathcal{A}_k| = 2$ and $N = 275$ steps.
  - This means $2^{275} \approx 6\times 10^{82}$ edges overall, which is more than the estimated atoms in the universe. This is too big of a graph to do dynamic programming!

Linear programs

For linear programs, there are much more efficient methods than dynamic programming!

This is the topic of L20 & L21.

We'll revisit integer programs in L22.

Outline

1. Mathematical programming
2. **Linear programming**
   a. Transit ridership problem
3. Linearizing non-linear problems
4. Modeling caveats and standard form

Main components of an optimization problem

1. Objective function
   - summarizes the objective of the problem (max, min)
2. Constraints
   - limitations placed on the problem; control allowable solutions
   - problem statement: ‘given…’, ‘must ensure…’, ‘subject to’
   - equations or inequalities or variable value types
3. Decision variables
   - quantities, decisions to be determined
   - multiple types (real numbers, non-negative, integer, binary)

An optimization problem with linear objective function AND linear constraints is called a **linear program** / a linear optimization problem

Linear Programming model

- There are n quantifiable decisions to be made. Choose their values such as to min/max an objective. $(x_1, x_2, \dots, x_n)$ decision variables

$$
\begin{aligned}
&\min c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
&s.t. \\
&a_{11}x_1 + a_{12}x_2 + \ \dots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ \dots + a_{2n}x_n = b_2 \\
&\dots \\
&a_{m1}x_1 + a_{m2}x_2 + \ \dots a_{mn}x_n = b_m \\
&x_i \geq 0
\end{aligned}
$$

- x_j: level of activity j (for $j = 1, 2, \dots, n$)
- c_j: increase in objective function that would result from each unit increase in the level of the activity j
- b_i: amount of resource i that is available for allocation to activities (m resources, $i = 1, 2, \dots, m$)
- a_{ij}: amount of resource i consumed by each unit of activity j
- Model parameters: c_j, b_i, a_{ij}. They are input constants for the model.

Example: transit ridership

- A transit agency is performing a review of its services. It has decided to measure its overall effectiveness in terms of the total number of riders it serves. The agency operates a number of modes of transport. The table shows the average number of riders generated by each trip (by mode) and the cost of each trip (by mode).

Mode	Heavy rail	Light rail	BRT	Bus
Avg. ridership per trip (r_i)	400	125	60	40
Avg. cost ($) per trip ($c_i$)	200	80	40	30

- Give a formulation of the problem to maximize the total average number of riders the agency services given a fixed daily budget of $5,000.

Outline

1. Mathematical programming
2. Linear programming
3. **Linearizing non-linear problems**
   a. Ride-hailing problem
4. Modeling caveats and standard form

Non-linearity

- In a Linear Program (LP),
  - objective function AND constraints MUST BE linear
  - variable type must be continuous
  - e.g. real numbers, non-negatives are OK
  - integers, binary NOT OK
- $\max\{x_1, x_2, \ldots\}$, $x_i y_i$, $|x_i|$, etc $\Rightarrow$ non-linear (if x_i and y_i are both variables)
  - sometimes there is a way to convert these types of constraints into linear constraints by adding some decision variables

Non-linear programming

$$\text{Maximize } z = 60x_1 - 5x_1^2 + 80x_2 - 4x_2^2$$

subject to:

$$\begin{aligned} 6x_1 + 5x_2 &\leq 60 \\ 10x_1 + 20x_2 &\leq 150 \\ 8 \geq |x_1| &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

Dealing with absolute values

Goal: We want to formulate the following as a linear program

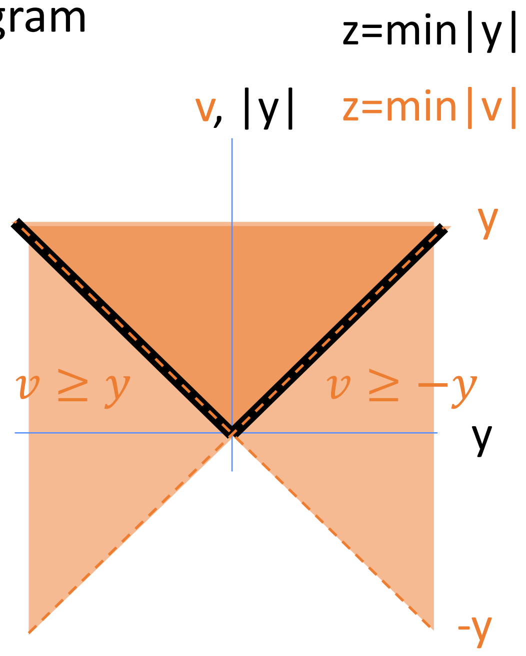
$$\min 5x + 2|y|$$
$$s.t.$$
$$x + y \geq 9$$

- Option 1

$$|y| = \max\{y, -y\}$$

- Option 2
  - Add a new variable v
  - Replace by $v \geq y$ and $v \geq -y$
- Formulation 1:

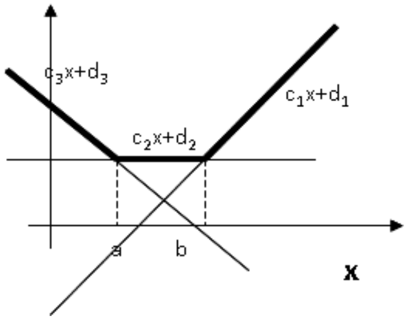
$$\min 5x + 2v$$
$$s.t.$$
$$v \geq y$$
$$v \geq -y$$
$$x + y \geq 9$$



Minimizing piecewise linear convex cost functions

Example

Costs defined as: $\begin{cases} c(x) = c_3 x + d_3, \ \forall x \in [-\infty, a] \\ c(x) = c_2 x + d_2, \ \forall x \in [a, b] \\ c(x) = c_1 x + d_1, \ \forall x \in [b, +\infty] \end{cases}$



What to do?

Introduce a new variable t with objective min t such that:

$$t = \max\{c_3 x + d_3, c_2 x + d_2, c_1 x + d_1\}$$

In linear form: $\begin{cases} t \geq c_3 x + d_3 \\ t \geq c_2 x + d_2 \\ t \geq c_1 x + d_1 \end{cases}$

Exercise: Construct an argument for the validity of this formulation

Ride Hailing Problem

- A new startup ride hailing company DropMe, has a budget of $150,000 to expand its taxi fleet.
- In order to increase their revenue, the firm is considering expanding its taxi fleet so that it can cover more suburban areas.
- Diminishing marginal returns: However, the larger the fleet, the less effective the reach of new passengers.
- DropMe is considering of adding two types of vehicles to its fleet (type A and type B) to combat the diminishing marginal returns by maintaining a certain "cool factor."
- Each addition of a type A vehicle costs $1,000 and each addition of type B vehicle costs $10,000.
- At most 30 more type A vehicles and at most 15 more type B vehicles can be added to DropMe fleet due to state law restrictions.
- The objective is to maximize the number of new passengers reached through this new fleet.
- Formulate this problem as a linear program.

	Nb. of additions	Nb. of new passengers reached
Type A	1 – 10	900
	11 – 20	600
	21 – 30	300
Type B	1 – 5	10,000
	6 – 10	5,000
	11 – 15	2,000

Outline

1. Mathematical programming
2. Linear programming
3. Linearizing non-linear problems
4. **Modeling caveats and standard form**

Caveats on mathematical programming model

It’s a model!

- It’s an abstract idealization of the real problem
- Simplifying assumptions and approximations are often needed to obtain a tractable model (i.e. model that can be solved in a reasonable amount of time)
- Models provide insights rather than accurate quantitative forecasts

Caveats on mathematical programming model

"Good model"

- Predicts the relative effects of alternative decisions with sufficient accuracy to permit a sound decision
- There is high correlation between the model predictions and what would actually happen in the real world, as opposed to high accuracy in the model forecasts

Caveats on mathematical programming model

“Optimal solutions”

- Are optimal with regards to the underlying model, which is a simplification of reality
- There is no guarantee that this solution will prove to be the best possible solution when implemented, because there are just too many uncertainties associated to the real problem
- A well formulated and validated (tested) model should lead to solutions that are a good approximation to an ideal course of action for the real problem

$$
\begin{aligned}
&\min c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
&s.t. \\
&\quad a_{11}x_1 + a_{12}x_2 + \ \ldots + a_{1n}x_n = b_1 \\
&\quad a_{21}x_1 + a_{22}x_2 + \ \ldots + a_{2n}x_n = b_2 \\
&\quad \ldots \\
&\quad a_{m1}x_1 + a_{m2}x_2 + \ \ldots a_{mn}x_n = b_m \\
&\quad x_i \geq 0
\end{aligned}
$$

Assumptions of the Linear Programming model

Main assumptions:

- Proportionality
- Additivity
- Divisibility
- Certainty

Does a linear programming formulation provide a satisfactory representation of a problem?

Converting constraints to standard form

Standard form

$$\min c^T x$$
$$s.t.\ \ A\,x = \ b$$
$$x \geq \ 0$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & \ddots & & \\ a_{m1} & & \cdots & a_{mn} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Non-standard form

- What about $a_1 x_1 \ + \ a_2 x_2 \geq \ b_1$?
- Replace with:
  - $x_3 \ := a_1 x_1 \ + \ a_2 x_2 - \ b_1$ (constraint)
  - Add constraint: $x_3 \geq \ 0$

$\overbrace{a_{11}x_1 + a_{12}x_2}^{+} \leq b_1$?

replace

$a_{11}x_1 + a_{12}x_2 + x_3 = b_1$

$x_3 \geq 0$ slack variable.

- What about $x_1 \leq \ 0$?
- Replace with:
  - Define $x_2 \ := \ -x_1$ (constraint)
  - Add constraint $x_2 \geq 0$.

- Why is this important?
  - Solvers (software) often require LPs to be formulated in standard form.

Additional exercises

Example: emergency response

- An emergency response center is to be located in a region with 4 major communities. The communities are located at

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$

  1. Choose the optimal location of emergency response center to minimize the **sum of distances** from each community
  2. Choose the optimal location of emergency response center to minimize the **maximum distance** from each community
- Formulate the above as linear programs.
- These are also known as **facility location problems**.
- Use the Manhattan distance
  - Consider 2 points:

$$\mathrm{p}_1 = (x_1, y_1), \mathrm{p}_2 = (x_2, y_2)$$

  - Manhattan distance metrics

$$\mathrm{d}(\mathrm{p}_1, \mathrm{p}_2) = |x_1 - x_2| + |y_1 - y_2|$$

    Makes sense in cases of grid-like road systems (like Manhattan)