DS745: Project Three

John Woodward

2023-11-08

**Shortest Network Path**

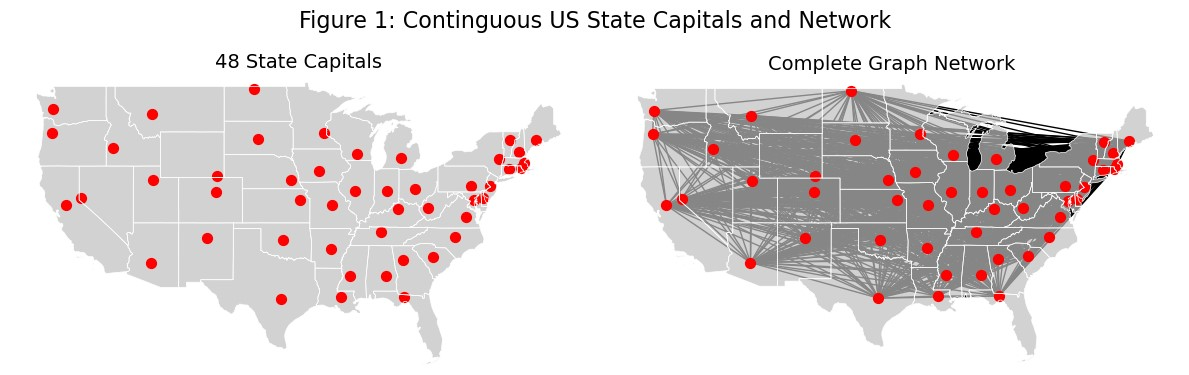
Welcome! We set off this network analysis project to find the shortest path between all US state capitols, visiting all capitols once and with one tour. This network optimization problem is crucial for many industry applications:

* Manufacturing sequencing (e.g., steel, microelectronics)
* Vehicle routing logistics (e.g., school bus, garbage truck)
* Automated medical imaging positioning (e.g., MRI, XRAY)

As we explore networks, we will apply linear programming (LP), a robust algorithm. LP is a mathematical method for optimizing an objective while following linear constraints. It formulates problems by linearly adding coefficients and decision variables to achieve a set goal, minimizing or maximizing the objective.

## 48-State Traveling Salesperson Problem (TSP)

To begin this TSP, I collected the longitude and latitude data (Moler, 2018) for the capital cities of the contiguous 48 states to conduct a network analysis (excluding HI and AK). Each city represents a node, and the edges between represent the distances as a helicopter would fly. Although it is possible to model driving durations in a car and other factors, I kept it simple to resemble an approximate solution and simplify the process.

In Figure 1, contiguous US states are depicted with red nodes representing state capitals. With 48 nodes connected, the right graph makes state colors barely visible. The number of edges (𝐸) in a complete graph with 𝑁 nodes is:

The number of tour possibilities (𝑇) for a Traveling Salesman Problem with 𝑁 cities is:

If one were to attempt a brute force solution, one would need to explore unique tours before proving optimality; that is many tours! This is where LP can help.

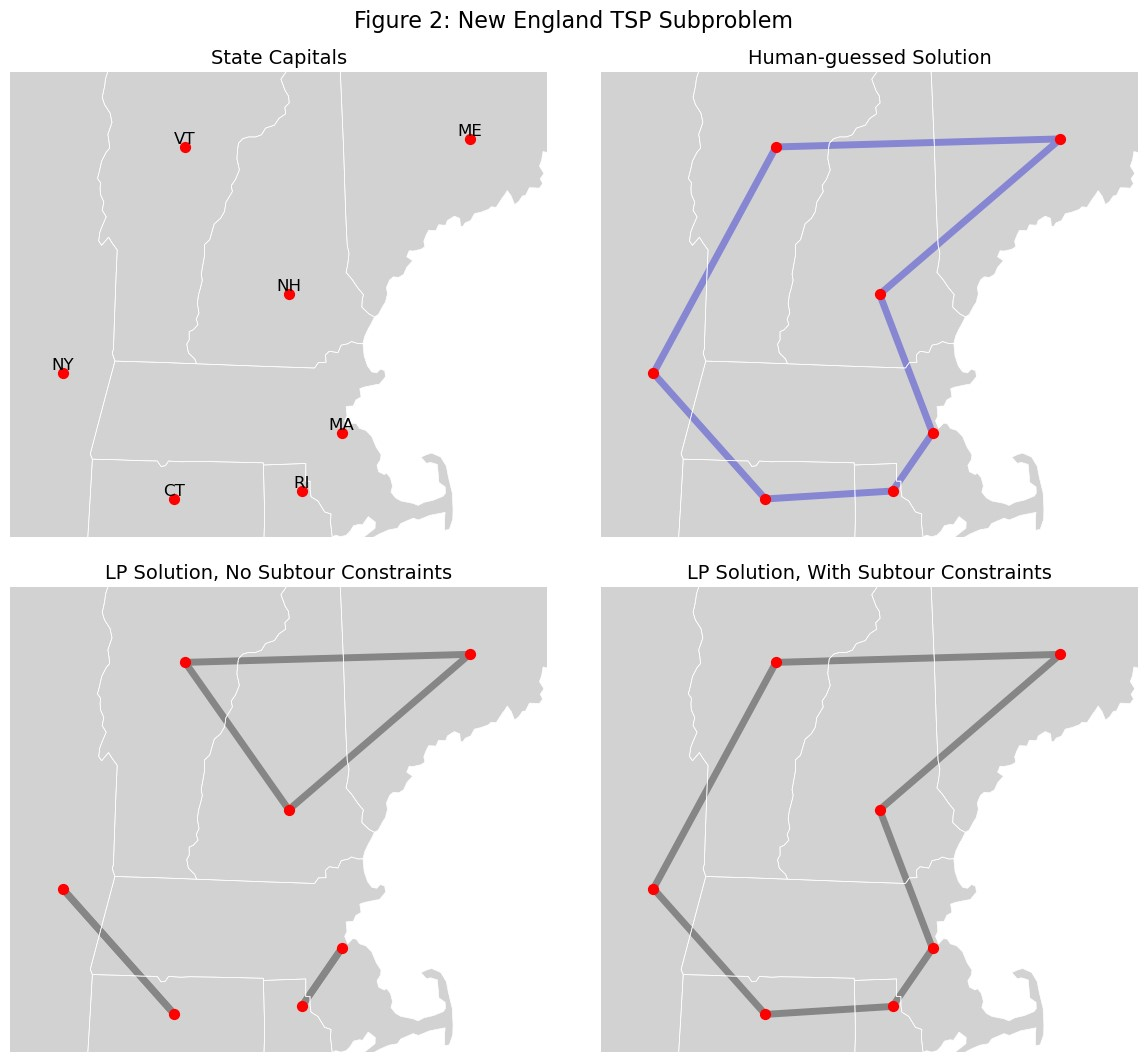
## Linear Programming (LP) Modeling

LP is effective where brute-force algorithms take far too long and one seeks an optimal solution. (If suboptimal was OK, other heuristic algorithms might suffice, too, like the greedy or evolutionary algorithms). First, we need to start with the goal we are optimizing with our model.

The primary metric we are concerned with is the total distance in miles traveled. We also wish to minimize the total distance, subject to previously mentioned constraints. A popular LP modeling method for TSP (Travelling salesman problem, 2023; Woche, 2019) is the following:

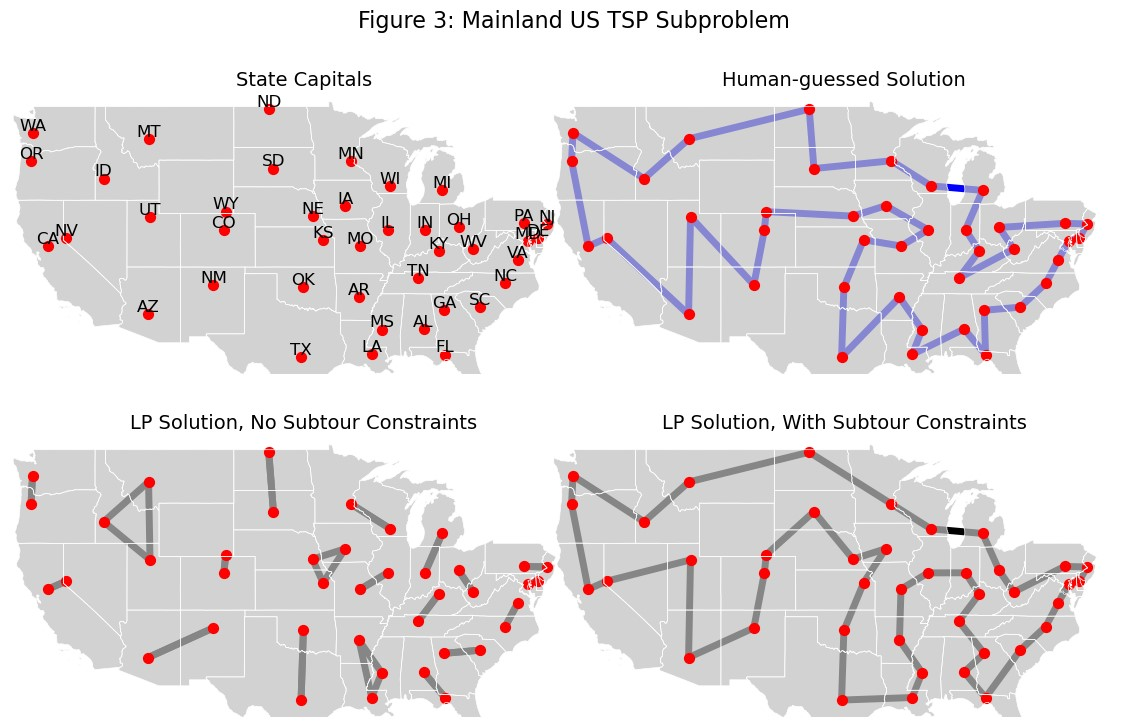
* The total distance is the network matrix multiplied by the distance matrix of the city-to-city miles (e.g., MA to CA, CA to MA).
* The decided *x* tour variables are visited or not (i.e., one or zero). For instance, traveling from MA to CA would be a zero because of the long distance, whereas NH-ME is more likely to be one (due to proximity).
* The subtour constraints are complex but ensure a single tour rather than multiple.

It is important to note that the problem's size can also affect LP. Moreover, when using LP to solve problems with integer solutions, the time required to solve the problem increases significantly. However, it is still faster than brute force. To strategically solve this TSP, Figure 2 shows a method that George Danzig (George Dantzig's contributions to integer programming, 2007) also used to simplify the problem to the 48-state problem in the 1950s:

Figure 2 illustrates a subset of the original 48-state traveling salesperson problem, focusing solely on New England. This simplifies the problem temporarily, making it more manageable with just seven states. As shown in the top right, I estimated the optimal solution, but can we confirm optimality?

It turns out LP problems can both be modeled with a "primal," say, minimizing, and a "dual," maximizing (Dual linear program, 2023). They are two sides of the same coin but offer two directions to meet in the middle. While the inverse of the original model can sometimes be more challenging to interpret, it acts as an upper and lower bound, respectively. Thus, when a primal and dual have the same objective value, say 689 miles, **with no gap**, it is proven mathematically to be the optimal answer!

I opted for LP using the previously presented model in the bottom plots. I used my Windows OS, Pyomo, and an LP solver (e.g., CBC). The bottom left showcases the importance of subtour constraints, which only took 0.1 ms and produced a 583-mile "journey," but it's unfeasible. We can avoid impractical subtours with the proper constraints, making the journey longer but feasible; thus, constraints prove invaluable in eliminating non-viable solutions.

After applying these constraints, the LP approach yields an optimal 639-mile tour, matching my human estimate. However, the machine accomplished that in a mere 0.84 ms compared to my 10 seconds on such a small TSP problem. Let's move on to the remaining state capitol subproblem in Figure 3! Figure 3 tackles most of the original 48-state traveling salesperson problem, excluding New England:

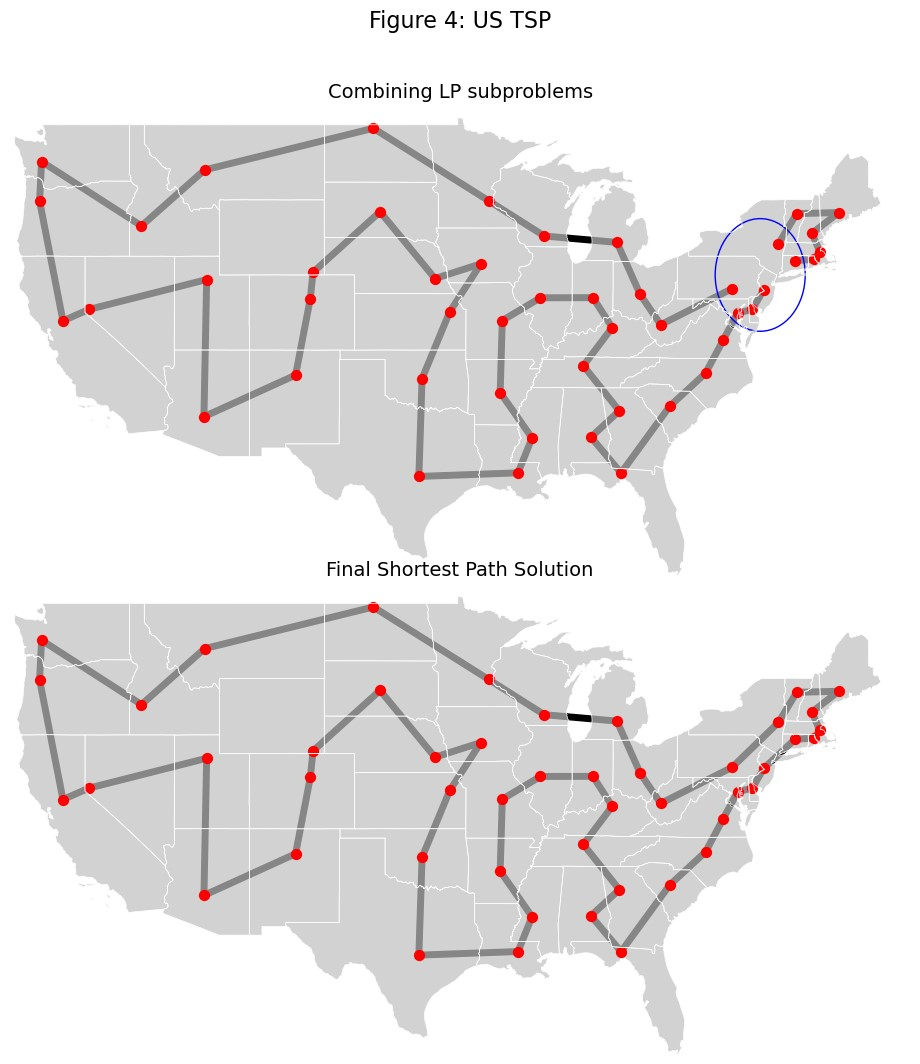
The top-right subplot reveals my challenges in planning a trip with a significantly larger problem size. I navigated through the map on my iPhone, methodically drawing a feasible tour totaling 10,519 miles. Was it optimal? If not, how far was I off the mark?

The bottom subplots display the LP solutions, one with subtour constraints and one without. These solutions offer intriguing insights into potential linkages that influence the final optimal route, depicted on the right. The left solution efficiently achieves the shortest "journey" of 8,117 miles in a mere 0.44 ms, but it isn't too helpful.

I used a series of iterative runs by combining computational and clever strategies. Starting with all constraints for 60 seconds, I saved the best feasible outcome. This suboptimal solution was then used as a "warm-start" for the LP, leading to incremental progress. An almost optimal solution emerged after four runs, each about five minutes long. Finally, an extended 2-hour run was performed to verify optimality - it could have been faster, as we'll discuss later. The solution, taking approximately 2.5 hours, confirmed the prior best feasible route as optimal at 9,996 miles, with a 5.2% deviation (523 miles) from the optimal solution.

Using the optimal route for the subproblem, let's integrate it with the other subproblem's optimal route. In solving the final TSP, I found it necessary to represent the problem visually. To do this, I mapped both subproblems on my phone and identified the closest nodes to each other in proximity.

Figure 4 (top subplot) illustrates the strategic disconnection of the two nearest edges. This step would allow them to reconnect (bottom subplot) later as part of the main problem's solution.

Once I had fed these two subtours into the LP solver as a pre-solved problem, the next step was to connect the missing segments in the tour, resulting in a fully connected route (the 10,821-mile route was combined in 0.86 milliseconds).

## Network Analysis Discussion

Solving large networks like the Traveling Salesman Problems (TSP) with Linear Programming (LP) presents various challenges. This discussion will explore these challenges, solutions, and the broader implications of network analysis and TSP.

Selecting a suitable LP solver is pivotal. GLPK is apt for more minor problems, but handling more significant TSP instances demands the prowess of robust solvers like CBC, CPLEX, or GUROBI. While CBC offers a cost-effective choice, it can be challenging to configure on Windows (How to install `coincbc` using Conda in Windows, n.d.). Furthermore, the expedited solving potential of multithreading is constrained by Windows and CBC compatibility issues. In contrast, Linux and Mac versions provide a multithreading advantage, reducing the resolution time for a 42-state TSP from hours to minutes.

The experience from this project was enlightening and enjoyable, providing skills for future applications. The methods developed in the code can assist students and professionals in network analysis and TSP problem-solving.

In conclusion, TSP problems are fascinating and pose real-world challenges, demanding mathematical modeling, algorithmic thinking, and efficient solver utilization. The choice of algorithm, multithreading, and operating system are crucial considerations. As displayed by TSP, network analysis is vital in optimizing routes, resource allocation, and decision-making in various domains, making it an invaluable field of study and application.

# References

*Dual linear program*. (2023, 08 05). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Dual\_linear\_program

*George Dantzig's contributions to integer programming*. (2007, 10 22). Retrieved from Discrete Optimization: https://www.sciencedirect.com/science/article/pii/S1572528607000412

*How to install `coincbc` using Conda in Windows*. (n.d.). Retrieved from Stack Overflow: https://stackoverflow.com/questions/58868054/how-to-install-coincbc-using-conda-in-windows

Moler, C. (2018, 09 17). *USA Traveling Salesman Tour*. Retrieved from MathWorks: https://blogs.mathworks.com/cleve/2018/09/17/usa-traveling-salesman-tour/

*Travelling salesman problem*. (2023, 11 8). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Travelling\_salesman\_problem

Woche, C. (2019, 12 31). *Model and solution of the traveling salesman problem with python and pyomo*. Retrieved from Medium: https://medium.com/analytics-vidhya/model-and-solution-of-the-traveling-salesman-problem-with-python-and-pyomo-db45f2631e8c