Decision Analytics for Business and Policy

Homework 4: Integer Optimization

Due Date: Oct 4, 2021 11:59am ET	**Submission:** Canvas

Traveling Salesman Problem.

Traveling salesman problem (TSP): "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?" The minimum distance does not depend on the origin you select, since the route is a loop. People sometimes call the loop a *tour*.

TSP is notoriously time-consuming to solve. In general, n-city TSP has $n! = n \times (n-1) \times (n-2) \times \cdots \times 1$ possible tours. For example, to visit all the state capitals of continental US, there are about 10^{60} potential tours. If we ask you to explicitly describe an efficient algorithmic procedure to search through these potential tours and find the shortest one, it would be very difficult.

Alternatively, one could formulate this problem as an integer programming problem compactly, and not worry about the design of an explicit algorithm. Our job as a modeler is then to describe the problem precisely as an optimization problem and give it to a computer:

$$
\begin{aligned}
\min_{x_{ij}} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} x_{ij} \\
\text{s.t.} \quad & x_{ij} = x_{ji} \quad \text{for all } i = 1, \ldots, n \text{ and } j = 1, \ldots, n \\
& x_{ii} = 0 \quad \text{for all } i = 1, \ldots, n \\
& \sum_{j=1}^{n} x_{ij} = 2 \quad \text{for all } i = 1, \ldots, n \\
& \sum_{i \in S} \sum_{j \in S} x_{ij} \leq 2(|S| - 1) \quad \text{for every non-empty subset of cities } S \subset \{1, 2, \ldots, n\} \\
& x_{ij} \in \{0, 1\} \quad \text{for all } i = 1, \ldots, n \text{ and } j = 1, \ldots, n
\end{aligned}
$$

(a) [80%] You are given the integer programming formulation above that solves the traveling salesman problem. The set $\{1, \ldots, n\}$ has $2^n - 2$ non-empty, proper subsets. The number of elements in any subset S is denoted $|S|$.

Input data include:

- There are n cities in total.
- Distance matrix D: D_{ij} is the distance between city i and city j, for $i, j = 1, \ldots, n$.

Decision variables are x_{ij}, for $i = 1, \ldots, n$ and $j = 1, \ldots, n$, denoting whether city i and j are connected along the tour.

Answer the following questions:

1. Explain the objective function and each constraint in words.

2. The formulation looks reasonably easy to write down. Does that mean it is easy for a computer to solve? How many constraints are there in total? Express that number as a function of n. For example, "$x_{ij} = x_{ji}$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, n$" alone leads to n^2 constraints.

(b) [20 %] Your second task is to implement this model in Python and Gurobi, and solve a small TSP instance, with $n = 5$. The distance matrix is given in the file `five_d.csv`. The element in the ith row and jth column is the distance between location i and location j, *i.e.*, D_{ij}. A code template, `HW4_template.ipynb`, is given, and it already includes the implementation of the constraints involving subsets S.

Report the objective value (shortest tour length) and the optimal solution (sequence of cities to visit).