Project Report

Title: Alternative Fuel Station Location Optimization in the Twin Cities Metropolitan Area
Notice: Dr. Bryan Runck
Author: Luke Zaruba
Date: December 21, 2022

Project GitHub Repository: Link here

Abstract
With renewable energy and sustainable transportation becoming key focuses for urban planning and development in the coming decades, infrastructure for supplying sustainable alternative fuels for transportation will become a key issue, especially given the United States' reliance upon gas and oil, along with the country's aging energy system. In order to efficiently and effectively provide adequate alternative fueling infrastructure to the nation, GIS and spatial data science techniques can be used to optimize coverage and minimize costs. This project aims to use two prominent methods developed in the field of operations research, for optimizing the rollout of alternative fueling infrastructure in the Twin Cities Metropolitan Area (TCMA) by maximizing coverage of demand and minimizing the resources needed to do so. The project shows how the techniques can be used at a local or regional level, but the analysis is scalable and can be used at both larger and smaller scales.

Problem Statement
The task at hand is to optimize new alternative fuel infrastructure placement to maximize the coverage of potential customers. As new infrastructure is being planned and developed, it should be a priority to create continuous coverage along major road networks, before beginning to implement alternative fuelling into a broader range of gas stations. By optimizing the placement of the infrastructure, fuelling can be implemented as seamlessly as possible into our changing energy system.

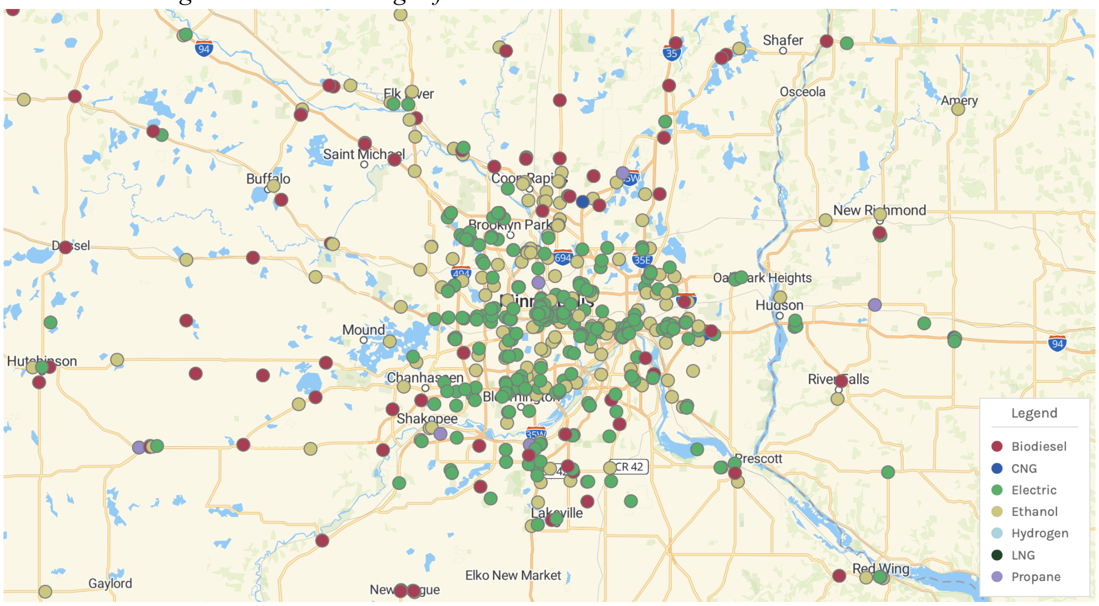
Table 1. Requirements for the analysis

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
2	Road Network	Met Council Functional Class Roads Dataset	Road geometry	Route system	MN Geospatial Commons	None
3	Candidate Facilities	Gas Station results from Google Places API	Point geometry	None	Google Places API	Combining multiple searches and cleaning up duplicates
4	Demand Points	Randomly generated points along the road network to simulate customers	Point geometry	None	Self-created	None

Input Data
The data needed in the analysis is comprised of four main components, a network, existing infrastructure, potential locations, and potential customers. The network is simply a road centerline dataset of the TCMA that only includes functional class roads. Functional classes are a system for classifying major roads and highways that was developed by the Federal Highway Administration, meaning that the dataset only contains roads that have higher traffic levels

and are used more frequently. This is beneficial to the analysis because stations located on these roads are more accessible to more people because they are not on small, local roads.
Existing infrastructure can be extracted from the U.S. Department of Energy Alternative Fuels Data Center's (AFDC) API, which contains all alternative fuel stations in the United States and Canada. However, this is not as useful if there is little to no current existing infrastructure, which is the case for most systems in the TCMA, other than electric charging stations and ethanol. For the sake of this project, existing infrastructure will not be taken into consideration, to simulate how the optimal spatial layout of the infrastructure would be configured, if it was built from scratch.

Figure 1. AFDC Existing Alternative Fueling Infrastructure in the TCMA.



Potential facility locations was be determined by using the Google Places API to search for gas stations. Multiple searches were performed, and then the datasets were merged, before finally removing duplicates and creating the final dataset. The easiest way to iteratively perform searches with the Google Places API is to create a grid pattern that the latitude and longitude parameters can follow and pass those values into the search function iteratively. Lastly, potential customers were simulated by creating points along the road network, which can be accomplished through several different approaches. The first is to simply create a fixed number of random points within the study extent, and then snap them to the nearest location on the road network. The second way is to randomly generate a fixed number of points, but instead of randomly distributing them across the entire study area, the dispersion can be weighted to allocate a certain number of points within different areas based on some characteristic of those areas, like population, for example. Lastly, points can be generated at a fixed distance along the road network, meaning that demand points will be consistently placed throughout the network. For the purposes of this project, the first option was selected, due to its ease of use and simplicity. With this decision, it is also easy to create multiple demand datasets and then use each dataset to test how strongly the results change based on the different locations of customers.

Table 2. Primary datasets that will be used in the analysis

#	Title	Purpose in Analysis	Link to Source
1	TCMA Roads	Dataset for network analysis from Met Council	[MN Geospatial Commons]()
2	Candidate Infrastructure	Potential locations for new alternative fueling infrastructure	[Google Places API]()
3	Potential Customers	Simulated customers along road network	Generated in analysis

Methods

There are several potential methods to solve the problem at hand, but two will be focused on, as potential solutions. The first method that can be used to solve the problem is the Location Set Covering Problem (Barcelos et al. 2020a), first proposed by Constantine Toregas (Toregas et al. 1971). The approach seeks to minimize the number of facilities needed to cover all areas within a maximum allowable measure of cost, like time or distance (Barcelos et al. 2020a). With this method, the number of facilities is not fixed, and 100% service coverage is guaranteed.

Figure 2. LSCP Optimization Problem (Barcelos et al. 2020a).

$$\begin{array}{lll} \textbf{Minimize} & \sum_{j=1}^{n} x_j & (1) \\ \textbf{Subject to:} & \sum_{j \in N_i} x_j \geq 1 \quad \forall i & (2) \\ & x_j \in 0, 1 \quad \forall j & (3) \end{array}$$

Where:

$$\begin{array}{lll} i & = & \text{index referencing nodes of the network as demand} \\ j & = & \text{index referencing nodes of the network as potential facility sites} \\ S & = & \text{maximal acceptable service distance or time standard} \\ d_{ij} & = & \text{shortest distance or travel time between nodes} i \text{and} j \\ N_i & = & \{j | d_{ij} < S\} \\ x_j & = & \begin{cases} 1, \text{if a facility is located at node} j \\ 0, \text{otherwise} \end{cases} \end{array}$$

The other approach to solving the problem is the Maximal Coverage Location Problem (MCLP), which was proposed in 1974 by Church and ReVelle (Church and ReVelle 1974). The goal of this method is not to guarantee 100% service coverage, but to maximize coverage with a set number of facilities, based on the fact that resources are often limited and constrained to a budget (Barcelos et al. 2020b).

Figure 3. MCLP Optimization Problem (Barcelos et al. 2020b).

$$\begin{array}{lll} \textbf{Maximize} & \sum_{i=1}^{n} a_i y_i & (1) \\ \textbf{Subject to:} & \sum_{j \in N_i} x_j \geq y_i \quad \forall i & (2) \\ & \sum_j x_j = p \quad \forall j & (3) \\ & y_i \in \{0, 1\} \quad \forall i & (4) \\ & x_j \in \{0, 1\} \quad \forall j & (5) \end{array}$$

Where:

$$\begin{array}{lll} i & = & \text{index referencing nodes of the network as demand} \\ j & = & \text{index referencing nodes of the network as potential facility sites} \\ S & = & \text{maximal acceptable service distance or time standard} \\ d_{ij} & = & \text{shortest distance or travel time between nodes} i \text{and} j \\ N_i & = & \{j | d_{ij} < S\} \\ p & = & \text{number of facilities to be located} \\ x_j & = & \begin{cases} 1, \text{if a facility is located at node } j \\ 0, \text{otherwise} \end{cases} \\ y_i & = & \begin{cases} 1, \text{if demand } i \text{ is covered within a service standard} \\ 0, \text{otherwise} \end{cases} \end{array}$$

For these two methods, the goal was then to use them both and sweep through various combinations of parameters to test how the outcomes changed. With the LSCP models, a model was created for each impedance threshold value from 0.5 to 2.0, with a step value of 0.5, and each of those models was tested on three different input demand datasets. Impedance threshold functions determine how distance behaves within a function, by defining how distances are weighted in the model as they change. The process for MCLP was slightly different because instead of using a "power" distance impedance function like in LSCP, a linear impedance function was used, meaning that no impedance threshold value was used. Instead, different numbers of facilities were tested (15, 20, and 25), each on the three different demand datasets. This made it so that each model was created within a nested loop and solved, before moving on to the next selection of parameters. After each iteration of solving a model, the number of facilities that were chosen (for LSCP) or the number of demand points covered (for MCLP) was calculated. With the completion of all models, summary statistics were calculated to determine the number of times that a facility was chosen throughout each model run. If a facility was chosen in every model for LSCP, then it would have a value of 12, for the 12 different models that were created, and if it was never chosen, it would have a value of 0. In the case of MCLP, the maximum score would be 9 and the minimum would again be 0. In figures 4 and 5, the data flow diagram and algorithms used for the analysis can be seen in greater detail.

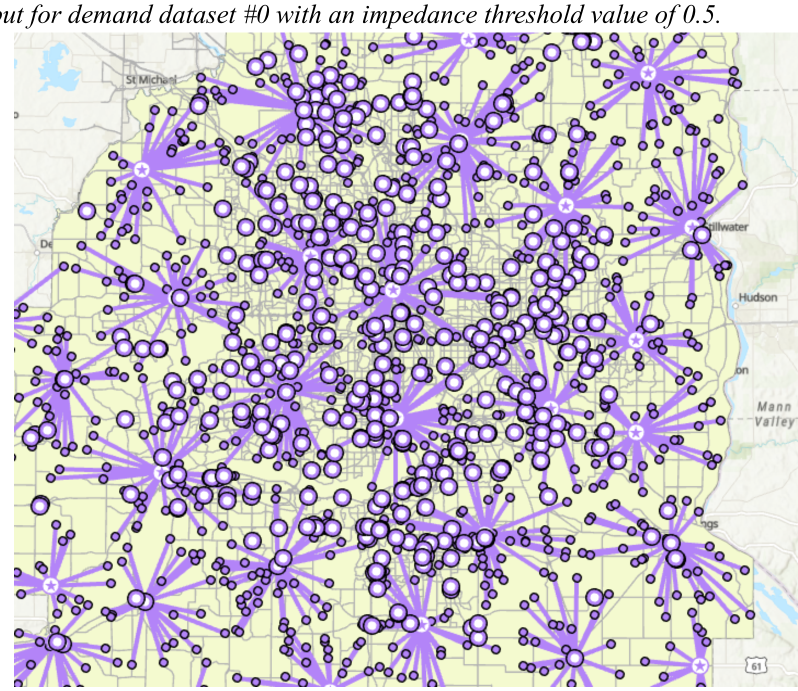
Figures 4 (left) and 5 (right). Data-flow diagram and algorithms (in pseudocode).


```
function optimalRadius(step)
  idealValue <= 111139 * step
  roundedValue <= round(idealValue, -3)
  if roundedValue < idealValue then
    roundedValue <= roundedValue + 1000
  end if
  return roundedValue


function searchGoogle(latitude, longitude, radius, keyword, json)
  params => (latitude, longitude, radius, keyword)
  response => get(base URL, payload = params)
  if json is true then
    return response.json
  end if
  else
    df <= pandas.DataFrame()
    for each feature in response.json
      attributes <= []
      attributes.append(feature.values)
      df.append(attributes)
    end for
    gdf <= df.toGDF()
    return gdf
  end else


function iterateSearch(xMin, xMax, yMin, yMax, step, keyword)
  radius <= optimalRadius(step)
  finalDF <= pandas.DataFrame()
  for x in range(xMin, xMax, step)
    for y in range(yMin, yMax, step)
      finalDF.concat(searchGoogle(y, x, radius, keyword, false))
    end for
  end for
  return finalDF.drop_duplicates()


function createDemandPoints(extent, nPoints, nDatasets)
  for i in range(nDatasets)
    arcpy.CreateRandomPoints(extent, nPoints)
  end for


function LSCP()
  for i in range(0.5, 2.0, 0.5)
    for j in range(3)
      model <= arcpy.makeLocationAnalysisLayer(MinFacilities, power=i, threshold=15000)
      arcpy.addLocations(model, demand, f"demand_{j}")
      arcpy.addLocations(model, demand, candidates)
      arcpy.Solve(model)
    end for
  end for


function MCLP()
  for i in range(15, 26, 5)
    for j in range(3)
      model <= arcpy.makeLocationAnalysisLayer(MaxCoverage, nFacilities=i, threshold=15000)
      arcpy.addLocations(model, demand, f"demand_{j}")
      arcpy.addLocations(model, demand, candidates)
      arcpy.Solve(model)
    end for
  end for
```

Results

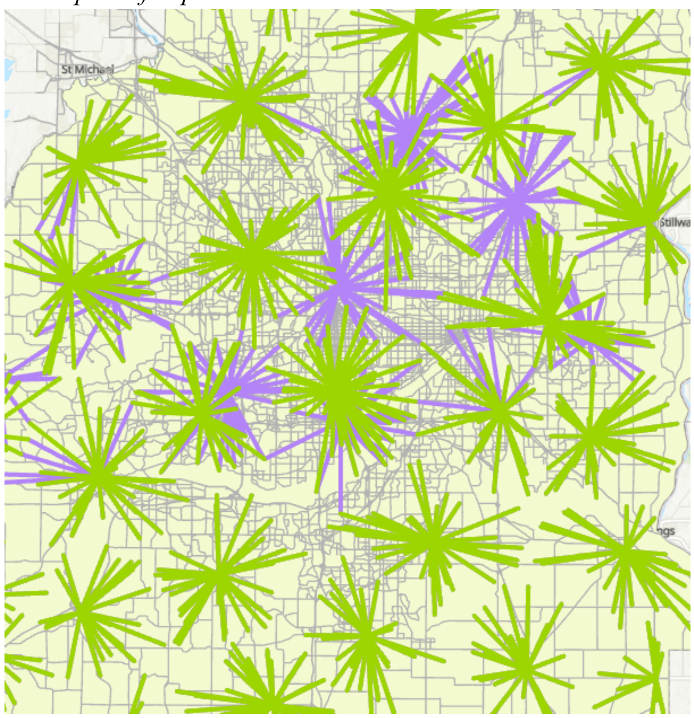
The results of the models show some clear and expected trends. An example of a model output can be seen in figure 6, which shows candidate facilities with larger white circles with purple outlines, chosen facilities with stars, and demand points with small purple circles. Additionally, the lines show which demand points were covered by which chosen facility.

Figure 6. LSCP output for demand dataset #0 with an impedance threshold value of 0.5.

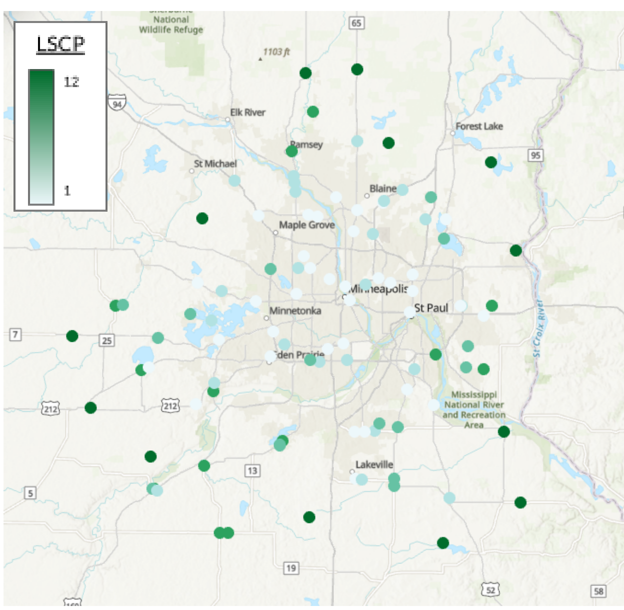


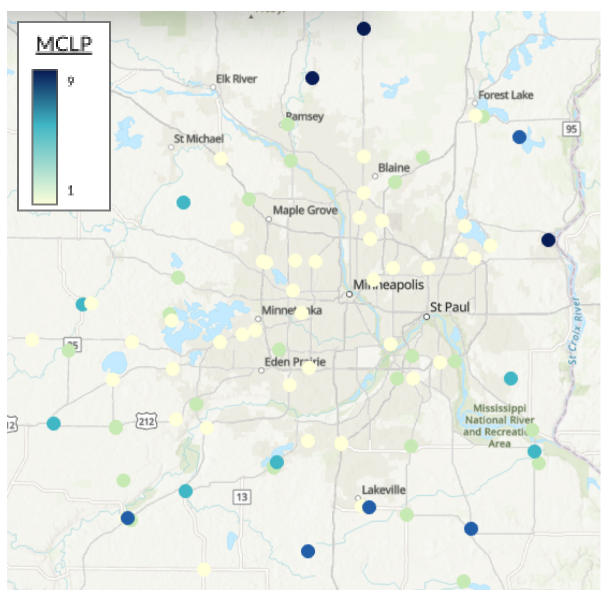
Changing the impedance threshold values for LSCP had very little effect on the model output, as seen in figure 7, where most of the purple lines (for a threshold value of 2.0) are covered by the green lines (for a threshold value of 0.5) since they are identical. The MCLP results show very similar results to the LSCP models, with the exception of a far fewer number of facilities being needed to cover only 98 to 99 percent of the demand. This will be discussed further in the next section of the report. Overall, it is clear that facilities near the edge of the project area were chosen more frequently. This can likely be attributed to the fact that these areas are more rural and there are fewer options to choose from, whereas there are many options in the more urban areas making it easier for those to change from changes to model parameters. The final results of the LSCP and MCLP models can be seen in figures 8 and 9, respectively.

Figure 7. A comparison of the impact of impedance threshold values on LSCP model results.



Figures 8 & 9. Facilities by the number of times they were chosen (minimum of at least once).





Discussion and Conclusion

Verifying optimization models looks a little different since there is nothing to compare results to other than different variants of the model itself, meaning that verification looks more like a sensitivity analysis. Sensitivity analyses are used to test how different parameter choices affect outcomes, and that is exactly what can be seen in tables 3 and 4, with the first two columns of each table showing parameter choices and the third column showing a metric with which results can be compared against each other. With the LSCP models, there is very little change seen between the different parameters, with the number of facilities chosen ranging from 33 to 35. This may show that the parameters that were chosen did not vary enough, or it may show that they do not have a strong impact on the results, regardless. With the MCLP results, the same patterns can be seen, with very little variation in the outputs of the models, in terms of the number of demand points covered.

Table 3. LSCP Sensitivity Analysis.

Impedance Threshold of 0.5	Demand Dataset #0	35 facilities chosen
	Demand Dataset #1	33 facilities chosen
	Demand Dataset #2	33 facilities chosen
Impedance Threshold of 1.0	Demand Dataset #0	34 facilities chosen
	Demand Dataset #1	34 facilities chosen
	Demand Dataset #2	33 facilities chosen
Impedance Threshold of 1.5	Demand Dataset #0	34 facilities chosen
	Demand Dataset #1	33 facilities chosen
	Demand Dataset #2	34 facilities chosen
Impedance Threshold of 2.0	Demand Dataset #0	34 facilities chosen
	Demand Dataset #1	33 facilities chosen
	Demand Dataset #2	34 facilities chosen

Table 4. MCLP Sensitivity Analysis

15 facilities	Demand Dataset #0	982 of 1000 covered (98.2%)
	Demand Dataset #1	987 of 1000 covered (98.7%)
	Demand Dataset #2	987 of 1000 covered (98.7%)
20 facilities	Demand Dataset #0	995 of 1000 covered (99.5%)
	Demand Dataset #1	997 of 1000 covered (99.7%)
	Demand Dataset #2	997 of 1000 covered (99.7%)
25 facilities	Demand Dataset #0	995 of 1000 covered (99.5%)
	Demand Dataset #1	997 of 1000 covered (99.7%)
	Demand Dataset #2	997 of 1000 covered (99.7%)

What can be seen, however, is that there are far fewer facilities used to cover almost all of the demand. This is likely due to the difference in the impedance threshold function selected for the MCLP models compared to the LSCP models. Had a linear function been used for LSCP, just as it was for MCLP, I believe that the number of facilities chosen would likely have ranged from 25 to 30. This shows the importance of making informed decisions regarding the choices that are made for parameter values and how impactful user choices can be on affecting the output of a model. With that being said, there is no right or wrong parameter choice, as they simply are used to model different real-world scenarios.

In summary, there are a few key takeaways from this analysis. First, when using a linear impedance threshold function, based on my inferences, it would take approximately 10 facilities to achieve a coverage of 90% of demand within 15 km of a facility. Second, to achieve 100% coverage of all demand within 15 km, it would take roughly 25-30 facilities, when using a linear impedance function. Third, if a “power” impedance threshold function is used, then the number of facilities needed for both the first and second points increases by about 10. These findings are merely trends that I have picked up on, and are not reflected by any actual results. Due to the processing power needed to perform each model iteration, not all possibilities were able to be modeled, but if the project were to continue, the highest priority would be to test some of the hypotheses mentioned.

References

Barcellos, Germano, James D. Gaboardi, Levi J. Wolf, and Qunshan Zhao. "Location Set Covering Problem (LSCP)." PySAL. 2020a. https://pysal.org/spopt/notebooks/lscp.html#Location-Set-Covering-Problem-(LSCP).

Barcellos, Germano, James D. Gaboardi, Levi J. Wolf, and Qunshan Zhao. "Maximal Coverage Location Problem." PySAL. 2020b. https://pysal.org/spopt/notebooks/mclp.html.

Church, Richard, and Charles ReVelle. "The maximal covering location problem." Papers of the Regional Science Association. Vol. 32, No. 1. 1974.

Toregas, Constantine, Ralph Swain, Charles ReVelle, and Lawrence Bergman. "The location of emergency service facilities." Operations Research. Vol. 19, No. 6, pp. 1363-1373. 1971.

U.S. Department of Energy. "Alternative fueling station locator." Alternative Fuels Data Center: Alternative Fueling Station Locator. U.S. Department of Energy. 2022. https://afdc.energy.gov/stations/#/find/nearest.

Self-score

Category	Description	Points Possible	Score
Structural Elements	All elements of a lab report are included **(2 points each)**: Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	**28**
Clarity of Content	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**.	24	**24**
Reproducibility	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	**27**
Verification	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**.	20	**17**
		100	**96**