Refuel Optimizer

Project Team:

1. Samiran Kawtikwar
2. Nowsheen Sharmili
3. Vishal Daylan

Table of Contents

Problem Statement

This project is based on a real-life problem faced by many Americans while being on a long drive. Due to the nature of the US economy and state taxation rules, the prices of gasoline vastly vary across different regions. Refueling is often necessary while on a long journey and the variation in cost of fuel along the drive is often neglected. By making deliberate and informed decisions on when and where to refuel, one can save a significant amount in overall fuel costs without necessarily affecting the quality of the journey. Even though there has been a tremendous body of literature in this area, this problem interestingly doesn't have a commercial solution that could be used by an individual. In this project, we bring the "when and where to refuel" problem under the lens of optimization to provide a solution that gives the locations of gas stations to fill at and tells how much fuel should be filled up at those stations.\

Previous Work

Though there have been several works addressing the vehicle routing problems, the first scholarly work that considered the vehicle refueling problem was done by (Lim, 2005). Their problem considered the optimal facility location of fueling facilities for limited vehicle range. However, their problem did not assess "where-to-buy" vehicle refueling problems. Also, most other works used CEPLEX and ISA algorithm but were lacking the where-to-buy concept. This type of work can be found in (Lin, 2008), (S.H. Lin, 2007), and (S. Khuller, 2008). (S.H. Lin, 2007) considered the fixed-path vehicle refueling problem which is similar to that addressed by commercial fuel optimizers, and thereafter developed a linear-time greedy algorithm. This work was extended by (Lin, 2008) who developed an algorithm that jointly determines the optimal path from origin to destination for non-fixed path. (S. Khuller, 2008) formulated the problem for both fixed and nonfixed paths using polynomial-time algorithms and developed approximation algorithms to solve the TSP. The previous literature shows that the researchers did not consider this type of vehicle refueling problem that we presented in the report in spite of the actual software product proliferation in the field.

Our solution

In this section, we discuss our solution in detail that solve the optimum refueling problem. Being limited by data availability, compute resources, and time numerous assumptions and estimations have been made throughout this work; these are also described in this section. We have split this section into 3 parts.

Generating Transportation Network:

This project uses the osmnx package to generate the transportation network. Due to the limitation of resources, we stick to only the state of Illinois and consider this as the complete network. The nature of roads on the transportation network is restricted to only primary and secondary highways as otherwise, the graph adds too many streets which makes it unnecessarily huge and difficult to handle. The osmnx package returns the transportation network as a directed multigraph, we convert this to a directed graph for simplicity and being able to use many networkx functions.

Generating gas station locations:

Getting the data of all gas stations with prices turned out to be extremely difficult. The gas stations are roughly uniformly distributed along highways and since we only deal with primary and secondary highways in this network, we randomly select 5% nodes of the total nodes in the graph to have gas stations.

Figure 1 shows the transportation network generated through this process with gas stations.

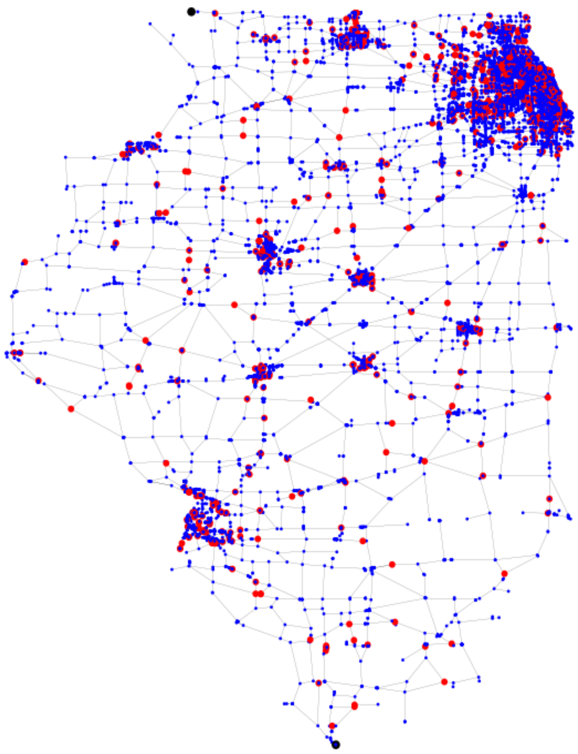

Figure 1 Illinois Transportation network with primary and secondary highways and gas stations (in red)

Estimating gas Prices :

The objective of the project depends on the Gas prices at the respective locations which are in the path traversed by the user. So, it is imperative to find the appropriate prices according to the location of the Gas Stations. For this we follow the following methodology. First, we import all the necessary libraries such as arcgis, pgeocode and geopy. For each node, we have the Latitude and Longitude data as coordinates. We use this data to find the zipcode of the node. Then using the geolocation data, we find the population and population density from the zipcode of the node. Then we compute the average population density for Illinois and the overall population density similarly. Then, we find the Fuel price of that Gas Station node by using the formula :

$$\text{Gas Price} = \text{Mu} + (\text{Ci} * \text{Sigma})$$

Where Mu = Mean of the Gas price

Sigma = Standard deviation of the Gas prices

$$\text{Ci} = \frac{(\text{Population Density of the location of the Gas node} - \text{Average Population Density})}{\text{Std-dev in pop-density}}$$

By using this , we find the Gas prices of each gas station according to their geolocation and hence, it would form the baseline of achieving the objective of this project.

Formulation

This section describes a simple formulation that we used to solve this problem, this formulation uses an already defined path and considers all the gas locations along this path and decides where to how much to fill up. This formulation is adapted from (Suzuki, 2008)

Let μ be the set of all the gas stops along the (shortest) route from origin o to destination d, and (i = 1, 2, . . ., n) be the elements of (see Fig. 2). The required inputs are as follows:

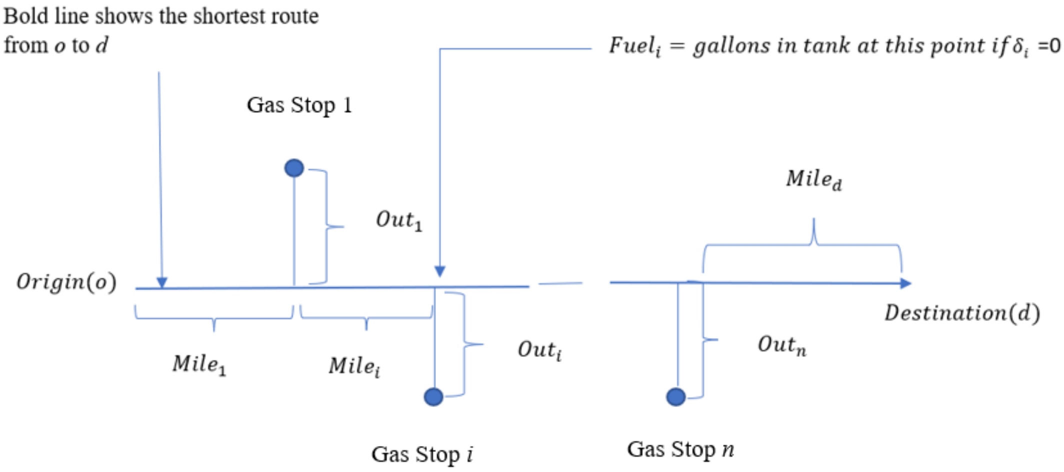

Figure 2 Sample Route

$Price_i$ = retail gas price (per gallon) at gas stop i

Out_i = number of miles that a vehicle must go out of route (OOR) to reach gas stop i

$Mile_i$ = distance (miles) from gas stop $i-1$ to i (not including Out_{i-1} or Out_i)

SF = amount of fuel (gallons) in tank at origin o (starting fuel)

MG = average fuel consumption rate (miles per gallon or MPG) for the whole trip

TC = vehicle tank capacity (e.g., 200 gallons)

LF = minimum fuel to be maintained in tank at all times (lower bound fuel)

MP = minimum amount of fuel to purchase at gas stops (e.g., 50 gallons)

EF= required amount of fuel in tank at the final destination d (ending fuel)

The formulation is provided below:

$$Minimize_{\varphi_i \geq 0, \delta_i} \left\{ \sum_{i \epsilon \mu} Price_i \emptyset_i \right\} \tag{1}$$

$$\delta_i \in \{0,1\} \; \forall i \in \mu \tag{2}$$

$$Fuel_i \geq LF \; \forall i \in \mu \tag{3}$$

$$Fuel_d \geq EF \tag{4}$$

$$\emptyset_i \geq \delta_i MP \; \forall i \in \mu \tag{5}$$

$$\emptyset_i \leq \delta_i TC \ \forall i \in \mu \qquad (6)$$

$$Fuel_i + \emptyset_i \leq TC \ \forall i \in \mu \qquad (7)$$

where:

δ_i = 1 if gas stop *i* is selected as a refueling point, 0 otherwise

$\emptyset_i$ = nonnegative amount of fuel (gallons) to purchase at gas stop *i*

$Fuel_i$= nonnegative amount of fuel in tank either at gas stop *i* before buying fuel (if $\delta_i = 1$) or at the point nearest to *i* along the route (if $\delta_i = 0$) (see Fig. 2)

$Fuel_d$ = remaining fuel at the destination *(d)*

We calculate $Fuel_i$ and $Fuel_d$ values by using the following formulas:

$$Fuel_i = SF - \frac{Mile_i + \delta_i Out_i}{MG}, \qquad if\ i = 1 \qquad (8)$$
$$Fuel_{i-1} + \emptyset_{i-1} - \frac{\delta_{i-1} Out_{i-1} + Mile_i + \delta_i Out_i}{MG}, if\ i \neq 1$$

$$Fuel_d = Fuel_n + \emptyset_n - \frac{\delta_n Out_n + Mile_d}{MG} \qquad (9)$$

where $Mile_d$ is the distance between gas stop *n* and the destination *d* (not including Out_n). Observe that the above model minimizes the cost of buying fuel (refueling cost) between *o* and *d* while ensuring that: (*i*) the remaining fuel in the tank does not fall below *LF* at any point in the route (constraint 3), (ii) the ending fuel is larger than or equal to *EF* (constraint 4), (iii) the minimum purchase quantity (of diesel fuel) is *MP* at any gas stop (constraints 5 and 6), and (iv) the sum of remaining fuel in the tank (before buying fuel) and the amount of purchased fuel does not exceed the tank capacity at any gas stop (constraint 7).

Generating Input Data

The model used in Formulation needs many entries as input data, since we have a raw directed graph with location and prices of gas stations, we use multiple graph theory techniques to find the required data. Here we describe each individually:

1. Finding Path -

   To find the path between source and destination, we use the shortest path function in the network with source and destination as (roughly) top left and bottom right point nodes in the graph (according to latitude and longitude).

   Since, the shortest path might go through an expensive area, there might exist a slightly longer path with cheaper fuel options that reduce the overall fuel cost, to solve this we find all edge-disjoint shortest paths from the source to the destination and run the optimization model on all those paths. In the given example, we could generate the 4 paths shown in Figure 3 to Figure 6.

2. Finding *Out* distances –

   At each node along the path, the model needs to know the location of the closest gas station. This could be achieved in many ways one could be by storing all pair shortest path distances but this has quadratic memory requirements and is not saleable.

   This problem is relatively easily solved by using a cutoff filter based on the GPS coordinates of the gas stations. We first filter out the gas stations that belong to a 10X10 km box around a node and then find the node out of them which is the closest. This saves on memory and computation in a big network as there could be millions of gas stations and nodes in the graph.

   The 10km cutoff can be further reduced to achieve faster performance.

3. Finding Mile –

   Once, the out distances are found, there might be some nodes that do not have any gas station in the 10 km vicinity, these nodes are considered dummy nodes for that particular path and are dissolved, the mile distance is then calculated by adding up the lengths of edges and converting it to miles.

   a. Rest Parameters:

      Parameters like starting fuel, ending fuel, miles per gallon, min fuel to fill up, and tank capacity are defined using realistic figures and industry standards for a typical mini SUV.

Results

We manage to solve the problem of traveling from East Dubuque, IL (Top left of IL) to the intersection of Ohio and Mississippi rivers (Bottom of IL). Our solution not only gives the optimum (shortest) path to travel this distance but also a detailed guide on where and how much to fill up to minimize total fuel cost.

Since, the shortest path and optimum refueling problems are solved separately the solution is suboptimal but with enough alternative shortest paths, we can make sure that a near-optimal solution could be achieved at a computational cost. Our solution provides no guarantees on the suboptimality bound for the dual cost but is indeed a practical solution to solve the "where and how much to fill" problem!

For the problem instance tested, we generate 4 alternative paths and find the fuel cost for each path. These paths are shown in Figure 3 to Figure 6

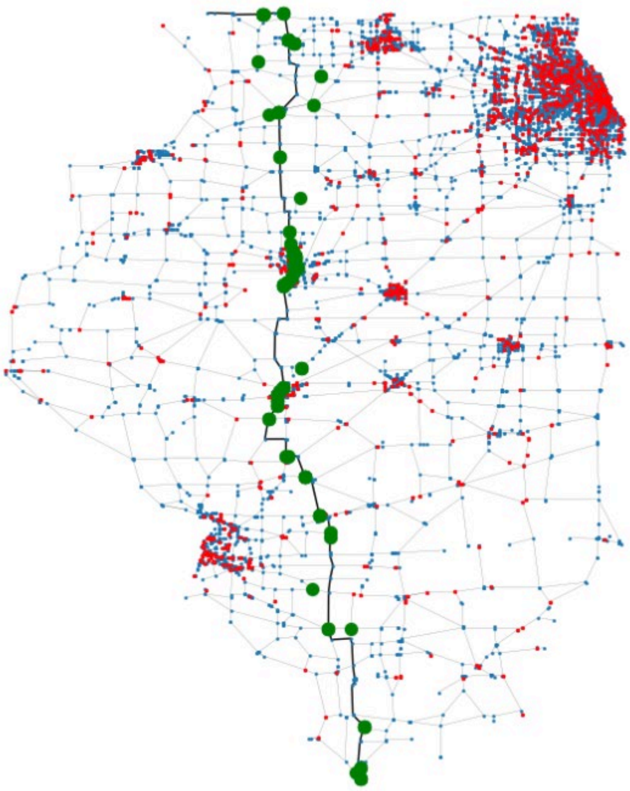

Figure 3 Path1 of 4, min refueling cost = $33.304

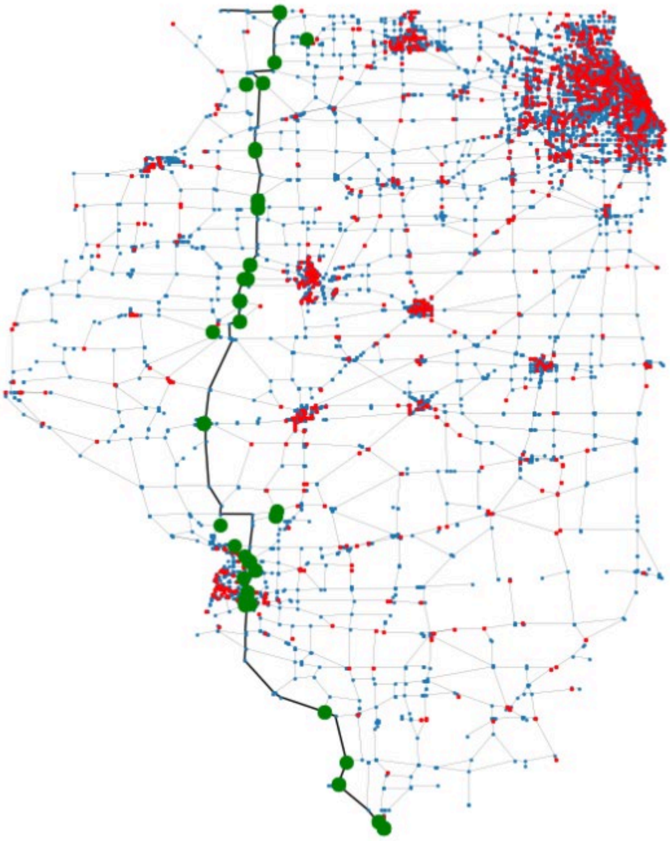

Figure 4 Path 2 of 4 min refueling cost = $33.32

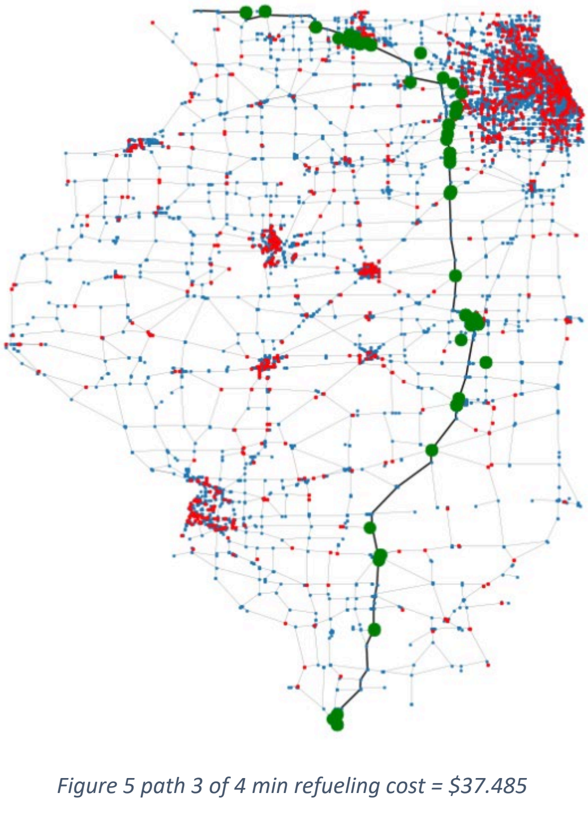

Figure 5 path 3 of 4 min refueling cost = $37.485

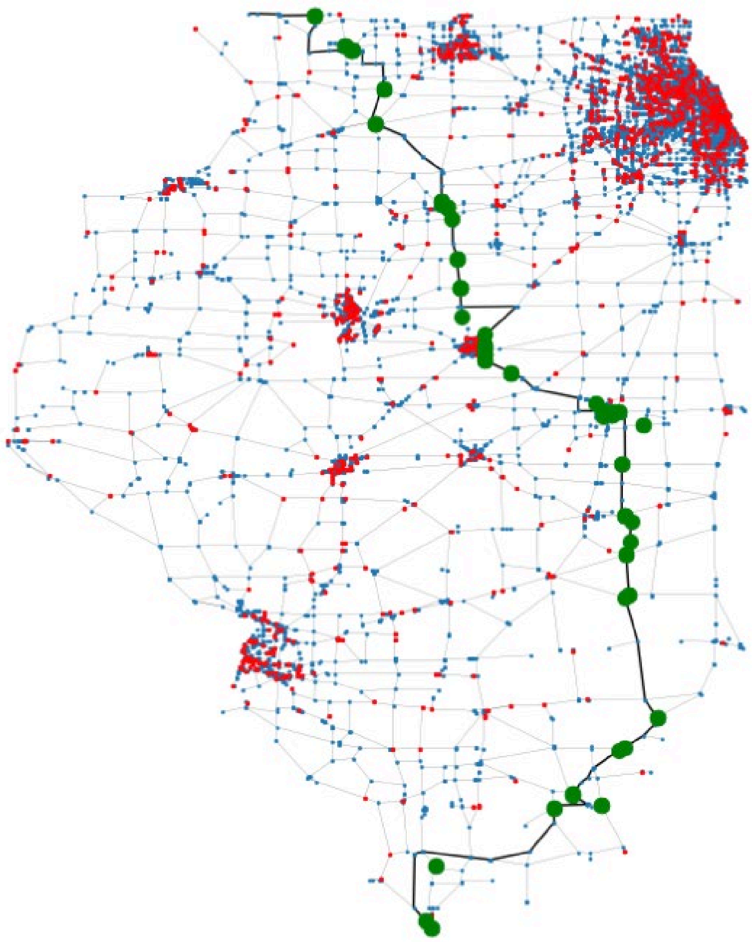

Figure 6 path 4 of 4, min refueling cost = $45.815

Out of these 4 paths, we picked the first path in this instance. The path picked naturally depends on the location of gas stations which are randomly generated hence this problem generates different results on each run.

With the first path picked, we also return the gas stations where the user should fill up along with the fill up quantities. For the Illinois data graph and the above instance, users had to only fill up once. The opt refueling station is shown in Figure 7.

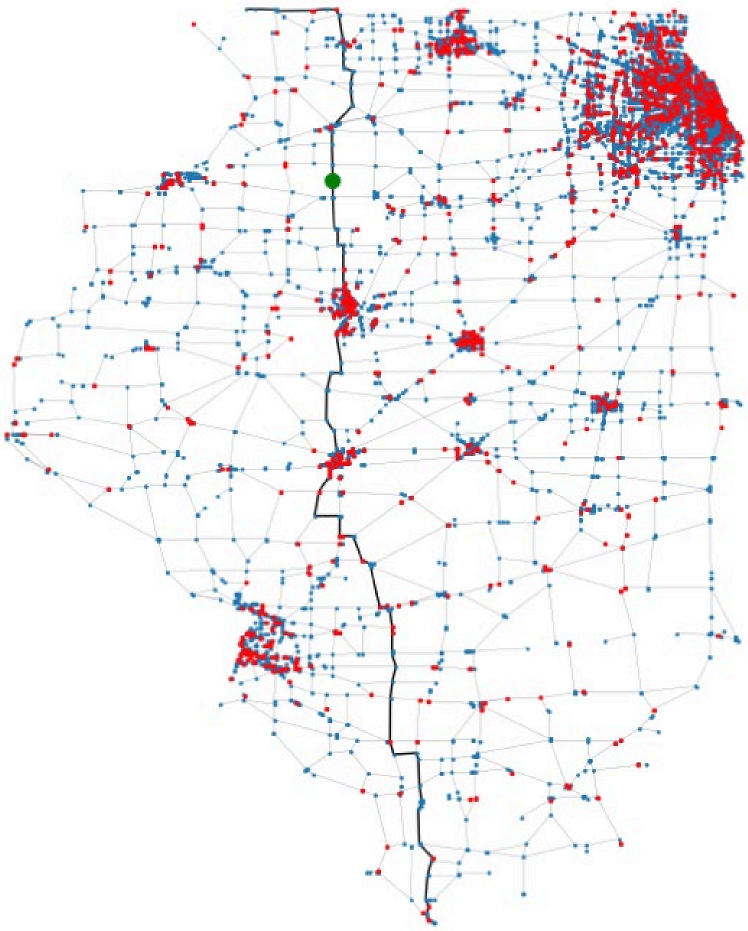

Figure 7 Optimum refueling station, fill up qty = 8 gallons

Limitations

Due to the unavailability of data, resources, and time, several assumptions have been carried out in this work. For resource limitation, we only stick to the state of Illinois, considering it as the complete network. Moreover, to reduce the size of the graph for maintaining sufficient computational efficiency, our work is limited to only primary and secondary highways. A bigger graph can be incorporated to generalize the model and tackle this limitation. In addition to it, our model considers homogenous consumption rate (MPG) for all the out of route roads. This assumption may not work for all the routes as the MPG can vary from one class of roads to another. An important extension of the work can be done by incorporating the road-class information (MPG by road class) into the model to improve $Fuel_i$ calculation.

References

Kan, J. K. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 221-227.

Lim, M. K. (2005). The flow-refueling location problem for alternative-fuel vehicles. *Socio-Eco Plan Sci*, 125–145.

Lin, S. (2008). Finding optimal refueling policies in transportation networks. *Proc 4th Int Conf Algorithmic Aspects Info Management*, 280-291.

R. Tavakkoli-Moghaddam, A. R. (2006). A memetic algorithm for a vehicle routing problem with backhauls. *Applied Mathematics*, 1049-1060.

Ralphs, T. K. (2003). On the capacitated vehicle routing problem. *Mathematical Programming*, 343-359.

S. Khuller, A. M. (2008). To fill or not to fill: The gas station problem. *Proc 15th Ann Euro Symp Algorithms*, 534-545.

S.H. Lin, R. G. (2007). A linear-time algorithm for finding optimal vehicle refueling policies. *Oper Res Lett*, 290-296.

Suzuki, Y. (2008). A Generic Model of Motor-Carrier Fuel Optimization. *Naval Research Logistics*.