VIP Notebook

Tejas Vermani

SEMESTER 2: FALL 2023

Week #: 1-3

Assigned Tasks:

- Read the paper “Karsts2004Accessibility” carefully. Each of you should read it individually. It is a fundamental paper in the field of transportation accessibility. We will later model “Accessibility” as an objective function in our multi-objective model.
- Find a suitable time to meet together after each of you are done reading it. Discuss the paper. Prepare a 15-min presentation summarizing the paper. Follow these steps:
    - Discuss what should be the content of the presentation (in terms of slide titles)
    - Divide the content among yourselves and prepare the corresponding slides.
    - Send me the following via email before our next week’s meeting:
        - The presentation file (in PowerPoint format).
        - List your names and percentage contribution (in preparing the presentation) in the body of the email. For example, Tejas 50%, Henok 50%. The two of you need to come to a consensus about this distribution.

What I did:

- Henok and I created said presentation and presented it to the subteam, learning about different accessibility measures and how newer models are looking to integrate more complex (location-based and utility-based) accessibility metrics, taking into account factors more comprehensive than just distance travelled between stations, time to station, and other simpler metrics used historically.

What I learned:

- I learned about the more complex metrics that are starting to be integrated into the models, as well as the different approaches to figuring out how to optimize accessibility to public transport within real-world constraints like budget.
- Gravity models: Gravity models use travel time and travel cost to calculate the accessibility of a particular location. They are relatively easy to calculate, but they can be inaccurate if they do not take into account the attractiveness of destinations.
- Cumulative opportunity measures: Cumulative opportunity measures calculate the accessibility of a location by counting the number of destinations that can be reached within a certain amount of time or for a certain amount of money. They are more accurate than gravity models, but they can be more complex to calculate.
- Isochrones: Isochrones are maps that show the areas that can be reached within a certain amount of time. They are useful for visualizing accessibility, but they can be difficult to interpret.

Challenges:

- The paper was quite dense, and understanding the actual mathematical equations that went into the modeling was quite difficult as well, but via collaboration with Henok, we were able to comprehend the terms and concepts, and will hopefully be able to apply the learning to our own models.

Week 4-5:

- **Tasks:**
    - Read the paper "Yang2021Optimizing" carefully. Each of you should read it individually.
        - Find a suitable time to meet together after each of you are done reading it. Discuss the paper. Prepare a 15-min presentation summarizing the paper. Follow these steps:
            - Discuss what should be the content of the presentation (in terms of slide titles)
            - Divide the content among yourselves and prepare the corresponding slides.
            - Send me the following via email before our next week's meeting:
                - The presentation file (in PowerPoint format).
                - List your names and percentage contribution (in preparing the presentation) in the body of the email.

What I did:

- Henok and I created said presentation on the Yang 2021 Paper, which mainly covered the importance of accessibility and efficiency of public transportation, maximizing the level of transit services under the constraint of a limited budget that will be invested in the new constructions of public transit stops, and minimizing the total cost of the project, which was essentially a multi-objective optimization problem that we need to learn to model for our own purposes.

What I learned:

- We saw the strategies and models of this research group play out in a case study done in Charlotte, NC, where some of the conclusions they reached were that systematic redesign for all block groups might be needed, and that in both models, the block groups chosen to have new constructions would cover entire block zones. Furthermore, they acknowledged the service redundancy of creating new stations in areas that don't heavily rely on public transport (wealthier areas) or block zones that are already saturated with availability to public transport relative to their population and demographics.

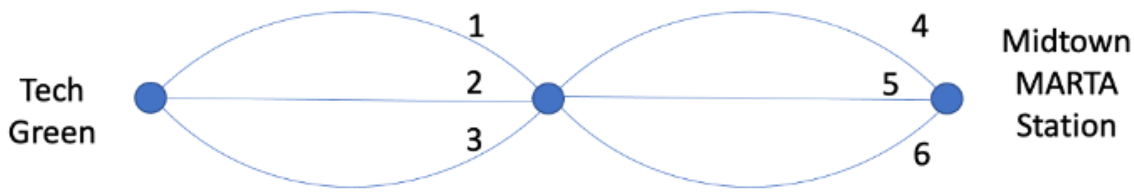
Challenges:

- Again, the paper and the mathematics were slightly difficult to interpret right off the bat, but again we solved that with brute force (re-reading the paper) and collaborative explanations between Henok and myself.

Week #: 6-7

Assigned Tasks:

- Carefully read Chapter 18 (Arora) Sections: 18.1, 18.2, 18.4, 18.5, 18.7, 18.8.
- Figure 1 shows alternative routes (i.e.., paths) between Tech Green and the Midtown MARTA station.



$$t_1 = 0.3x_1$$
$$t_2 = 1 + 0.2x_2$$
$$t_3 = 7 + 0.1x_3$$
$$t_4 = 15 + 0.3x_4$$
$$t_5 = 5 + 0.2x_5$$
$$t_6 = 8 + 0.1x_6$$

t: link performance function (in min)
x: flow on the link (vehicles per hour)
Demand: 200 (vehicles per hour)

- (i) Suppose we want to assign those 200 vehicles among the different routes. We have two objectives
- Formulate this problem as a multi-objective optimization problem. Specifically, list all the parameters, decision variables, objectives (given above), and constraints.
- Solve the formulated problem considering objective 1 and objective 2 independently. In other words, solve two single-objective problems.

What I did:

- We modeled the objectives in python as objective functions by establishing 15 decision variables, 6 for the links and 9 for the paths.

- We then studied the scipy.optimize.minimize function to see how to format data properly for the library function, and we studied what the output would be: the output was a unique data structure that had to be parsed in order to get the values we wanted, which we took and plugged back into the objective functions to results for them.

What I learned:

- I learned how to use the scipy optimize minimze function, as well as how to program an objective function with multiple decision variables. I did this twice for the two objectives in this problem.

Challenges:

- Digging through the documentation of scipy to figure out how to format the parameters was a challenge, and the same applies for the result once we did figure that out; how to parse through the resulting data structure that contained the data we needed.
- Also, programming remotely with the other members of the team for the first time was a bit of a struggle, so I created a github repo to allow access and efficiently share our changes to the code.

Week #: 8-9

Assigned Tasks:

- Solve the formulated problem considering both the objectives (you can find them in the task documents from the previous week). Use the weighted-sum approach (Arora 18.4).
- Before applying the weighted-sum approach, you need to normalize the objective functions. Use the approach discussed in 18.2.6 to normalize them. While determining , use the first approach discussed in that section.
- After solving the problem for different set of weights, plot the Pareto optimal frontier

What I did:

- This assignment was a continuation of the first assignment of programming the objective functions individually; this time, we needed to apply the weighted-sum scalar method to combine the two individual normalized objective functions into one that could be plugged into the scipy optimize minimize function. We did this with a loop iterating from 0 to 1 in increments of 0.1, applying the weight to one objective and 1 - the weight to the other weight (sum of the weights is 1). For each weight combination, we get a pareto point, which we plotted to get the pareto frontier.

What I learned:

- I learned the value of being able to compress multiple objectives into one, as we can leverage libraries like scipy and other tools and algorithms.
- We also learned how to correctly parse through the output of the minimize functions, which we needed to do to get values to plug into the objective functions to get our pareto points.

Challenges:

- Thinking through each step was difficult, as there were programming issues we ran into.

- For example, I didn’t realize that if the callable objective function that I was passing into minimize() as the first parameter took parameters itself, I could pass them into the minimize call as well in the form of a tuple under the param name “args”. This helped massively as we needed to pass the weight into the function in order to get the proper outputs.

Week #: 10-13

Assigned Tasks:

- Solve the same multi-objective problem that you solved last time using the weighted average method. However, this time, first linearize the two objective functions using first-order approximation. Then, use these linearized objectives to solve the problem. Solve the same problem using the NSGA-II algorithm using Pymoo (https://pymoo.org/)
- Compare the Pareto frontiers you obtained after solving the same problem using three different approaches.
- Me specifically: prepare to present my portion of the final presentation.

What I did:

- We attempted this multiple ways:
  - First we attempted to leverage the sympy library's linearize capabilities to linearize our objective functions.
  - We also attempted to use the scipy equivalent, and did a full implementation that outputted something that didn't resemble what we expected it to.

What I learned:

- What I learned was that as the complexity of these problems increases, I need to be even more precise at each step, making sure that I am using the correct libraries and functions for each step.

Challenges:

- It turns out what we had to do was a Taylor series expansion, and because we had 6 links, we had to do it to the 6th degree, which our team leader admitted would be too difficult to program in the time we had. I would love to attempt it again later on, and it would be interesting to see the results compared to those of the weighted sum approach and the pymoo NSGA-II results.

Final Presentation

Slide 1: Thank you Alex. Apart from learning about traffic assignment problems and different emerging performance measures, we explored different solvers and algorithms for multi-objective optimization. We used this simple traffic assignment problem with both user equilibrium and system optimal as objectives. Our motivation behind solving this problem was to learn how traffic assignment problems can be modeled and solved using different python libraries and algorithms, so that we can do the same for the real network of the City of Peachtree Corners, while optimizing safety, mobility, accessibility, and equity.

To solve this problem, we applied some scalar methods. The diagram on the left shows four different scalar methods we explored. For illustration of the solution, I am using the weighted-sum approach, which we implemented using the Python library Scipy. This approach entails assigning a weight to each normalized objective, where the sum of the weights applied is 1, and solving the problem as a single-objective problem. We repeated the procedure 11 times by varying the weights from 0 to 1 at an increment of 0.10, each time obtaining one Pareto point. The obtained Pareto frontier is shown in this figure. Determining which point to select as the final solution, depends on the perspective of the decision-maker; our job is to present the Pareto frontier.

Slide 2: Generally, the scalar methods approximate the Pareto frontier efficiently. However, how close the approximation is to the true Pareto frontier is something we also wanted to determine. To do that, various vector methods can be used. To benchmark the Pareto frontier, a genetic algorithm-based approach called NSGA-II is usually used, which is computationally inefficient. Thus, we prefer methods like Weighted-sum for optimizing real-world problems. We used another Python library Pymoo which is developed for solving multi-objective problems. While trying to apply NSGA-II, we realized that this algorithm handles equality constraints poorly and terminates pre-maturely. Nonetheless, we explored the Pymoo library and solved a few example problems using NSGA-II to learn how the library and different built-in algorithms can be used to solve multi-objective problems.

Now I welcome Dennis from the other group to present their work.