Strategic Development of Graphical User Interface Tools for Enhancing Transportation Mobility Among Vulnerable Groups During Pandemics

Final Report

by

Yuche Chen

300 Main St., Columbia, SC 29208
University of South Carolina
Phone: (803) 777-9105; *E-mail:* chenyuc@cec.sc.edu

Xuanke Wu, Oak Ridge National Laboratory
Juan Caicedo, University of South Carolina
Yu Qian, University of South Carolina

December 2024

Center for Connected Multimodal Mobility (C^2M^2)

200 Lowry Hall, Clemson University
Clemson, SC 29634

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by the Center for Connected Multimodal Mobility (C^2M^2) (Tier 1 University Transportation Center) Grant, which is headquartered at Clemson University, Clemson, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Strategic Development of Graphical User Interface Tools for Enhancing Transportation Mobility Among Vulnerable Groups During Pandemics		5. Report Date December 2024 6. Performing Organization Code
7. Author(s) Yuche Chen, Ph.D.; ORCID: 0000-0003-2577-2448 Xuanke Wu Ph.D.; ORCID: 0000-0002-2525-0351 Juan Caicedo, Ph.D.; ORCID: 0000-0001-6409-231X Yu Qian, Ph.D.; ORCID: 0000-0001-8543-2774		8. Performing Organization Report No.
9. Performing Organization Name and Address 1. Department of Civil and Environmental Engineering University of South Carolina 300 Main St., Columbia, SC 29208 2. Buildings and Transportation Science Division Oak Ridge National Laboratory 1 Bethel Valley Rd, Oak Ridge, TN 37830		10. Work Unit No. 11. Contract or Grant No.
12. Sponsoring Agency Name and Address Center for Connected Multimodal Mobility (C2M2) USDOT Tier 1 University Transportation Center Clemson University 200 Lowry Hall, Clemson Clemson, SC 29634		13. Type of Report and Period Covered Final Report (March, 2024 – December, 2024) 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Safety concerns, social distancing requirements, and limited on-road vehicles during the pandemic have resulted in disruptions or reduced operations in many public transportation services, exposing underlying transportation mobility problems. Despite these challenges and travel restrictions, essential travel demands persist, such as accessing healthcare, grocery stores, employment opportunities, and other fundamental services. However, due to the decreased service frequency of buses, trains, and other modes of public transit, vulnerable groups are disproportionately affected by limited mobility options compared to other populations. This exacerbates existing inequalities and affects their ability to meet essential needs under this challenging circumstance. Disadvantaged and disabled travelers often rely more on public transportation to access those essential services for its affordable ticket prices. Therefore, addressing these social exclusion challenges and providing a minimal transportation service level is crucial to ensure equitable access to essential services for vulnerable populations throughout the pandemic and beyond. To improve the life quality of people with disabilities and elderly people by addressing social exclusion, accessibility, and mobility issues, previous team work has developed a car-sharing optimization model with Sample Average Approximation (SAA) and Rolling Horizon (RH) methods, so that city planners, Non-profit Organizations (NPOs), and some other policymakers are enabled to better allocate limited resources to implement the car-sharing system when little to no historical travel information is available for low-density population areas. Nevertheless, understanding and implementing the technical algorithm and framework can be challenging for individuals who lack sufficient training, primarily due to its complexity and the absence of user-friendly interfaces and/or step-by-step user manual guidance. In order to reduce the application barrier for general users, this project aims to develop a more user-friendly Graphical User Interface (GUI) using the Tkinter library within Python and provide comprehensive manual guidance.

17. Keywords		18. Distribution Statement	
GUI Development; Transportation Mobility; Vulnerable Passengers, Ride Sharing		No restrictions.	
19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	30	NA

Table of Contents

List of Tables

List of Figures

EXECUTIVE SUMMARY

Safety concerns, social distancing requirements, and limited on-road vehicles during the pandemic have resulted in disruptions or reduced operations in many public transportation services, exposing underlying transportation mobility problems. Despite these challenges and travel restrictions, essential travel demands persist, such as accessing healthcare, grocery stores, employment opportunities, and other fundamental services. However, due to the decreased service frequency of buses, trains, and other modes of public transit, vulnerable groups are disproportionately affected by limited mobility options compared to other populations. This exacerbates existing inequalities and affects their ability to meet essential needs under this challenging circumstance. Disadvantaged and disabled travelers often rely more on public transportation to access those essential services for its affordable ticket prices. Therefore, addressing these social exclusion challenges and providing a minimal transportation service level is crucial to ensure equitable access to essential services for vulnerable populations throughout the pandemic and beyond. To improve the life quality of people with disabilities and elderly people by addressing social exclusion, accessibility, and mobility issues, previous team work has developed a car-sharing optimization model with Sample Average Approximation (SAA) and Rolling Horizon (RH) methods, so that city planners, Non-profit Organizations (NPOs), and some other policymakers are enabled to better allocate limited resources to implement the car-sharing system when little to no historical travel information is available for low-density population areas.

The objective of this project was to develop a user-friendly Graphical User Interface (GUI) application to simplify the adoption of the developed algorithm for car-sharing systems, aiding decision-making for ride-sharing companies or other transportation agencies. Additionally, it provides comprehensive manual guidance. We used Tkinter, a standard Python package for GUI development, to streamline all core functions derived from the developed algorithms. These functions encompass interactive data input processes, such as retrieving city network data and user-defined points of interest (POI) like retirement communities, healthcare centers, grocery stores, entertainment clubs, etc. The application also handles automated data processing, calculations, decision-making, and result reporting. Once established, this tool can be applied to other locations (cities, states, or regions) to assist decision-makers with their car-sharing operation networks, provided that basic input data are supplied. Therefore, public transportation practitioners only need to focus on providing transportation services according to the scheduled plans, maintaining a high level of service while keeping within a reasonable budget.

In this project, a series of car-sharing system plans for vulnerable groups during pandemics was conducted using the road networks of Charleston, Clemson, Columbia, and Myrtle Beach in South Carolina. At each step of the streamlined procedure, the geographical location of the city, POI locations, and key parameters are inputted and visualized. This approach provides a more straightforward understanding of the system's performance, making it a more user-friendly tool for individuals who lack sufficient training.

CHAPTER 1
INTRODUCTION AND BACKGROUND

As the global population grows, ages, and urbanizes, it becomes more and more crucial to ensure accessible transportation, ensuring that neither aging nor disability creates barriers to social inclusion (Sochor & Nikitas, 2016). Matherly and Mobley highlighted significant gaps in communicating with vulnerable populations about transportation during emergency situations (Matherly & Mobley, 2011). In times of emergency, the resilience of the transportation system is measured by its ability to withstand disruptions and adapt to changing conditions in order to safely meet travel demands. (AbdelMagid et al., 2023). In the past several years, the COVID-19 pandemic has caused significant disruptions to transportation systems worldwide, exacerbating mobility challenges, especially among vulnerable populations such as older adults, ethnic minority groups, rural, and disabled populations. These groups are particularly affected as they are more likely to lack access to vehicle ownership or the physical capability to operate a vehicle (Liu et al., 2023). Consequently, shared transport services are being considered as a potential solution to address transport poverty and social exclusion during these times (De Paepe et al., 2023).

Safety concerns, social distancing requirements, and limited on-road vehicles during the pandemic have resulted in disruptions or reduced operations in many public transportation services, exposing underlying transportation mobility problems. Despite these challenges and travel restrictions, essential travel demands persist, such as accessing healthcare, grocery stores, employment opportunities, and other fundamental services. However, due to the decreased service frequency of buses, trains, and other modes of public transit, vulnerable groups are disproportionately affected by limited mobility options compared to other populations. This exacerbates existing inequalities and affects their ability to meet essential needs under this challenging circumstance. Disadvantaged and disabled travelers often rely more on public transportation to access those essential services for its affordable ticket prices. Therefore, addressing these social exclusion challenges and providing a minimal transportation service level is crucial to ensure equitable access to essential services for vulnerable populations throughout the pandemic and beyond.

To improve the life quality of people with disabilities and elderly people by addressing social exclusion, accessibility, and mobility issues, Dr. Yu Qian has previously developed a car-sharing optimization model with Sample Average Approximation (SAA) and Rolling Horizon (RH) methods (Liu et al., 2023), so that city planners, Non-profit Organizations (NPOs), and some other policymakers are enabled to better allocate limited resources to implement the car-sharing system when little to no historical travel information is available for low-density population areas.

Nevertheless, understanding and implementing the technical algorithm and framework can be challenging for individuals who lack sufficient training, primarily due to its complexity and the absence of user-friendly interfaces and/or step-by-step user manual guidance. Tkinter is a standard built-in package in Python for coding Graphical User Interface (GUI) programs. It is stable and straightforward, though not without its imperfections (Moore, 2018). Therefore, in order to reduce the application barrier for general users, this project aims to develop a more user-friendly GUI using the Tkinter library within Python and provide comprehensive manual guidance (Conway, 1995). The project framework workflow is illustrated in Fig. 1.1.

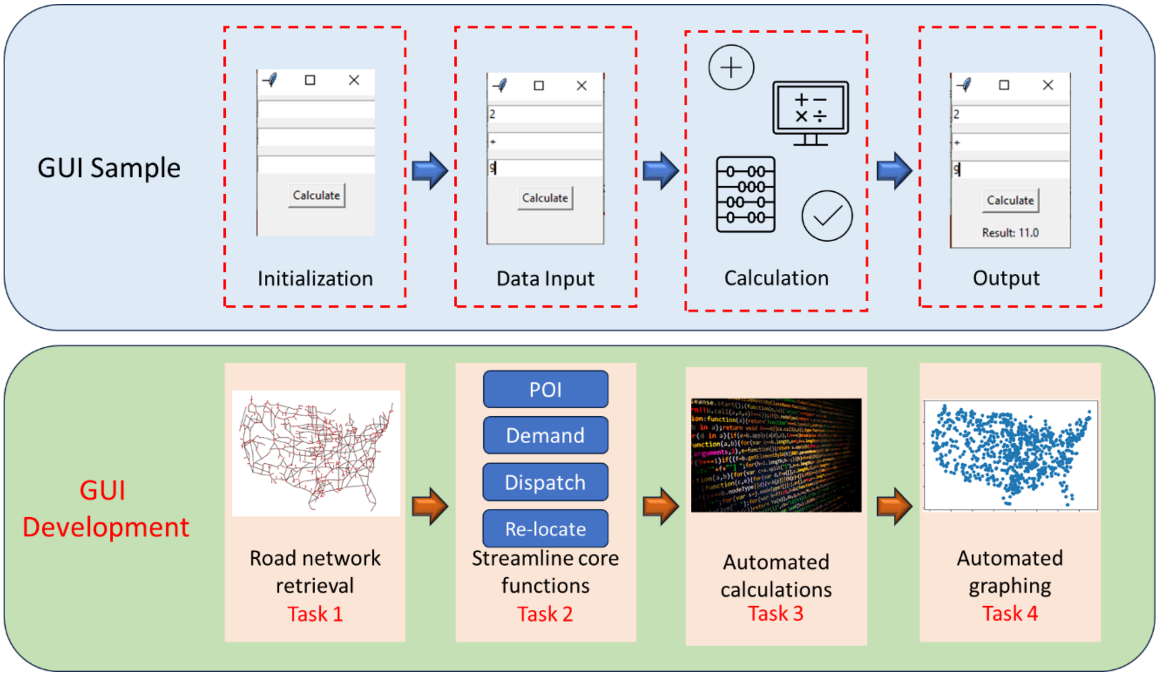

Figure 1.1 Project Framework Overview.

The objectives of this study are as follows.

1. Develop a user-friendly GUI application to simplify the adoption of the developed algorithm for car-sharing systems, aiding decision-making for ride-sharing companies or other transportation agencies.
2. Translate the complexities of existing algorithms into a user-friendly interface accessible to personnel with varying levels of technical expertise.
3. Provide a comprehensive user manual to guide users in utilizing the GUI application effectively.

The remainder of this report is organized as follows: Chapter 2 discusses the development of the GUI tool, Chapter 3 provides a detailed user guide with two application scenarios, and Chapter 4 summarizes the study and presents the conclusions.

CHAPTER 2
TOOL DEVELOPMENT

This chapter covers the development of the GUI tool, from the framework and data preparation to methods for generating passenger demand.

2.1 Framework

A detailed framework on transportation mobility analysis tool development is presented in Fig. 2.1. There are mainly four stages for the tool development in the flowchart: Preparation, Coding, Testing, and Launching.

- Preparation Stage:
  - Data preparation: Acquiring coordination information for target locations (including the selected city and related POIs), road networks, travel distances, and times from Google Maps and OpenStreetMap.
  - Passenger demand: Generating spatial-temporal passenger demand distributions using the Monte Carlo Simulation Method.

- Coding Stage:
  Developing the GUI application with Tkinter, a standard Python package for GUI development, which includes data input, automated calculations, and result visualization.

- Testing Stage:
  Testing the functions of the developed GUI application, including city and POI selection, input of user-defined parameters, and analysis functions.

- Launching Stage:
  Once the GUI tool passes functional testing, compiling the original code into an executable file. This makes it easier to use the tool without needing to install Python and its supporting packages.

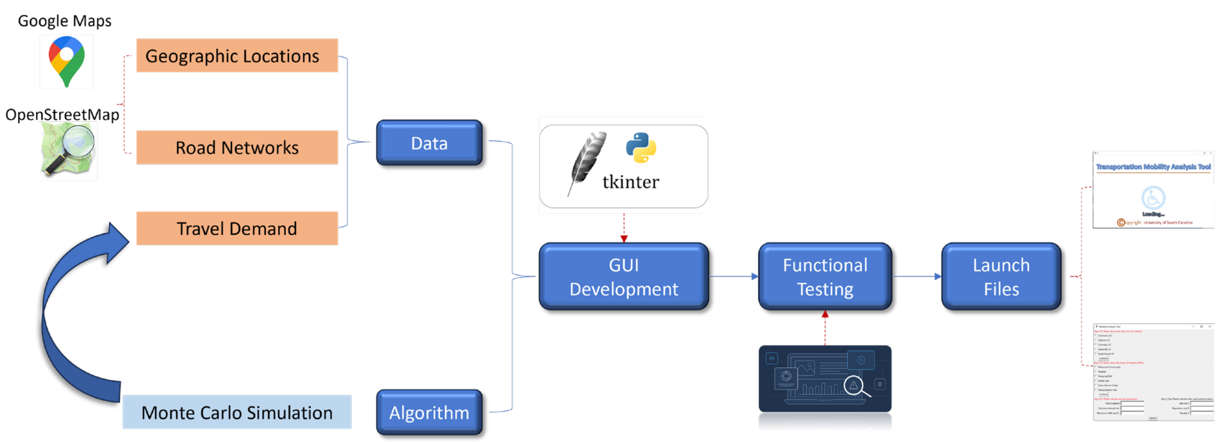

Figure 2.1 Development Flowchart.

2.2 Data Preparation

In this project, our primary focus is on five metropolitan areas within the State of South Carolina (SC): Charleston (Charleston County), Clemson (Pickens County), Columbia (Richland County), Greenville (Greenville County), and Myrtle Beach (Horry County). All basic county networks are sourced from OpenStreetMap as shown in Figures 2.2 through 2.6.

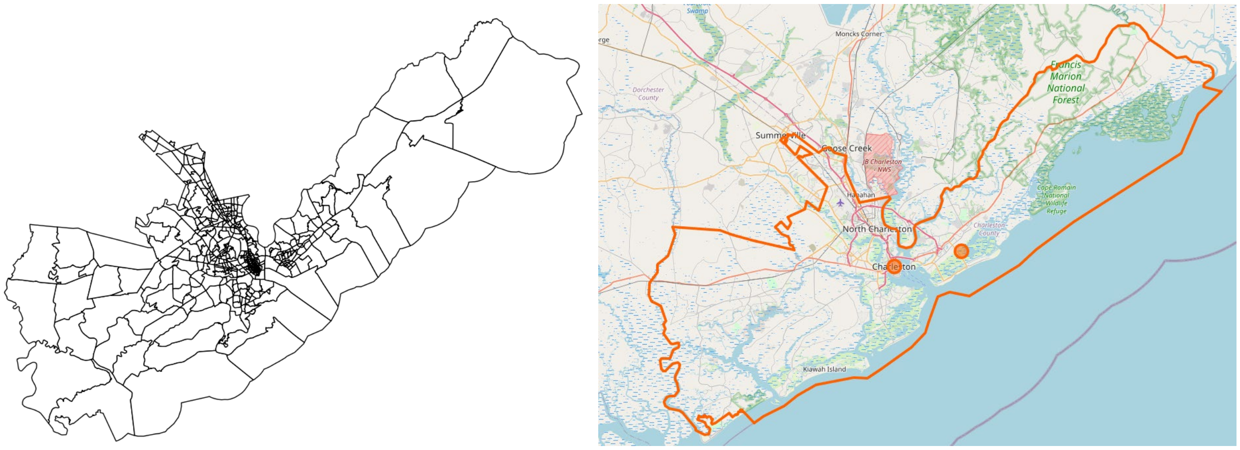

Figure 2.2 Charleston County Limit (Left); OpenStreetMap (Right).

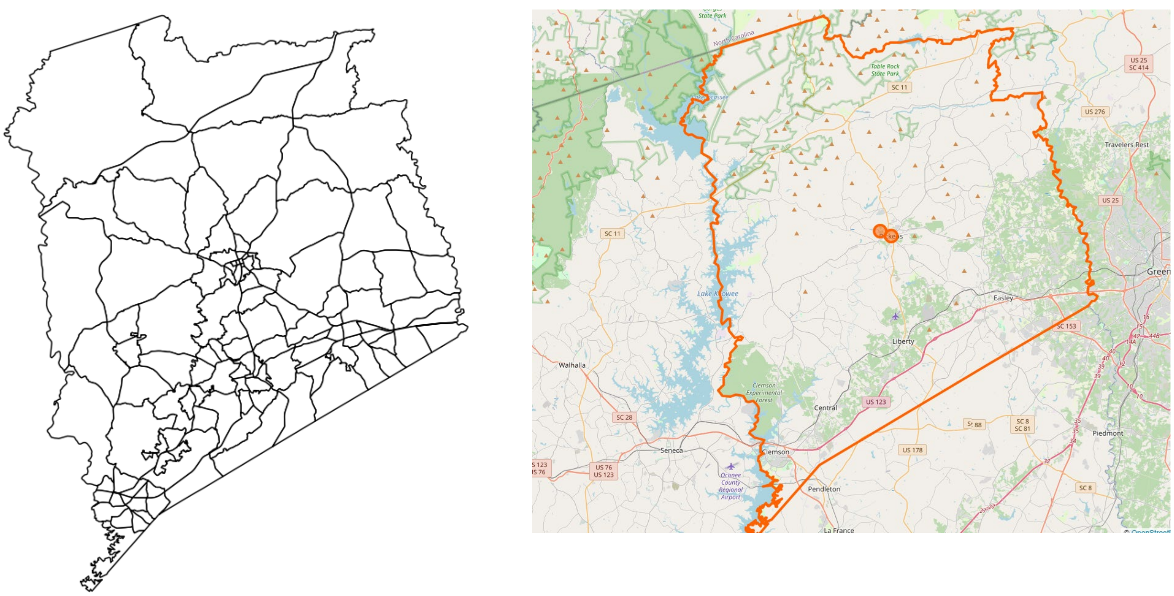

Figure 2.3 Pickens County Limit (Left); OpenStreetMap (Right).

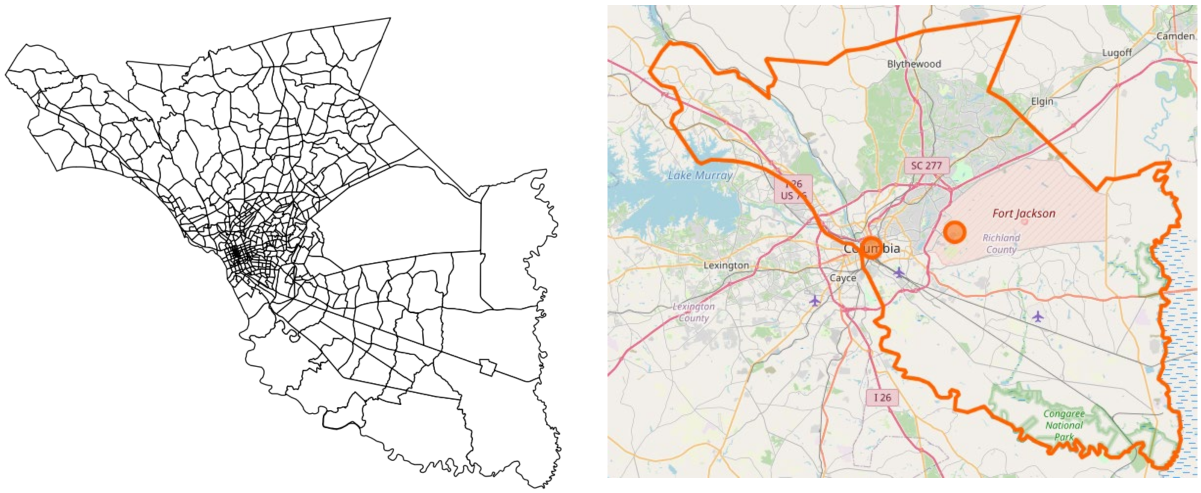

Figure 2.4 Richland County Limit (Left); OpenStreetMap (Right).

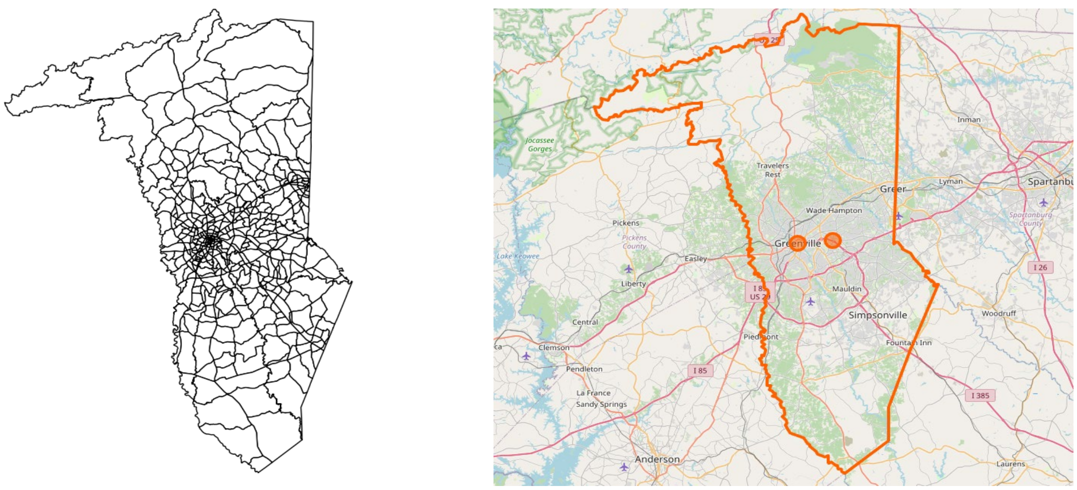

Figure 2.5 Greenville County Limit (Left); OpenStreetMap (Right).

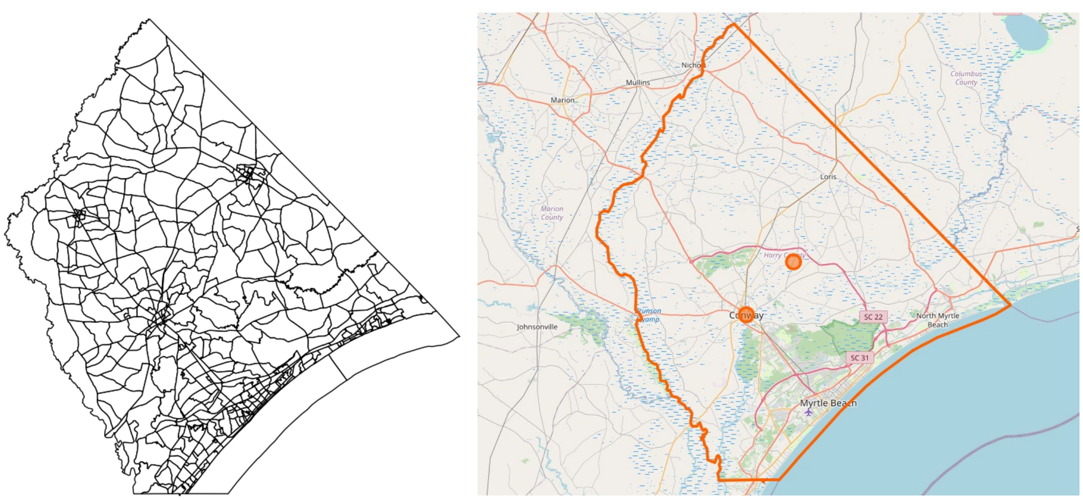

Figure 2.6 Horry County Limit (Left); OpenStreetMap (Right).

According to Liu et al. (Liu et al., 2023), retirement communities, hospitals, shopping malls, dental care facilities, senior service centers, and transportation hubs were among the most frequently visited places by vulnerable groups during the pandemic. Regrettably, these locations often faced challenges due to insufficient public transportation support, especially during city lockdowns or periods of limited transportation operations.

For each city's POIs, we acquired their addresses and coordinates from Google Maps. Furthermore, this study estimates the travel time between each origin-destination (OD) pair using Google Maps. Detailed information for each metropolitan area in South Carolina can be found in Table 2.1 to Table 2.10.

Table 2.1 POIs in the City of Clemson.

No.	Name	Address	Type
1.	Clemson Downs	500 Downs Loop, Clemson, SC 29631	Retirement community
2.	Everlan of Clemson	150 Pershing Ave, Clemson, SC 29631	Retirement community
3.	Rogers Plaza	1393 Tiger Blvd, Clemson, SC 29631	Shopping mall
4.	Carolina Oaks Dental Care of Clemson	1000 College Ave, Clemson, SC 29631	Dental care
5.	Vickery Parke Assisted Living	131 Vickery Dr, Central, SC 29630	Senior service center
6.	Amtrak Clemson Station	1105 Tiger Blvd, Clemson, SC 29631	Transportation hub

Table 2.2 Travel Time Matrix of each OD Pair in Clemson /min.

	1.	2.	3.	4.	5.	6.
1.	--	6	8	11	7	8
2.	6	--	4	8	6	6
3.	6	5	--	5	6	2
4.	10	9	5	--	8	3
5.	7	6	5	8	--	7
6.	7	5	2	2	7	--

Table 2.3 POIs in the City of Charleston.

No.	Name	Address	Type
1.	Merrill Gardens at Carolina Park	3501 Merrill Pl, Mt Pleasant, SC 29466	Retirement community
2.	Daniel Pointe Retirement Community	514 Robert Daniel Dr, Charleston, SC 29492	Retirement community
3.	Mount Pleasant Gardens	1025 Hungry Neck Blvd, Mt Pleasant, SC 29464	Retirement community
4.	East Cooper Medical Center	2000 Hospital Dr, Mt Pleasant, SC 29464	Hospital
5.	Mount Pleasant Towne Centre	1218 Belk Dr, Mt Pleasant, SC 29464	Shopping mall
6.	Beachy Dental at Mount Pleasant	1051 Johnnie Dodds Blvd, Mt Pleasant, SC 29464	Dental care
7.	Mount Pleasant Senior Center	840 Von Kolnitz Rd, Mt Pleasant, SC 29464	Senior service center
8.	North Charleston Transit Center	4565 Gaynor Ave, North Charleston, SC 29405	Transportation hub
9.	Charleston International Airport	5500 International Blvd, North Charleston, SC 29418	Transportation hub
10.	Port of Charleston	196 Concord St, Charleston, SC 29401	Transportation hub

Table 2.4 Travel Time Matrix of each OD Pair in Charleston /min.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
1.	--	19	11	13	11	14	13	26	27	25
2.	20	--	13	12	12	11	12	13	13	22
3.	12	9	--	5	4	6	5	18	18	17
4.	13	11	5	--	5	4	1	19	19	17
5.	11	11	2	6	--	6	6	19	19	18
6.	13	10	6	4	6	--	4	17	18	14
7.	13	12	6	2	6	4	--	20	20	16
8.	27	15	19	19	19	18	19	--	10	19
9.	27	16	20	20	20	19	20	10	--	22
10.	24	21	17	15	17	12	15	19	20	--

Table 2.5 POIs in the City of Columbia.

No.	Name	Address	Type
1.	Merrill Gardens at Columbia	2205 Gregg St, Columbia, SC 29207	Retirement community
2.	Still Hopes Episcopal	1 Still Hopes Dr, West Columbia, SC 29169	Retirement community
3.	Lexington Medical Center	2720 Sunset Blvd, West Columbia, SC 29169	Hospital
4.	Columbiana Centre	100 Columbiana Cir, Columbia, SC 29212	Shopping mall
5.	Total Dental Care Of South Carolina	1061 St Andrews Rd, Columbia, SC 29210	Dental care
6.	Seven Oaks Senior Citizens Center	200 Leisure Ln Columbia, SC 29210	Senior service center
7.	Amtrak Station	850 Pulaski St, Columbia, SC 29201	Transportation hub
8.	Columbia Metropolitan Airport	3250 Airport Blvd, West Columbia, SC, 29170	Transportation hub

Table 2.6 Travel Time Matrix of each OD Pair in Columbia /min.

	1.	2.	3.	4.	5.	6.	7.	8.
1.	--	13	14	19	14	18	10	19
2.	13	--	11	20	14	19	6	12
3.	15	12	--	16	10	15	13	11
4.	17	18	12	--	9	11	15	17
5.	12	13	8	8	--	5	10	13
6.	17	18	12	11	5	--	15	18
7.	10	6	12	17	11	16	--	15
8.	21	12	13	21	15	20	15	--

Table 2.7 POIs in the City of Greenville.

No.	Name	Address	Type
1.	Merrill Gardens at Greenville	90 Payne St, Greenville, SC 29601	Retirement community
2.	Holiday Haywood Estates	1180 Haywood Rd, Greenville, SC 29615	Retirement community
3.	Greenville Community Residence	158 Cavalier Dr, Greenville, SC 29607	Retirement home
4.	Medical Center Clinics	877 W Faris Rd, Greenville, SC 29605	Hospital
5.	Centro de Greenville	2-8 E Washington St, Greenville, SC 29601	Shopping mall
6.	Haywood Mall	700 Haywood Rd, Greenville, SC 29607	Shopping mall
7.	Rocky Creek Dental Care - Greenville	1322 E Washington St # D1, Greenville, SC 29607	Dental care
8.	AARP Foundation Senior Services (permanently closed)	301 University Ridge Ste. 5550, Greenville, SC 29601	Senior service center
9.	Greenlink Transit Center	100 W McBee Ave, Greenville, SC 29601	Transportation hub
10..	Greenville Downtown Airport	100 Tower Dr # 2, Greenville, SC 29607	Transportation hub

Table 2.8 Travel Time Matrix of each OD Pair in Greenville /min.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
1.	--	12	14	9	3	11	5	4	3	8
2.	11	--	12	14	9	4	10	10	10	7
3.	13	13	--	11	13	12	11	11	13	10
4.	8	14	10	--	8	13	8	6	8	10
5.	4	9	13	9	--	8	3	3	1	7
6.	12	4	11	14	10	--	8	11	11	6
7.	6	8	10	9	3	7	--	5	5	5
8.	4	11	12	8	4	10	5	--	4	9
9.	3	10	13	9	2	9	3	4	--	7
10.	11	7	10	11	8	6	5	9	9	--

Table 2.9 POIs in Myrtle Beach.

No.	Name	Address	Type
1.	Myrtle Beach Grove Senior Living	3620 Happy Woods Ct, Myrtle Beach, SC 29588.	Retirement community
2.	Oceanside Village	600 Oceanside Dr, Surfside Beach, SC 29575.	Retirement community
3.	Jensens Retirement Community, South Carolina	1700 US-17 BUS, Myrtle Beach, SC 29577.	Retirement community
4.	South Strand Medical Center	5050 US Highway 17 Bypass South, Myrtle Beach, SC 29588.	Hospital
5.	Coastal Grand Mall	2000 Coastal Grand Cir, Myrtle Beach, SC 29577.	Shopping mall
6.	Grand Strand Plaza Shopping Center	1490 S Kings Hwy, Myrtle Beach, SC 29577.	Shopping mall
7.	Aspen Dental - Myrtle Beach, SC	1209 N Retail Ct, Myrtle Beach, SC 29577.	Dental care
8.	Indigo Carolina Forest	176 Village Center Blvd, Myrtle Beach, SC 29579.	Senior service center
9.	Myrtle Beach International Airport	1100 Jetport Rd, Myrtle Beach, SC 29577.	Transportation hub
10.	Myrtle Beach Transfer Center	Myrtle Beach, SC 29577.	Transportation hub

Table 2.10 Travel Time Matrix of each OD Pair in Myrtle Beach /min.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
1.	--	17	13	7	9	14	9	19	11	14
2.	17	--	15	12	16	15	17	26	14	18
3.	14	13	--	10	6	2	7	17	3	6
4.	8	13	12	--	7	12	8	18	10	13
5.	9	17	6	7	--	6	1	12	4	7
6.	15	15	2	12	6	--	6	16	4	6
7.	10	18	7	8	1	6	--	12	6	7
8.	19	27	17	17	12	17	12	--	15	12
9.	16	18	8	13	7	8	9	18	--	11
10.	14	19	6	12	6	6	7	11	8	--

2.3 Demand Generation

During the algorithm development stage, forecasting passenger demand is challenging without real-world data support. Therefore, in this study, we generate vulnerable group traveling demand by Monte-Carlo simulations (Lampa & Samolejová, 2020; Liu et al., 2023).

Monte Carlo Simulation is a type of computational algorithm that uses repeated random sampling to obtain the likelihood of a range of results of occurring with a prescribed probability distribution (Vujic; “What Is Monte Carlo Simulation?,” 2021).

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

Eq. (1) estimates the arithmetic mean of travel requests, where x_i represents the number of travel requests during period i.

There are several probability distributions that can be used to model travel demand for the vulnerable population, including normal distributions, bimodal gamma distributions, and uniform distributions. Previous research has concluded that when demand data follows a normal distribution, the total cost is minimized (Liu et al., 2023). A demand sample generated by the Monte Carlo simulation with a normal distribution is shown in Fig. 2.7.

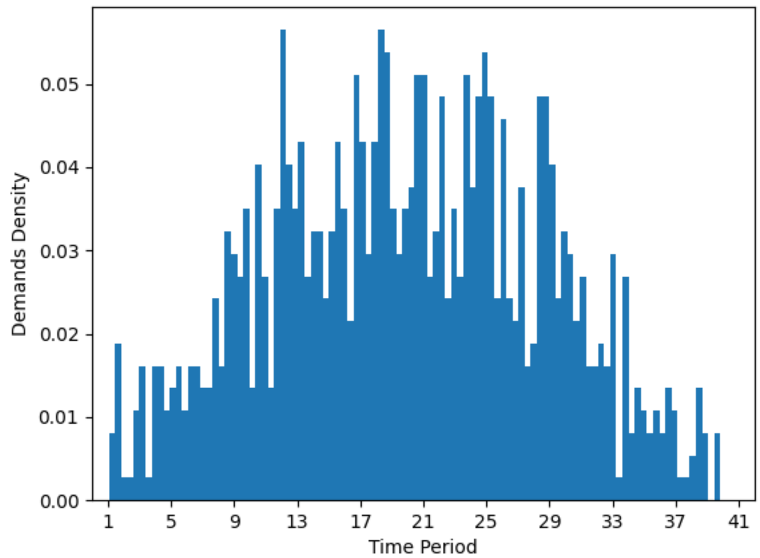

Figure 2.7 Demand Sample Generated by Monte Carlo Simulation (Normal Distribution).

The x-axis in Fig. 2.7 represents the time periods, dividing the daily operation hours into smaller intervals to enhance the accuracy of the estimation.

2.4 Coding Packages

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI ("Tkinter," 2024; "tkinter — Python interface to Tcl/Tk,"). The official reference manual of Tkinter can be found at https://docs.python.org/3/library/tkinter.html.

To start a Python GUI coding program, use the command "from tkinter import *" or "import tkinter as tk" to import all available commands from the Tkinter package.

Table 2.11 Command Options in Tkinter.

Option	Explanation
tk.Tk()	Construct a toplevel Tk widget, which is usually the main window of an application, and initialize a Tcl interpreter for this widget. Each instance has its own associated Tcl interpreter.
tk.button()	Create and manipulate 'button' action widgets.
tk.Checkbutton()	Create and manipulate 'checkbutton' boolean selection widgets.
Tk.destory()	Destroy one or more windows.
tk.Entry()	Create and manipulate 'entry' one-line text entry widgets.
tk.IntVar()	Value holder for integer variables.
tk.label()	Create and manipulate 'label' non-interactive text or image widgets.
tk.title()	Update or replace the Title Bar of the Tkinter application.
--grid()	Geometry manager that arranges widgets in a grid.
--mainloop()	Keeps the application alive, constantly listening for events such as key presses, mouse clicks, or window resizing

Additionally, this program requires several other packages, including GeoPandas/Pandas ("GeoPandas 0.14.4 — GeoPandas 0.14.4+0.g60c9773.dirty documentation,"; "pandas - Python Data Analysis Library,") and Shapely ("The Shapely User Manual — Shapely 2.0.4 documentation,") for geographic data processing, as well as Matplotlib ("Matplotlib — Visualization with Python,") and Pillow ("Pillow,") for image data processing and result visualization.

CHAPTER 3
CASE STUDIES

This chapter provides a detailed manual that introduces the tool's functions step by step, followed by two case studies from the City of Columbia, demonstrating its application under scenarios with either a limited operating budget or minimum fulfillment rate requirements.

3.1 Demonstration of the GUI Window

Step 0: Run the Python code to initialize the GUI tool.

Once the code has been run, the GUI loading and main interfaces will be displayed as shown in Fig. 3.1 and Fig. 3.2, respectively.

Note: The loading page lasts for approximately three to four seconds, after which the main interface automatically appears.

Figure 3.1 Transportation Mobility Analysis Tool GUI Loading Interface.

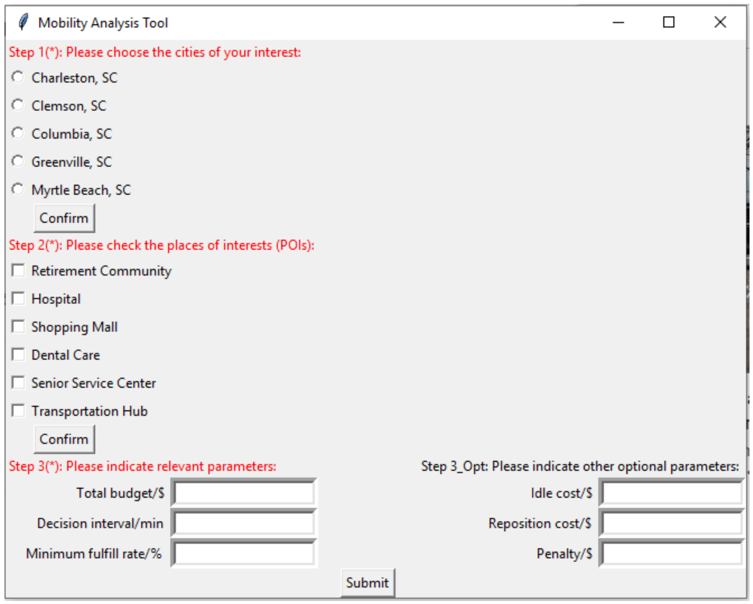

Figure 3.2 Transportation Mobility Analysis Tool GUI Main Interface.

Step 1: Select the target city from the options listed, then click the “Confirm” button.

For example, “Columbia, SC” is selected in Fig. 3.3, and then click the “Confirm” button for Step 1. The geographic location of the City of Columbia will be displayed as the response to the confirmation of city selection, see Fig. 3.4.

Note: Users must click the “Confirm” button before proceeding to the next step.

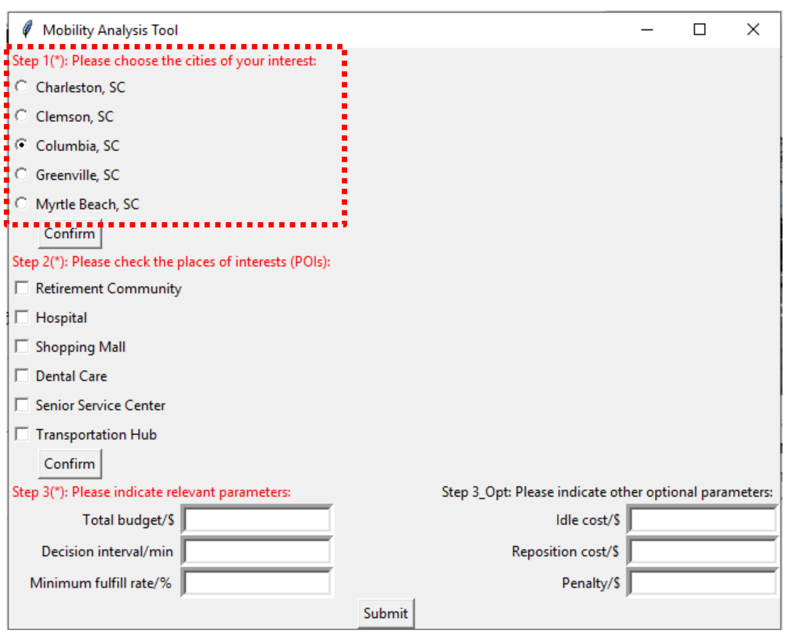

Figure 3.3 Select the Target City.

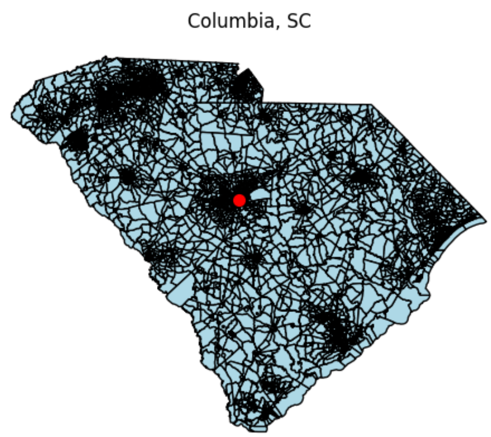

Figure 3.4 The City of Columbia.

Step 2: Select the POIs and then click the "Confirm" button.

For instance, all categories have been selected as the POIs for the City of Columbia in Fig 3.5, and a more detailed map will show the POI locations in the region of the city (Fig. 3.6).

Note: 1. The "Hospital" option is not available for Clemson. Please always leave this box unchecked when Clemson is selected in Step 1 to avoid errors.

2. Users must click the "Confirm" button before proceeding to the next step.

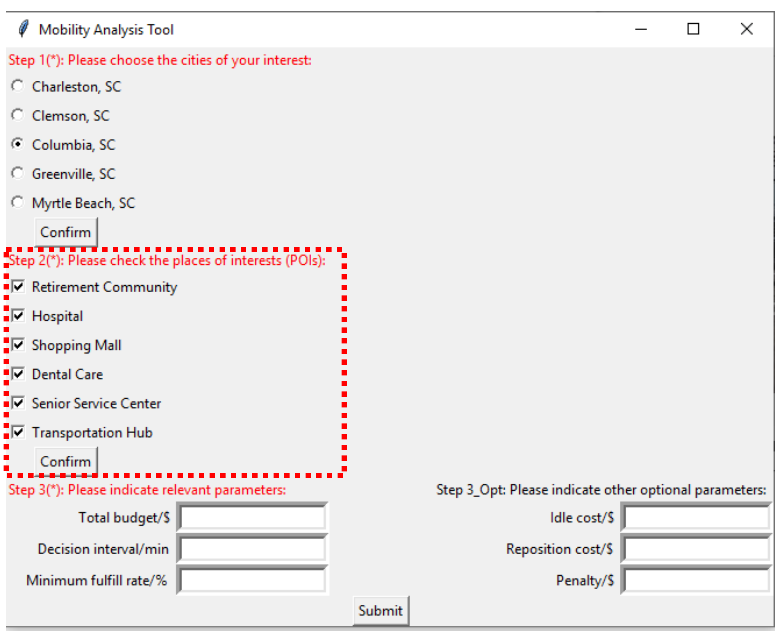

Figure 3.5 Select POIs for the target city.

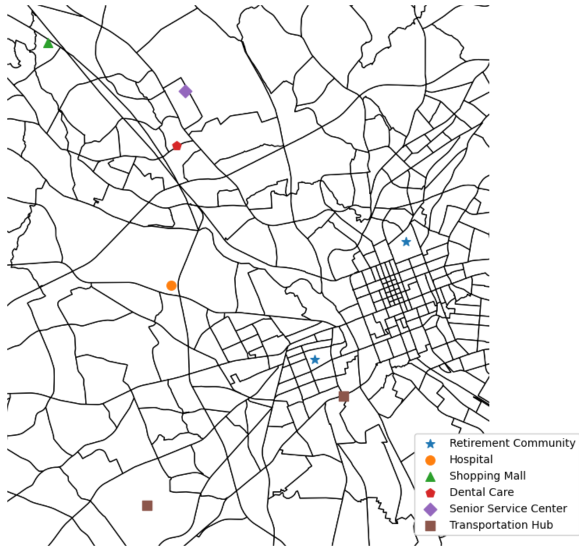

Figure 3.6 POIs in the City of Columbia.

Step 3: Input necessary parameter values such as maximum daily budget, decision intervals, and minimum fulfill rate.

A summary of other hypothetical cost categories in Step 3 (Fig. 3.7 and Fig. 3.8) is described below:

- Daily budget: Maximum daily budget in dollars.
- Decision interval: The time interval for the length of each decision period, measured in minutes. The daily operation hours are from 10:00 AM to 3:00 PM (5 hours).

  For example, with each time period set to 15 minutes, there will be a total of 20 time periods daily.
- Vehicle operating costs:
  - Vehicle idle cost (optional): Per vehicle per time period.
  - Vehicle reposition cost (optional): Per vehicle per time period.
- Minimum fulfilled rate: Per ride request.
- Penalty (Optional): Penalty if demand does not meet the minimum fulfillment rate per request.

Note: 1. Users are responsible for providing accurate values for both required and optional parameters to maintain the accuracy of the results.
2. If users do not provide specific values for optional parameters, the tool will use the default values for the entire operating period (Liu et al., 2023): idle cost = $0.50, reposition cost = $3.00, and penalty cost = $2.00.

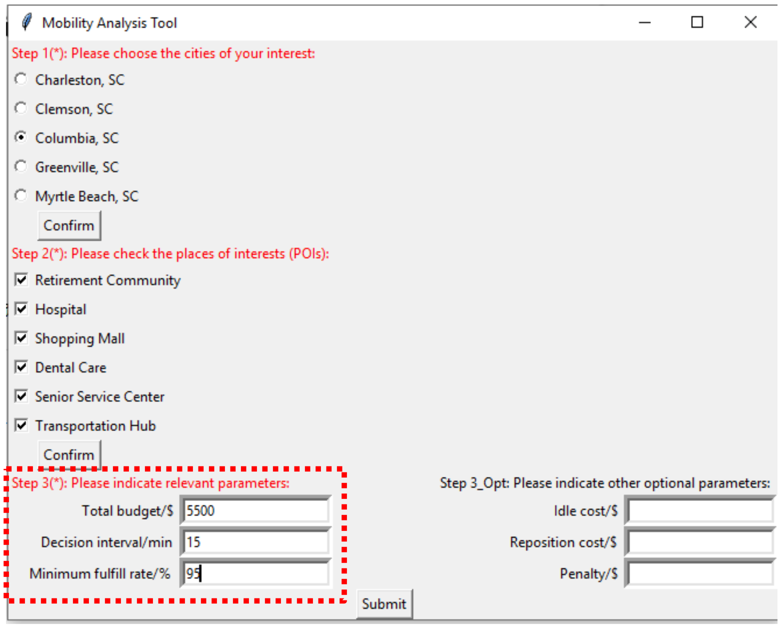

Figure 3.7 Required Parameters.

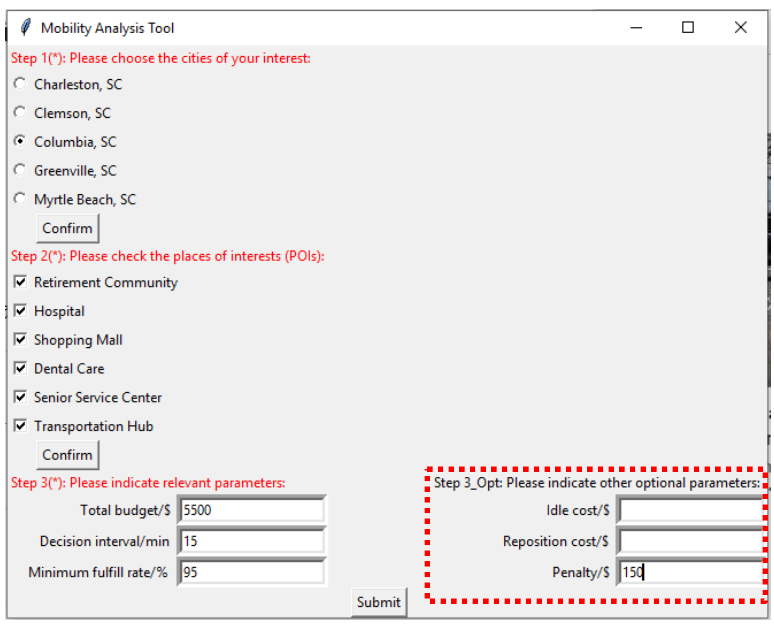

Figure 3.8 Optional Parameters.

Step 4: With all the parameters given by the users, then click "Submit" button at the bottom.

Analyzed results will be displayed after automated computation. The analyzed results from the City of Columbia example are shown in Fig. 3.9. The figure shows that the highest profit (i.e., the negative of the Total Daily Cost on the Y-axis) of $37,489 is achieved with 540 vehicles, resulting in an overall fulfillment rate of 0.891.

Note: The computing time for this process will vary, with factors such as the complexity of the calculation network (POIs selected in Step 2) and the performance capabilities of the computer influencing the timeframe. For instance, the computing time for the Columbia case is approximately 15 minutes on a computer equipped with an Intel ® Core™ i7-8700 CPU @ 3.2GHz processor and 64.0 GB RAM.

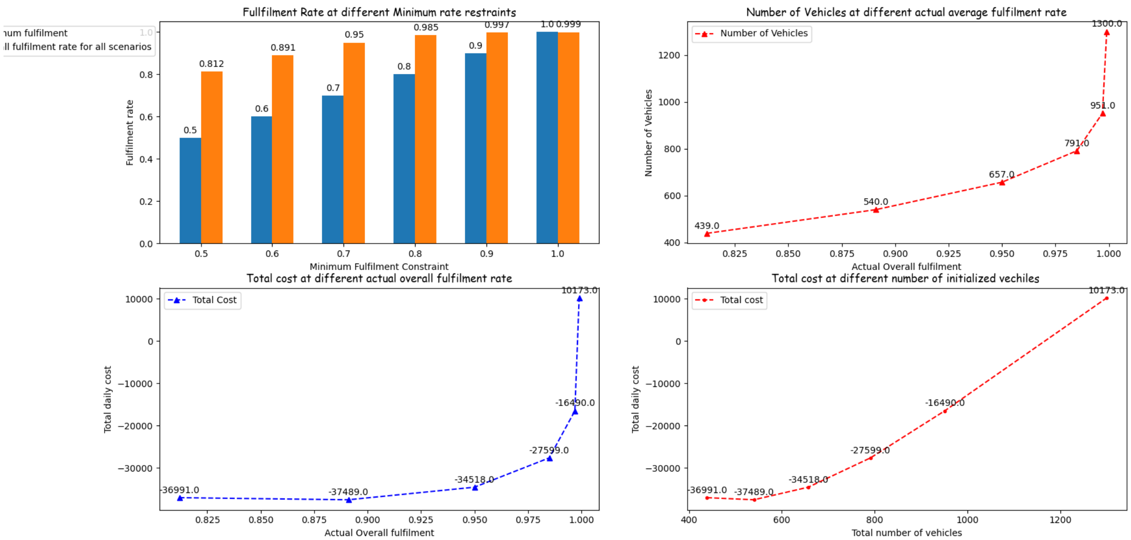

Figure 3.9 Analysis Results for Columbia, SC ($5,500).

3.2 Scenario #1: A Limited Budget

When the daily budget is limited to $2,000, and the GUI is rerun for all checked residential categories in Columbia (with a computing time of approximately 15 minutes), the corresponding analytical results are shown in Figs. 3.10 through 3.13.

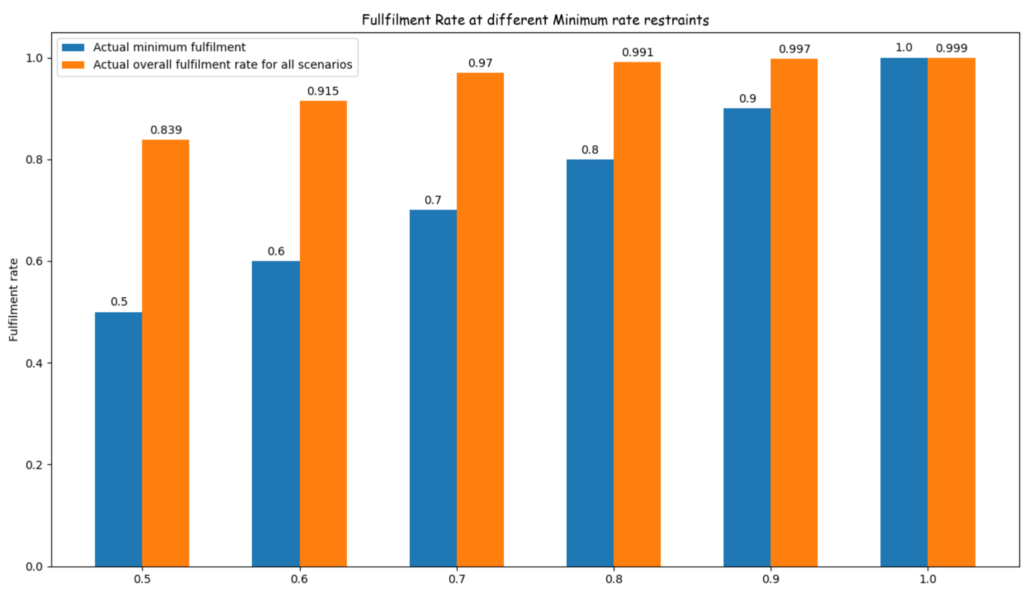

Figure 3.10 Fulfillment Rate at Different Minimum Rate Restrictions ($2,000).

Compared to the sub-figures in Fig. 3.9, the actual average fulfillment rates of ride-sharing vehicles in Fig. 3.10 increase under a limited budget, indicating that the total number of vehicles is smaller than it would be with a higher budget. Similarly, Fig. 3.11 and Fig. 3.12 show that more vehicles are operating at a higher fulfillment rate to accommodate the same passenger demands.

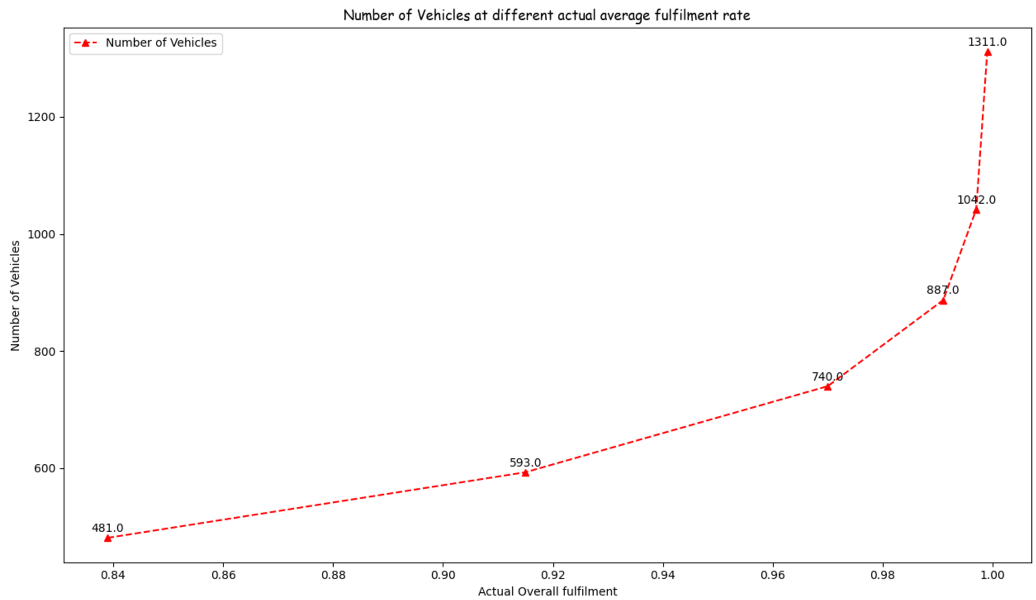

Figure 3.11 Number of Vehicles at Different Actual Average Fulfillment Rates ($2,000).

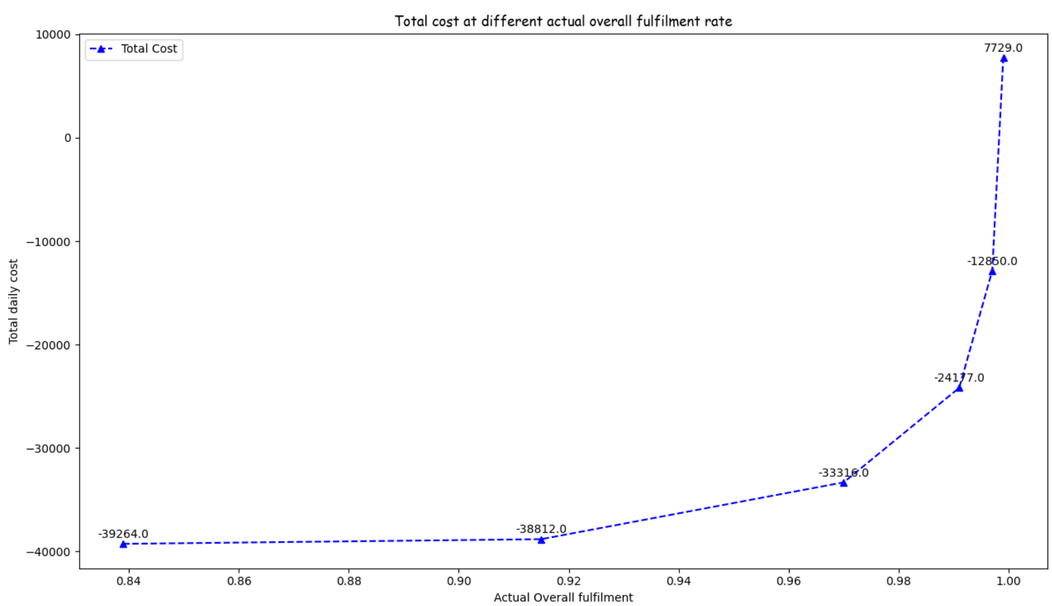

Figure 3.12 Total Cost at Different Actual Average Fulfillment Rates ($2,000).

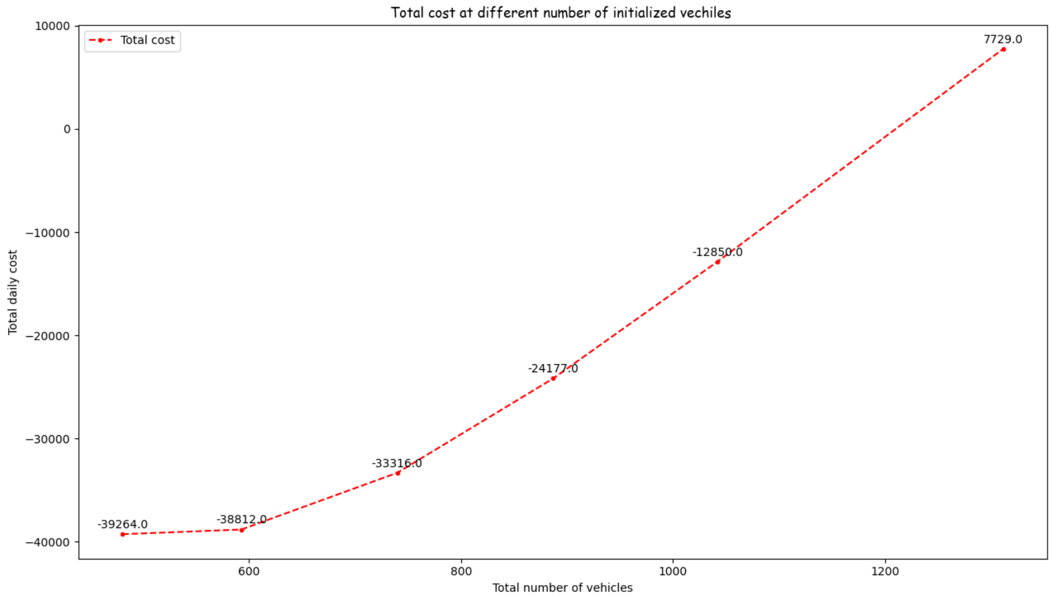

Figure 3.13 Total Cost at Different Number of Initialized Vehicles ($2,000).

Fig. 3.13 demonstrates that if the fleet size exceeds 1,200 vehicles, the operating costs will outweigh the daily budget of $2,000 and push the company into a deficit.

3.3 Scenario #2: Minimum Fulfillment Rate Requirements

From Section 3.2, we have concluded that with a lower daily budget, the actual average fulfillment rate increases. In this scenario, we further aim to improve the actual fulfillment rate for all vehicles to 90% and determine the required daily budget.

According to the previous study (Liu et al., 2023), there are two key concepts related to the "fulfillment rate": the "minimum fulfillment rate" and the "actual fulfillment rate". The "minimum fulfillment rate" refers to the lowest fulfillment level achieved by a vehicle during its operating hours, while the "actual fulfillment rate" indicates the overall fulfillment level for the vehicle throughout the entire day.

Based on the empirical results, at least one solution meets the 90% actual fulfillment rate requirement. Specifically, setting a higher minimum fulfillment rate is key to improving the actual fulfillment rate. Meanwhile, a minimum fulfillment rate of 60% is the most profitable solution for the company. To achieve this, a minimum daily budget of $350 is recommended, as shown in Figs. 3.14 through 3.17.

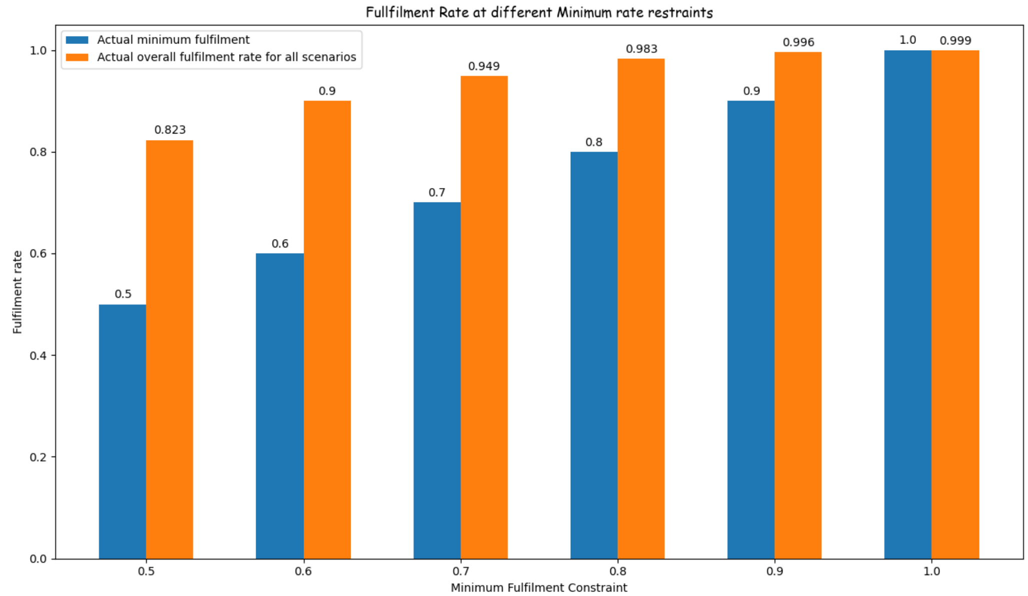

Figure 3.14 Fulfillment Rate at Different Minimum Rate Restrictions ($350).

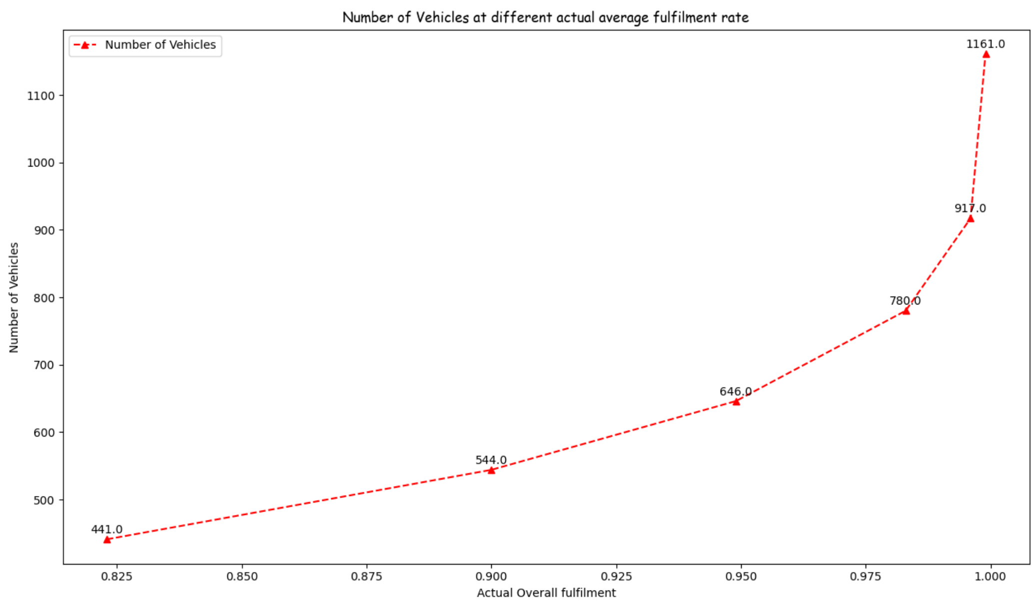

Figure 3.15 Number of Vehicles at Different Actual Average Fulfillment Rates ($350).

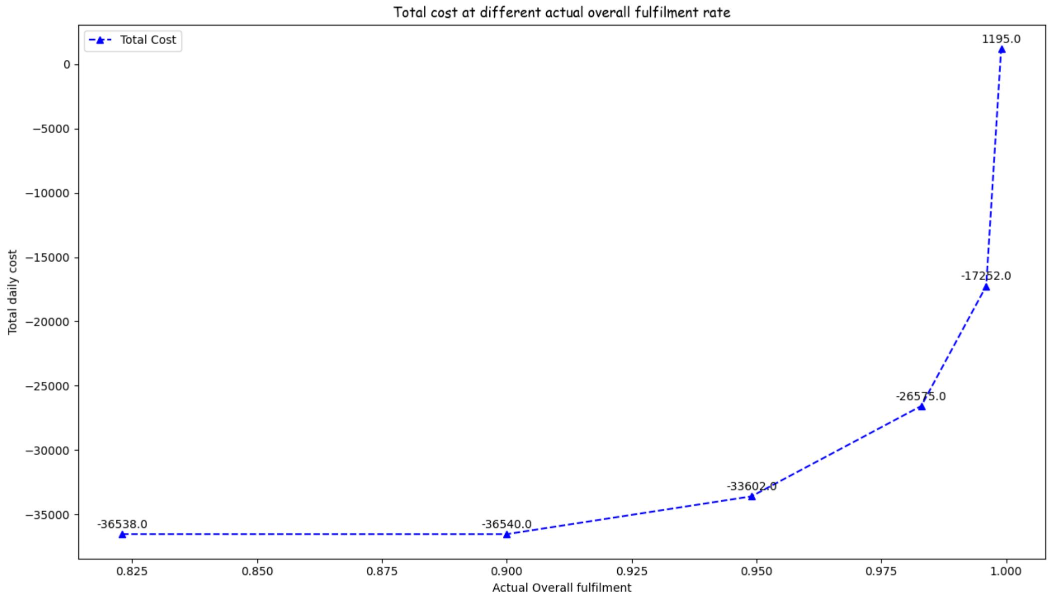

Figure 3.16 Total Cost at Different Actual Average Fulfillment Rates ($350).

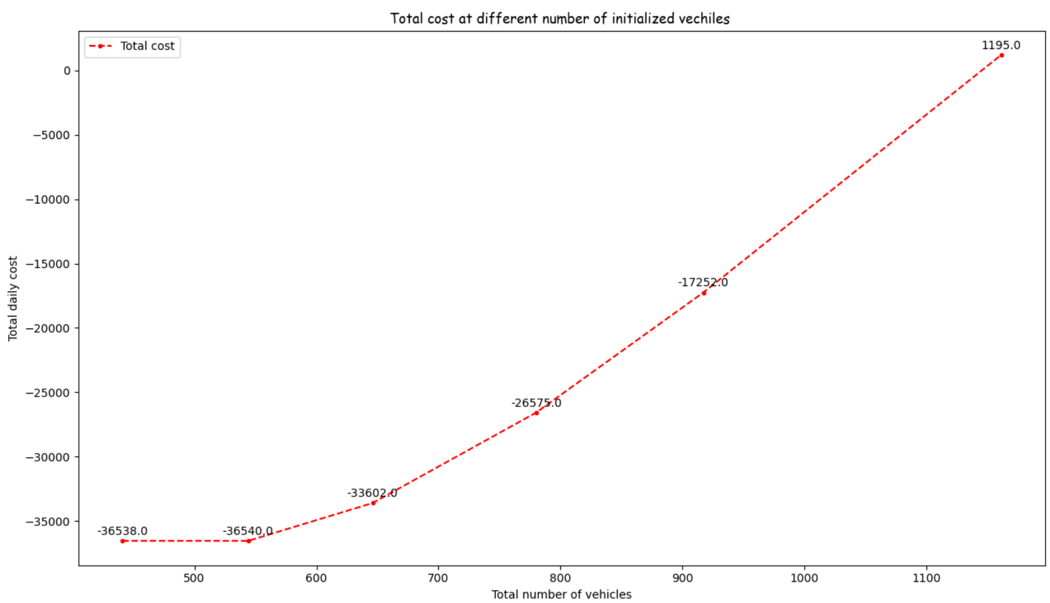

Figure 3.17 Total Cost at Different Number of Initialized Vehicles ($350).

CHAPTER 4
CONCLUSION

This project focuses on developing a user-friendly GUI application to simplify the adoption of the developed algorithm for car-sharing systems, aiding decision-making for ride-sharing companies or other transportation agencies.

We use Tkinter, a standard built-in Python package for GUI development, to streamline all core functions derived from the developed algorithms. These functions encompass interactive data input processes, such as retrieving city network data and user-defined points of interest (POI) like retirement communities, healthcare centers, grocery stores, entertainment clubs, etc. The application also handles automated data processing, calculations, decision-making, and result reporting. Once established, this tool can be applied to other locations (cities, states, or regions) to assist decision-makers with their car-sharing operation networks, provided that basic input data are supplied. Therefore, public transportation practitioners only need to focus on providing transportation services according to the scheduled plans, maintaining a high level of service while keeping within a limited budget.

In this project, a series of car-sharing system plans for vulnerable groups during pandemics is conducted using the road networks of Charleston, Clemson, Columbia, and Myrtle Beach in South Carolina. Step-by-step manual guidance is provided using the case of the City of Columbia. At each step of the streamlined procedure, the geographical location of the city, POI locations, and key parameters are inputted and visualized. This approach provides a more straightforward understanding of the system's performance, making it a more user-friendly tool for individuals who lack sufficient training. Additionally, although this project mainly focuses on the metropolitan areas in the State of South Carolina, it is worth noting that this tool can be easily extended and applied to other cities or states by following the same data processing procedures demonstrated in Chapter 2.

Overall, the developed GUI application successfully integrates complex algorithmic functions into an accessible interface, facilitating effective decision-making in car-sharing operations. This tool not only enhances operational efficiency but also ensures that services remain responsive to the needs of vulnerable populations, particularly during critical times such as pandemics. By focusing on user-friendliness and comprehensive functionality, the application stands as a significant advancement in supporting public transportation planning and management.

REFERENCES

AbdelMagid, M., AbdelRazig, Y., Smith, D., Horner, M., Choi, J., Kim, K., & Ventimiglia, B. (2023). Transportation system performance capabilities for vulnerable populations. *International Journal of Disaster Risk Reduction*, *96*, 103991. https://doi.org/10.1016/j.ijdrr.2023.103991

Conway, M. J. (1995). Python: A GUI development tool. *Interactions*, *2*(2), 23–28. https://doi.org/10.1145/205350.205354

De Paepe, L., Van Acker, V., & Witlox, F. (2023). To share or not to share, by whom is the question. Acceptability and acceptance of shared transport services by vulnerable groups. *Transport Reviews*, *43*(5), 935–969. https://doi.org/10.1080/01441647.2023.2185314

GeoPandas 0.14.4—GeoPandas 0.14.4+0.g60c9773.dirty documentation. (2024). Retrieved June 18, 2024, from https://geopandas.org/en/stable/

IBM (2021)*. What Is Monte Carlo Simulation?* https://www.ibm.com/topics/monte-carlo-simulation

Lampa, M., & Samolejová, A. (2020). FLEET OPTIMIZATION BASED ON THE MONTE CARLO ALGORITHM. *Acta Logistica*, *7*(1), 17–21. https://doi.org/10.22306/al.v7i1.151

Liu, Y., Qian, Y., Comert, G., & Begashaw, N. (2023). Mobility support for disadvantaged and disabled travelers during pandemic or similar situations. *Expert Systems with Applications*, *212*, 118786. https://doi.org/10.1016/j.eswa.2022.118786

Matherly, D., & Mobley, J. (2011). Transportation and Emergency Management Tool Kit for Communications with Vulnerable Populations: Key Research Findings. *Transportation Research Record*, *2234*(1), 62–70. https://doi.org/10.3141/2234-07

Matplotlib—Visualization with Python. (2024). Retrieved June 18, 2024, from https://matplotlib.org/

Moore, A. D. (2018). *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter*. Packt Publishing Ltd.

Pandas—Python Data Analysis Library. (2024). Retrieved June 18, 2024, from https://pandas.pydata.org/

Pillow. (2024). Pillow (PIL Fork). Retrieved June 18, 2024, from https://pillow.readthedocs.io/en/stable/index.html

Sochor, J., & Nikitas, A. (2016). Vulnerable users' perceptions of transport technologies. *Proceedings of the Institution of Civil Engineers - Urban Design and Planning*, *169*(3), 154–162. https://doi.org/10.1680/jurdp.14.00054

The Shapely User Manual—Shapely 2.0.4 documentation. (2024). Retrieved June 18, 2024, from https://shapely.readthedocs.io/en/2.0.4/manual.html

Tkinter. (2024). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Tkinter&oldid=1214582950

Tkinter—Python interface to Tcl/Tk. (2024). Python Documentation. Retrieved June 18, 2024, from https://docs.python.org/3/library/tkinter.html

Vujic, J. L. (2008). Monte Carlo Sampling Methods. *Monte Carlo*.