**Boston Coffee Shops and the Travelling Salesman**

**Introduction**

**TSP and Logistics**

The Travelling Salesman Problem (TSP) has worried logistics managers and route planners for quite a long time now. But what does the TSP mean? To cite Wikipedia and its respective site: "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?". In fact, it is a problem of operational research and its solution lies not too far from the traditional method of Simplex.

Moreover, the optimization of FTL (Full-Truckload) mileage is the key questions to be answered although, a set of questions may derive from it. For example, how can the FTL mileage be maximized? Assuming a fleet of two or more trucks is available, what is the optimized route for each truck?

But, for reasons of simplicity, this study will focus on minimizing the miles of a truck's full circle of their delivery points without taking the full truck load aspect into consideration.

**Boston**

Boston is a very vivid metropolis of the state of Massachusetts. Boston and its peripheral areas are home to almost 5 million people and spans at a great 232 km².

Coffee, as in many other cities of the USA, is loved and worshipped there. As a result, Bostonians have a large number of coffee shops to choose where they get their favorite brew from.

This poses a serious nerve-racking problem to logistics planners since the question derived from the above is: how can a route planner, optimally plan the delivery points of a driver so that the distance covered, to pass through its designated coffee shops, is the best possible.

**Data Used**

The data used for the answering of the above question is derived from the official site of the data hub of the city of Boston and is available here: <https://data.boston.gov/dataset/boston-neighborhoods/resource/07caa2dd-1499-491e-a1d2-7bcba414cdd7> . The csv includes all 26 of the Boston neighborhoods. The format of the csv can be seen below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| OBJECTID | Name | Acres | Neighborhood\_ID | SqMiles | ShapeSTArea | ShapeSTLength |
| 27 | Roslindale | 1605.568238 | 15 | 2.51 | 69938272.67 | 53563.9126 |
| 28 | Jamaica Plain | 2519.245394 | 11 | 3.94 | 109737890.4 | 56349.93716 |
| 29 | Mission Hill | 350.8535636 | 13 | 0.55 | 15283120.1 | 17918.72411 |
| 30 | Longwood | 188.6119467 | 28 | 0.29 | 8215903.537 | 11908.75715 |
| 31 | Bay Village | 26.53983916 | 33 | 0.04 | 1156070.771 | 4650.635493 |
| 32 | Leather District | 15.63990811 | 27 | 0.02 | 681271.6719 | 3237.140537 |

**Methodology**

To begin, we have the csv with the names of the Boston neighborhoods and some other features concerning their spatial area they cover. A strategic plan of how to proceed is deemed necessary for the project to be complete in time. Below, the steps followed and their brief explanation are listed.

1. Get Coordinates

The first step is to take advantage of the Foursquare API and get the latitude and longitude coordinates of each neighborhood. This task has already been covered during the courses and the coding remains the same. The resulting dataframe is depicted below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **OBJECTID** | **Name** | **Acres** | **Neighborhood\_ID** | **SqMiles** | **ShapeSTArea** | **ShapeSTLength** | **lat** | **long** |
| 27 | Roslindale | 1605.568 | 15 | 2.51 | 6.99E+07 | 53563.91 | 42.29121 | -71.1245 |
| 28 | Jamaica Plain | 2519.245 | 11 | 3.94 | 1.10E+08 | 56349.94 | 42.30982 | -71.1203 |
| 29 | Mission Hill | 350.8536 | 13 | 0.55 | 1.53E+07 | 17918.72 | 42.33256 | -71.1036 |
| 30 | Longwood | 188.6119 | 28 | 0.29 | 8.22E+06 | 11908.76 | 42.34153 | -71.1102 |

1. Data exploration

A thorough observation of data is necessary to identify possible anomalies in it. Neighborhoods whose coordinates we could not get, were dropped. For example, Harbor Islands, a selection of points in the map that corresponds to more than one points and thus their coordinates were not identified, was dropped.

1. Get Venues

Using the Foursquare API, every Boston venue at a radius of 5 around each neighborhood was fetched. Again, this process has been covered in the courses and its coding did not differ too much. The dataframe at this point looks like the one shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Neighborhood** | **Neighborhood Latitude** | **Neighborhood Longitude** | **Venue** | **Venue Latitude** | **Venue Longitude** | **Venue Category** |
| Roslindale | 42.29121 | -71.1245 | Peters Hill | 42.29362 | -71.12806 | Scenic Lookout |
| Roslindale | 42.29121 | -71.1245 | Roslindale House Of Pizza | 42.28799 | -71.12655 | Pizza Place |
| Roslindale | 42.29121 | -71.1245 | Guira Y Tambora | 42.29185 | -71.12225 | Cuban Restaurant |
| Roslindale | 42.29121 | -71.1245 | BCYF- Flaherty Pool | 42.28813 | -71.12291 | Pool |
| Roslindale | 42.29121 | -71.1245 | Target | 42.2882 | -71.12666 | Big Box Store |

1. Clean Venues

All the different venues may address all the different stores and shops of Boston.



It was observed that the venue needed for our study were the ones named ‘Coffee Shops’.

So, the ones that do not cater to our needs were thrown away and only the ones that refer to Coffee Shops were kept.

The resulting dataframe was one with 27 rows and 7 columns, rather small one could argue. But taking into consideration that a driver serves 20 points in average in a workday, 27 points seem if not reasonable, then maybe rather high.

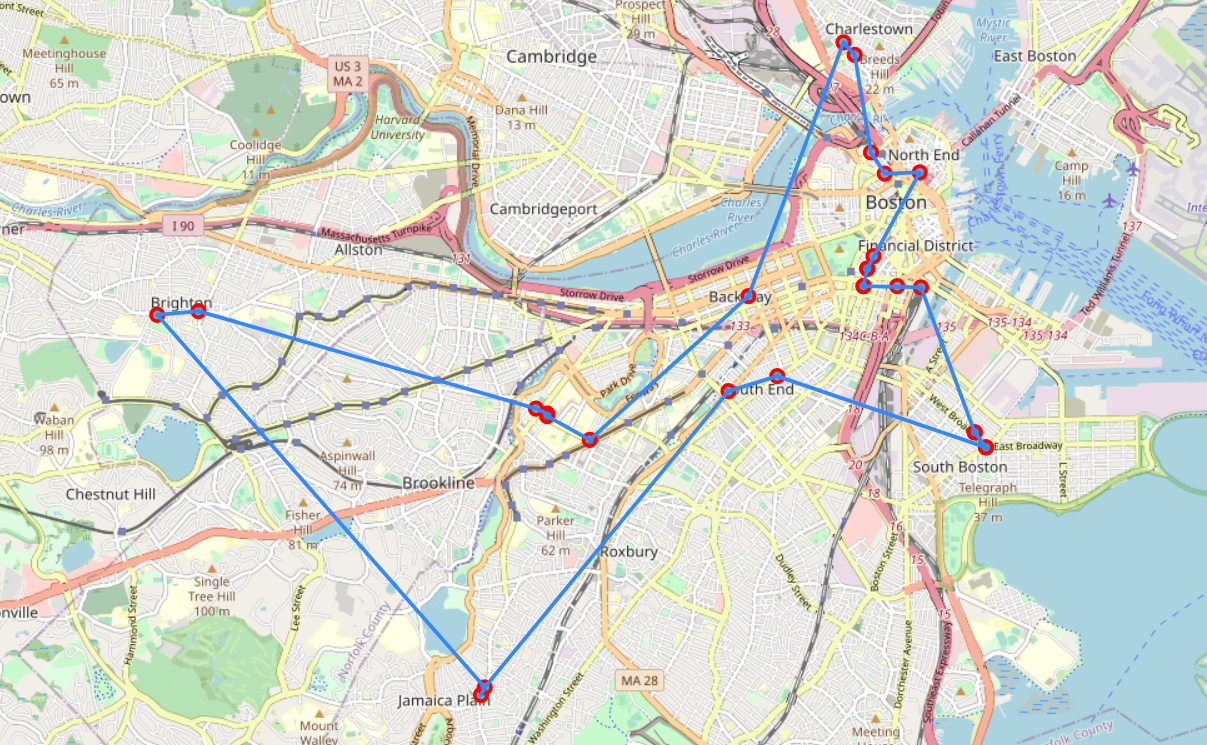
1. Route Optimization

Gurobi Optimization offers a great library of Optimization tools for the solution of the Travelling Salesman Problem. Prior to using the functions our dataframe must be split into a list and a dictionary. It is also worth mentioning that the coordinates must be concatenated in a single value. Later, the phases that result to the final optimized route are three:

* 1. Calculate the distance of one point from every other point.
  2. Define a function that eliminates sub-tours, a point being visited more than once.
  3. Find the shortest route.

**Results**

The resulting points and the route to be planned by the logistics planner and followed by the driver is shown below.



The coffee shops are not numbered, since the starting point, or rather, the warehouse was not defined. But, the coffee shop closer to the warehouse should be the first point visited, while the rest of the route remains the same.

**Discussion**

The Travelling Salesman Problem, as already mentioned, in its simplest form is not one of rather troublesome nature, when the tools are readily available to a professional. The most difficult parts of this research were the engineering of the problem, in terms of parameter tuning, data exploration, data cleaning. Further studies can focus on the addition of extra parameters to the problem. Fox example, that of a fleet of more than one trucks, or the maximization of mileage of FTL.

**Conclusion**

To conclude, the derived resulting map connects all the venues derived for the answering of the question and it is shown in the following parts of code and presentation. Lastly, combining the abilities of fetching venues and data through the use of a Foursquare API and bringing them to a form of analytical understanding through the use of the Gurobi Optimization library is a methodology of great and insightful results.