Mathematical Optimization for Decision Making

Yuanzhe(Roger) Li

12/10/2019

Outline

- Overview of Mathematical Programming
- Mixed-Integer Programming (MIP) Models
- Examples of MIP Applications
  - Traveling salesman problem
  - Warehouse allocation problem
- The Role of Mathematical Optimization in Data Science

Mathematical Programming

- *Mathematical programming* is the selection of a best element (with regard to some criterion) from some set of available alternatives.
- A mathematical programming problem consists of
  - a set of ***variables*** that describe the state of the modeled system,
  - a set of ***constraints*** that define the region in which the states are allowed,
  - an ***objective function*** that provides an assessment of the system for any given states, and
  - external data that serve as input ***parameters***.

Formulating a Mathematical Programming Problem

- The general form of a mathematical programming model with n variables and m constraints is

$$\begin{aligned} \max \text{ or } \min \quad & f(x_1, ..., x_n) \\ \text{s.t.} \quad & g_i(x_1, ..., x_n) \leq 0, \quad i = 1, ..., m \\ & (x_1, ..., x_n) \in \mathbf{S} \end{aligned}$$

- The inequality constraints can cover $\leq$ and $\geq$, as well as equailty cases.
- $\mathbf{S}$ may be either continuous (e.g., $\mathbb{R}^n$) or discrete (e.g., $\mathbb{Z}^n$).

Nomenclature for Mathematical Programming

- A ***solution*** is an assignment of values of variables
- A ***feasible region*** is the set of all possible solutions that satisfy all the constraints. $\mathcal{F} = \{\mathbf{x} \in \mathbf{S} | g_i(\mathbf{x}) \leq 0, i = 1, ..., m\}$
- The ***objective value*** of a solution is obtained by evaluating $f(\cdot)$ at the given solution.
- An ***optimal solution*** (for a minimization problem) is one whose corresponding objective value is less than or equal to that of all other feasible solutions.

Categorizing Mathematical Programming Problems

- The type of a mathematical programming problem depends on the forms of the objective (f) and the constraints (g_i's), as well as the set S.
  - *Unconstrained* ($m = 0$ and $\mathbf{S} = \mathbb{R}^n$) vs. *Constrained*
  - *Linear* (f and g_i's are linear and $\mathbf{S} = \mathbb{R}^n$) vs. *Nonlinear*
  - *Convex* (f is a convex function and the feasible region $\mathcal{F}$ is a convex set) vs. *Nonconvex*
  - *Continuous* (e.g., $\mathbf{x} \in \mathbb{R}^n$) vs. *Discrete* (e.g., $\mathbf{x} \in \mathbb{Z}^n$)
- Knowing the type of the problem helps us understand how difficult it will be to solve it, and choose a suitable mathematical programming solver.

Mixed-Integer Programming (MIP) Models

- The general form of a MIP is

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, i = 1, ..., m \\ & \mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned}$$

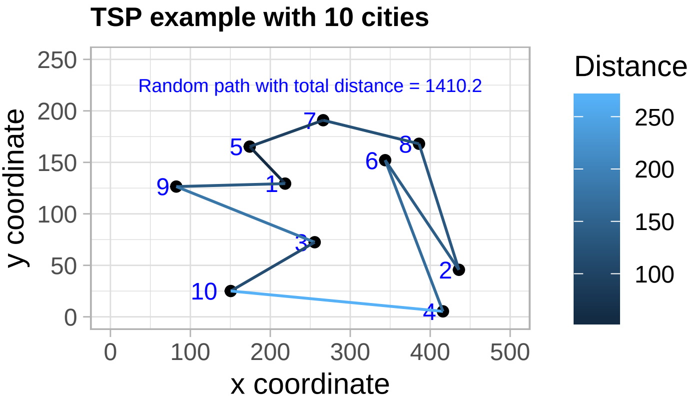
- Why do we need integer variables?
  - model indivisible physical entities (e.g., shares of stocks)
  - *binary (0-1) variables* can be used to model logical conditions or combinatorial structures.

Example 1 - traveling salesman problem (TSP)

- Problem definition (from wikipedia)

The travelling salesman problem (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

Example of a random path



- Finding the optimal path by brute force takes $O(n!)$

Formulate TSP as a MIP

Miller-Tucker-Zemlin formulation

- Label the cities with $1, ..., n$ and define:

$$x_{ij} = \begin{cases} 1 & \text{if a path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

- Use another variable u_i to denote the ordering of the cities.

Formulate TSP as a MIP (continued)

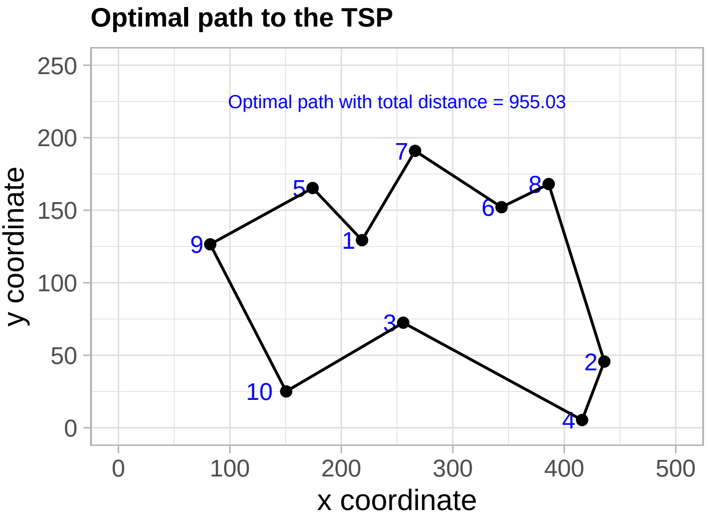
$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} d_{ij} x_{ij} & \\
s.t. \quad & x_{ij} \in \{0,1\} & i,j = 1,...,n \\
& u_i \in \mathbf{Z} & i = 2,...,n \\
& \sum_{i=1, i \neq j}^{n} x_{ij} = 1 & j = 1,...,n \\
& \sum_{j=1, j \neq i}^{n} x_{ij} = 1 & i = 1,...,n \\
& u_i - u_j + n x_{ij} \leq n-1 & 2 \leq i \neq j \leq n \\
& 0 \leq u_i \leq n-1 & 2 \leq i \leq n
\end{aligned}
$$

MIP solution to TSP

- We use the GLPK solver to optimize the MIP formulated above.

trip_id	from (i)	to (j)	distance
1	1	5	57.053
2	5	9	99.740
3	9	10	122.182
4	10	3	115.221
5	3	4	173.915
6	4	2	44.997
7	2	8	132.161
8	8	6	45.231
9	6	7	86.718
10	7	1	77.813

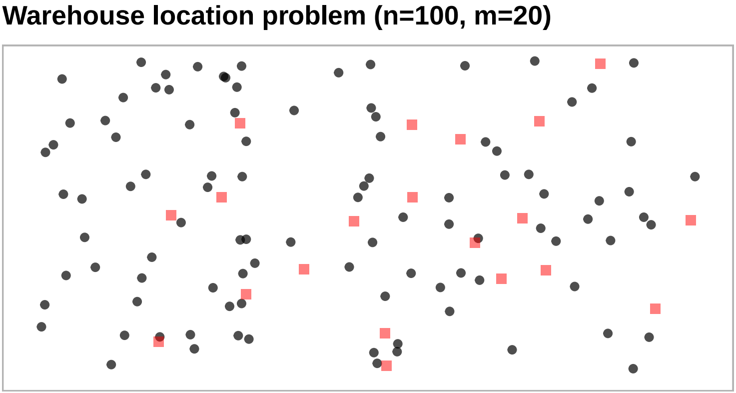
Visualize the optimal path of the TSP



Example 2 - facility location problem (FLP)

- Wikipedia definition: *A branch of operations research concerned with the optimal placement of facilities to minimize transportation cost while considering other factors.*
- We consider a simple warehouse allocation problem, where facilities can be built in m potential locations to serve n customers. The goal is to minimize the total transportation cost and fixed facility cost.

Visualization of a FLP



Black dots represent customers. Red squares show potential locations to build warehouses.

Formulate FLP as MIP

Warehouse allocation problem

- Let $I = \{1, ..., n\}$ be the set of customers, $J = \{1, ..., m\}$ the set of potential locations.
- Define two set of binary variables:

$$y_j = \begin{cases} 1 & \text{a warehouse is built at location } j \\ 0 & \text{otherwise} \end{cases}$$

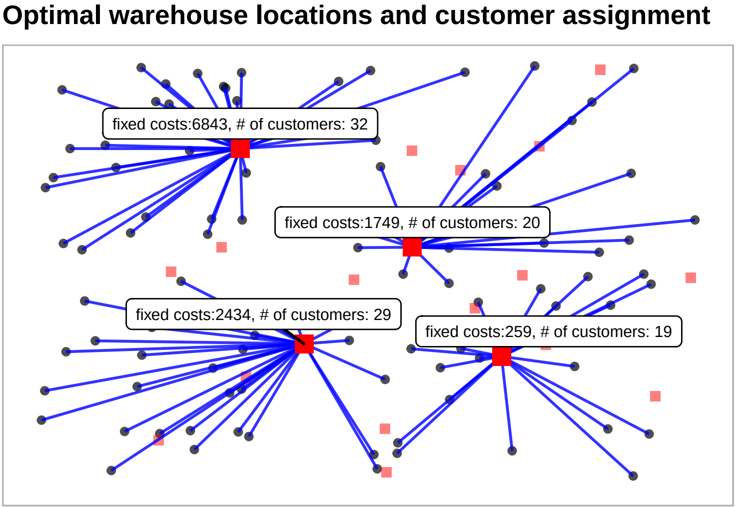
and

$$x_{ij} = \begin{cases} 1 & \text{if customer } i \text{ will be served by warehouse at } j \\ 0 & \text{otherwise} \end{cases}$$

Formulate FLP as a MIP

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} \mathsf{transportation\_cost}_{ij} \cdot x_{ij} + \sum_{j=1}^{m} \mathsf{fixed\_cost}_{j} \cdot y_j \\
s.t. \quad & \sum_{j}^{m} x_{ij} = 1, \quad \forall i = 1, ..., n \\
& x_{ij} \leq y_j, \qquad \forall i = 1, ..., n \quad j = 1, ..., m \\
& x_{ij} \in \{0, 1\}, \quad \forall i = 1, ..., n \quad j = 1, ..., m \\
& y_j \in \{0, 1\}, \quad \forall j = 1, ..., m
\end{aligned}
$$

MIP solution to the FLP



Mathematical Optimization and Machine Learning

- Use machine learning and mathematical optimization in conjunction.
  - E.g., use machine learning model predictions as input parameters to formulate the final decision making problem as a mathematical program.
- Use mathematical optimization to directly solve machine learning models.

References

- Ted Ralphs 2015. *ISE 347/447: Financial Optimization (Fall 2015)*, lecture notes, Lehigh University.
- Dirk Schumacher 2018. *ompr: Model and Solve Mixed Integer Linear Programs*, R package version 0.8.0.
- To get serious about mathematical optimization, checkout Boyd and Vandenberghe (2004)