CALIFORNIA STATE UNIVERSITY SAN MARCOS

PROJECT SIGNATURE PAGE

PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

PROJECT TITLE: Carpool Matching by Preferences for the CSUSM Community

AUTHOR: Adriana Caetano

DATE OF SUCCESSFUL DEFENSE: 12/05/2022

THE PROJECT HAS BEEN ACCEPTED BY THE PROJECT COMMITTEE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE.

Dr. Ali Ahmadinia	Ali Ahmadinia	12/8/2022
PROJECT COMMITTEE CHAIR	SIGNATURE	DATE
Sreedevi Gutta	Sreedevi Gutta	12/8/2022
PROJECT COMMITTEE MEMBER	SIGNATURE	DATE
PROJECT COMMITTEE MEMBER	SIGNATURE	DATE

CARPOOL MATCHING BY PREFERENCES FOR THE CSUSM COMMUNITY

PROJECT
submitted in partial fulfilment
of the requirements for the degree of
Master of Science
in Computer Science

by
Adriana Caetano

Advisor
Dr. Ali Ahmadinia

Department of Computer Science
California State University – San Marcos

Fall/2022

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Professor Ali Ahmadinia, from the Computer Science Department at CSUSM, for supporting me on my topic of choice. His guidance and contribution for the development of this project is immeasurable. I would like to thank all the professors at CSUSM Computer Science Master Program, their guidance and knowledge inspired me through this journey, along with all my classmates.

I would also like to thank Belinda Garcia, Director of Parking and Commuter Services, who provided me with data from the transportation survey and the carpool program enrolment. She worked closely with me to fine tune the project on which features were important for the development of an algorithm that could satisfy campus needs. I would like to thank Cameron Stevenson from the Institutional Planning and Analysis department, who provided a real data set of class enrolment for Spring 2022 with over ten thousand unidentified students' schedules for in campus classes. Without these it would not have been possible to develop this project with realistic results.

In addition, I would like to express my gratitude to my family, for their love, comprehension, and support. They are my biggest motivation and they helped me to get till the end of this program.

Lastly, I would like to thank Adam Ramirez, who inspired me on developing this project.

Adriana Caetano

ABSTRACT

Carpooling is an alternative transportation mode where two people that are not from the same household share a vehicle to get to a destination. It reduces traffic and emissions, it helps reducing costs for users, and it can be a social time for participants. University students can benefit from carpooling, not only on transportation savings, but also in strengthening relationships with other students. However, creating long-term carpooling for university students with varied schedules and spread-out locations is not a simple task. In this project, we created an automated tool to create carpools for the CSUSM community based on user's profile and preferences. We take a greedy approach, to pair drivers with riders when searching location and schedule matches, with a heuristic function, to compute their compatibility score. We have pre-processed routes from most common locations saved in the database, which speeds up the computation time. For the target pool size of 1500 candidates, carpools can be formed in less than a minute and over 80% of the candidates are able to find ride-mates.

Keywords: Long-term carpooling with preferences matching; Greedy approach; Heuristic compatibility function

CONTENTS

List of Figures

List some figures using captions and cross-references and pointing to the figure caption style. Then create a table of figures.

LIST OF TABLES

List some tables using captions and cross-references and pointing to the table caption style. Then create a table of tables.

LIST OF ABBREVIATIONS

CSUSM: California State University – San Marcos

PRISMA: Preferred Reporting Items for Systematic Reviews and Meta-Analysis

GPS: Global Positioning System

GGA: Guided Genetic Algorithm

SB: Simulation Based Approach

CAC: Clustering Ant Colony Algorithm

ANTS: Approximated Non-deterministic Tree Search Algorithm

GA: Genetic Algorithm

k-NN: k-Nearest Neighboor Algorithm

NN: Neural Networks

ML: Machine Learning

IP: Integer Programming

SQL: Structured Query Language

Caltrans: California Department of Transportation

1 INTRODUCTION

After a long period of the global coronavirus pandemic lockdown and restrictions, students are finally back on campus. Unfortunately, the pandemic along with other external and internal factors increased the costs of living for Americans, with a soaring impact on gas prices of over $2 per gallon this past year [1]. A known measure to help keeping mobility expenses down is through shared mobility. Carpooling is a modality of shared mobility where two or more people, that are not from the same household, drive together to share the expenses. Besides the individual benefits of cost reduction and socialization with ride-mates, there is also the reduction in traffic, emissions, and parking demand [2]. This is a good socio-economic and environmentally friendly alternative of transportation mode that can be adopted by students to get to campus.

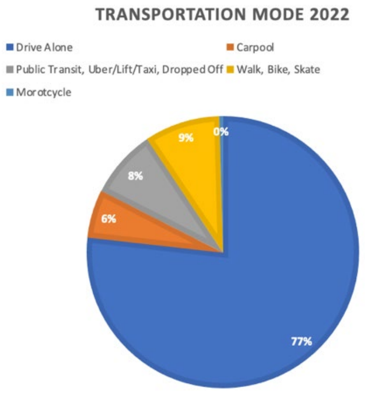


Figure 2: Student's Transportation Mode in 2022
74% of the students drive alone, with public transport as second most used transportation mode

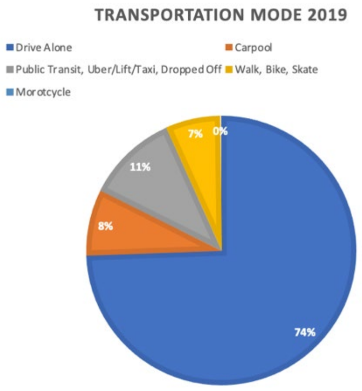
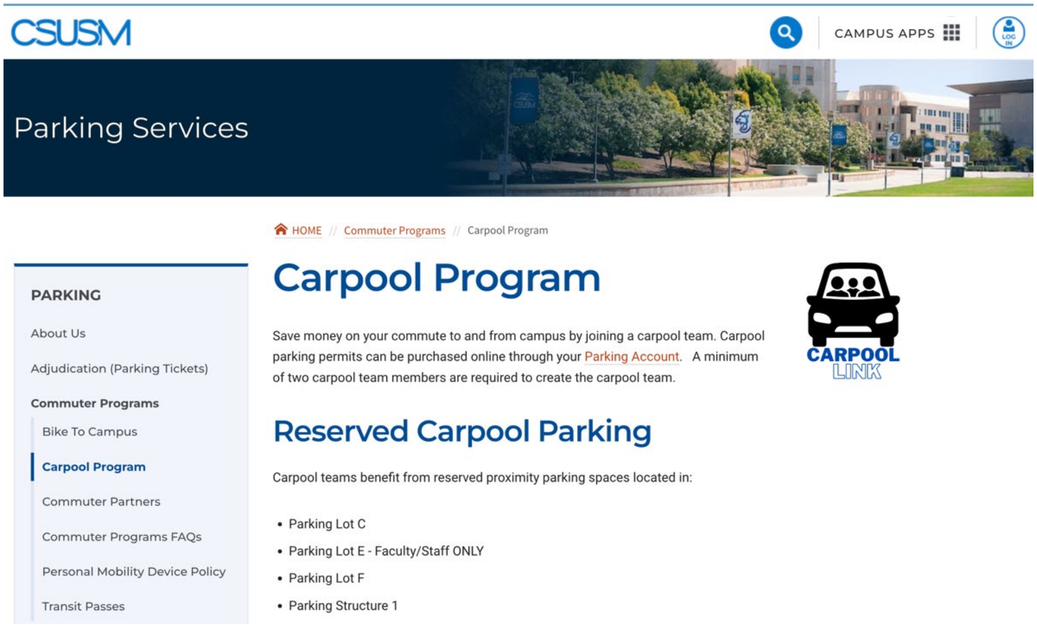


Figure 1: Student's Transportation Mode in 2022
77% of the students drive alone, with walk/bike/skate as second most used transportation mode

California State University San Marcos, CSUSM, conducts a bi-yearly Transportation Survey as part of the Sustainability Master Plan [3]. Figure 1 shows the last survey administered during Spring 2022. From a total of 1309 student respondents, 76.8% drive alone to school, and 5.7% carpool to school. Comparing to the previous survey from 2019, figure 2, which interviewed 1066 students, the answers were 74.4%, and 7.9%, respectively. An increase on driving alone and a decrease on carpooling of about two percent each. This negative change could be an effect of the post-lockdown of the global coronavirus (Covid-19) pandemic. Spring/22, when the survey was conducted, was the first semester where students were back in campus for classes. Mobility was an area very affected by the pandemic, and the concern of being surrounded by other people would explain the increase on individual transportation mode and a decrease on shared mobility. Yet, the population are learning how to live with the virus. Sharing vehicles can be safe if we form small groups that would stay together through periods of time, as long as participants agree on following safety guidelines from the authorities.



CSUSM

CAMPUS APPS

Parking Services

HOME // Commuter Programs // Carpool Program

PARKING

- About Us
- Adjudication (Parking Tickets)
- Commuter Programs
  - Bike To Campus
  - Carpool Program
  - Commuter Partners
  - Commuter Programs FAQs
  - Personal Mobility Device Policy
  - Transit Passes

Carpool Program

Save money on your commute to and from campus by joining a carpool team. Carpool parking permits can be purchased online through your Parking Account. A minimum of two carpool team members are required to create the carpool team.

CARPOOL LINK

Reserved Carpool Parking

Carpool teams benefit from reserved proximity parking spaces located in:

- Parking Lot C
- Parking Lot E - Faculty/Staff ONLY
- Parking Lot F
- Parking Structure 1

Figure 3: CSUSM Carpool Program Webpage
The Carpool link leads to a form for students to get help finding carpool partners

In order to promote alternative transportation modes, the university has a Commuter Programs page with different ways to get to school, and it offers a Carpool Program with preferred parking spaces for carpools, as shown on figure 3. There is a carpool link where they can fill up a form with personal information and schedule to get help finding carpool partners. It is embedded in the Commuter Programs section, under the Parking Services of the university website [4].

1.1 The Problem

Although the university has a mobile application where students have access to a variety of tools like checking their courses and athletic events, reserving study spaces, shopping at the university store, and so on, there is no access to the carpool program through the app. Therefore, if a student needs help on finding a carpool partner, they must access it through the university website. The access to the link is not intuitive, and not easy to find. The explicit explanation on 'get help finding a carpool partner' is at the bottom of the page, and the student needs to scroll down to read it.

After a student submit a form through the link, the process of filtering and matching students is manual, so it is not as efficient as it could be. Since this service was created on Fall/2019, there was 148 requests from students to get help to find a partner. This is the total number over a span of 7 semesters. As pointed in the work "Ridesharing in North America", when the carpool program has low enrolment, there is a low chance of matching, so the program tends to fail [5]. During almost two years, most students were having classes online and interaction among students were very limited. Now that they are back in campus, they need an efficient tool to facilitate making connections and finding partners for carpooling. On the same Transportation Survey, students who drive alone were asked what would encourage them on adopting an alternative mode of transport, and 18.7% from 2022 and 22.1% from 2019 answered they need help on finding carpool partners. This is a great opportunity to enhance the adoption of carpooling as a transportation mode to campus. During Spring 2022, CSUSM had 14,311 enrolled students, with over ten thousand of them with in person classes. If we consider that roughly 75% of them drive alone to school and that we could engage almost 20% to carpool that are willing to carpool but need help on finding carpool partners, we could have at least 1500 students carpooling to and from campus right away.

1.2 PROPOSED SOLUTION

The aim of this project is to form long-term carpool for students, promoting a safe environment for shared mobility. Through this point-of-view, carpool for university students is similar to the vanpooling problem, since both are long-term carpools with a common destination, and a capacity constraint. The main difference is the varied weekly schedule of the participants compared to the uniform daily schedule or workers from the vanpool. All the research papers reviewed on this study treat university carpools as dynamic ridesharing, with no clear intention on forming long-term carpools.

Our solution takes a greedy approach to pair drivers with riders when searching location and schedule matches with a heuristic function to compute the preferences using a compatibility score. From the driver's origin location, we search for candidates, and while the seats are not occupied, the search for candidates continue throughout the route of the driver to school. The goal is to create long-term carpools with the highest compatibility score among participants, increasing the probability of the carpool formation to last the whole semester.

To speed up the computation time, routes from student's common zip code locations are pre-processed and stored in the database. With this approach, we can create carpools in less than a minute for the target pool of 1500 candidates when the route from an origin location is found in the database. Otherwise, when a zip code origin is not found on the database, we need to create the path, then this new route is saved into the database for future reference.

2 LITERATURE REVIEW

There is a vast literature about carpooling with different focuses. In this chapter we review the shared-mobility problem (examples include carpooling, ridesharing, vanpooling, dial-a-ride, share-a-ride, and shared-taxi problems) through the Computer Science and the Social Sciences perspectives. First, we briefly summarize the carpool history in the United States. Then, we focus on the human factor of the carpool to find in the literature what are their social-demographic characteristics, motivations, and perceptions. Lastly, we examine what solution approaches and techniques researchers have used to optimize computing time, resources, and overall efficacy of the program.

2.1 CARPOOL HISTORY

According to Chan et al., the United States government introduced carpooling during the World War II to save resources for the war. The authors divide the history into five distinct phases to explain how the habit of people sharing a ride when they have similar schedules and routes fluctuate over the years reflecting the economy, change of focus, and the evolution of the technology available. During the 1970's decade, carpooling reached the pick of adoption for 20.4% American commuters. Since then, with lower fuel prices and stable economy, it declined to a steady 10% since the beginning of 2000's [5]. Recently, with new technologies such as internet, smart phones, GPS, and mobile applications, shared mobility is back on focus, with a positive trend over the past years [6].

2.2 HUMAN FACTOR: SOCIAL SCIENCES PAPERS

Carpooling benefits has been extensively studied. According to Shaheen, there are three main actors that benefits with carpooling. The society in general, with less volume of cars circulating there is a reduction on traffic, emissions, and fuel consumption; the institutions, which can save investments on the parking structures; and the individual, who gain accessibility to places not always served by public transit, benefits on mobility costs savings, and on convenience [2]. On the extensive survey on carpooling practices conducted by Aguiléra, almost one hundred research papers from the past decade were studied. The author concludes that psychological factors are more important than other variants. Trust, convenience, discomfort, difficulty on finding matches, and perceived security are some of the barriers that the system needs to overcome. She argues that environmental benefits are difficult to measure since many carpool passengers could be previous users of public transportation, so they would not necessarily be contributing on the reduction of number of cars circulating on roads [7]. Olsson has a similar study and points out that young adults are more comfortable on sharing rides and concludes that social-demographic factors are not determinant on carpooling adoption. Situational factors, such as gas price increase, are more relevant [8].

On an analysis of tweets from users of two distinct cities in a spam of 50 days, it was possible to collect user's top three most visited locations with time stamps, their profiles, and who they interact with. This data was used to simulate a ridesharing program to measure carpooling enjoyability, followed by a survey submitted to the same users that had their data collected to answer some questions about their preferences. The results of the survey are promising, showing that almost 40% would be interested in enjoyable carpool solutions. When asked about their primary motivation on carpooling, not surprisingly, the top choice with 41% is because shared mobility saves them money, but 24% chooses interesting people. This is a promising avenue for exploration and a relevant feature that must be considered on ridesharing tools [9].

On an analysis of a carpool app usage data and interviews, a research paper presents an overview of practical and psychological barriers of using the carpooling app. Some technical improvements are suggested, and the importance of the human contact in the process to make the carpool happen is highlighted. The psychosocial barriers can be lessened with some reassurance of reliability on the ride and a set of guidelines on waiting time, max detour, and riders' behavior [10].

Some universities have conducted surveys to understand students' needs on a carpooling service. To support students for the three universities of Tessaloniki city where 85% of the university population are potential users of the service, the most important features mentioned are a trip cost calculator, and that the carpool matching should be specific for the university community [11]. With focus on electric vehicles shared mobility, students from universities in Spain and Belgium are not willing to pay for e-cars to commute, but they have a positive attitude towards carpooling and sharing electric cars [12]. For an American university in New York, the 85% of the students who drive to school do it because it is convenient. The author points out that the students are influenced by the 'car culture', where driving a car is a symbol ‘independence and status. Half of the students are interested in participating in carpooling. The most important ride-mate characteristic is smoker/non-smoker with 57% of votes, followed by ‘none’ with 24%, and gender accounts for 11% of votes [13].

During Fall/2022, CSUSM Parking Services included preferences on the carpool form. Nine students submitted carpool requests under the new format, and four students did not select any characteristics for their ride-mates. The most chosen characteristic with three votes is smoker/non-smoker, followed by CSUSM status (undergraduate, graduate, staff, or faculty) with two votes, and gender with one. Comparing with the student profile of Spring/22, 62.4% of students identify themselves as a female, against 55.5% on our sample; and 89.5% is an undergraduate student, against 77.8% [14]. This is a very small population to represent the university community, but it can guide us on what is relevant for our audience.

2.3 Solution Approaches: Mathematics and Computer Science Papers

There is a vast collection of studies on the shared mobility problem, with an increasing number of publications over the past decade. To narrow down the search, we used specific problems related to the shared mobility. As mentioned by many of the authors, this is a NP-hard problem, so there are different solution approaches with different objectives for different problem variations.

2.3.1 Types of Problems

Shared mobility problems have many variants and can be categorized into cost sharing or profit, with many different objectives such as maximize number of users, minimize total distance travelled, and so on. On this project our focus is cost sharing and social factors. Here are the keywords used in the search with a similar scope of this project problem:

- **Carpooling problem:** driver and rider have common origin and destination.
- **Ridesharing problem:** a match of driver and rider with similar routes and schedules.
- **Vanpooling problem:** a carpooling using vans.

2.3.2 Surveys and Reviews

On an analysis of over one hundred published papers on different variants of the shared mobility problem over the past decade, the authors offer a review and classification of the studies based on the PRISMA method, describing current trends and analysis. They observe that time-windows and multi-objective problems are trendy topics, and, for most of the researchers, minimizing the total cost is their primary goal for all problem types. Other recurrent objectives on many of different problems are minimizing total distance, minimizing travel time, and maximizing number of matched rides. Heuristic or Metaheuristic are the most common solutions, followed by exact approaches, a hybrid solution using a combination of exact and heuristic algorithms, and other solution approaches [6]. Mourad et al. provide a 20-year range survey on shared mobility systems for people, goods, and people and goods. The study presents some case studies using optimization, simulation, and data-analysis methods. Exact and Heuristic approaches are used with different objective functions for travel distance, travel time, number of participants, travel cost, emissions, reliability, occupancy, and number of vehicles, while respecting routing, time, capacity, cost, and synchronization constraints. The authors suggest development on research for dynamic ridesharing systems, autonomous mobility services, and passenger and goods shared mobility. They identify a lack of qualitative research from the user point-of-view [15].

2.3.3 Different Approaches

For this project we reviewed 25 published works with different solutions for the shared mobility problem variants. A summary table is on the Appendix section.

Some studies created new solutions, like the bee colony optimization algorithm (BCO) for solving the long-term carpool. The authors did experiments on randomly distributed sets with different size instances to compare the performance with the GGA, Ants, SB and CAC algorithms, the BCO achieves better solutions in less time [16]. Different research created a combination of decomposition and network flow algorithms. Tests on a limited synthetic randomly generated dataset shows 65% reduction of service time, and 56% reduction of number of vehicles used [17].

Other studies implemented variations of local search algorithms with heuristic approaches. Looking for a solution for the carpool with three people or more, a study found optimal solution using Linear Programming and Enumerated Procedure for small instances of passengers/ candidates within 6 hours.

Heuristics solutions are better for larger instances with faster solution. Compared to Simulated Annealing, Tabu search has a greater execution time for larger problem sizes, but with a shorter carpooling route cost, finding the optimal solution more often [18]. For the maximum carpool matching problem using local search with similarity function, the authors investigate approximation algorithms and the impact of capacity constraint with mathematical proof for each variant. They conclude that a matching problem is an instance of Knapsack problem and can be solved using dynamic programming in polynomial time [19].

A comparison of different vehicle routing algorithms found that the best performance in the simulator environment was the hybrid approach, using Dijkstra for the initial shortest path and GA to adapt to new events [20]. Another study used a meta-heuristic approach using many variants of discrete differential evolution algorithms with binary decision variables to find a solution using bids and concluded that differential evolution with neighborhood search outperforms all other variants [21].

When user preference is part of the problem, researchers used different methods. A comparison of the most common ML algorithms on a synthetic dataset with 108 users with gender, age, smoker, distance, response, music attributes divided into four datasets with different sizes shows that the kNN produced the best efficiency results for all datasets using Holm Test [22]. A carpool application created for the employees and students at the Hashemite University, in Jordan, also uses k-NN to find potential matches based on the user preferences [23]. Genetic Algorithm with a customized fitness function was used on a real dataset of 119 users with four different destinations, with objective function with fairness, and costs for distance, time, and preferences (number of users, smoker, blacklist) and concludes that sparse and clustered initial populations achieve similar results in finding a global solution [24]. An exact algorithm was used on a network-flow model using Lagrangian Relaxation to find close-optimal solution. Tested on different size instances of synthetic generated data with gender and smoker characteristics, the most time-consuming instance was solved in less than 1 hour. Feasible for long-term carpools using the same route with the goal of forming long-term groups to increase carpooling adoption [25]. A Guided Monte Carlo method with matching matrix and objective function was used to solve carpooling for universities, and the authors describe high quality of service with low computational time [26]. A social compatibility score was used on an optimization model with objective functions to maximize social compatibility or saved distance, and a heuristic solution using both. The heuristic solution is better suited for larger sets, having a less than 20% lower performance compared to IP approach. Splitting routes allows for increase in the number of matched riders/drivers ratio, it's beneficial when the number of drivers is low, however the number of stops has a positive linear correlation with computational time. The social score is used a parameter rather than a constraint, allowing for more matches even if not all preferences are met. Maximizing social score has a small impact on distance saved, but maximizing distance saved has a big impact on social score [27].

Two different papers were covering the same study on matching algorithms. They developed four different matching algorithms with focus on maximizing matched users, users shared distance, time to favor social integration, and physical proximity of users' trips. The four algorithms were tested on two real datasets from a city in the U.S.A., and a city in Italy. The most recent published research concludes that each correspondent algorithm performed better on its own metric. The Saved Distance Matching algorithm has the best performance overall [28]. On the first to be published, results show that a small increase in delay time drastically increases the pool options to incorporate social factors to the carpool while still increasing cost savings of 15% to 20% [29].

One main difference between all research and this project is the quality of the data. Our dataset has almost ten thousand real students' data with their locations and schedules for the Spring/22 semester. The biggest dataset used with real data has less than 2000 users from two different locations and were used on more than one study [28] [29]. The only long-term carpool with preferences match that uses real data has a dataset of 119 users [24]. Another main difference is the approach of creating a long-term carpool for users with different schedules. Not a single paper mentioned this challenge on their constraints. An approach of saving users on the nodes of routes for a faster lookup on an inverted data structure is used by Schreieck [30]. Inspired by their approach, we create a list of zip codes extracted from the route's nodes and save it into the database. We use this list to search for candidates on the driver's path. Our dataset does not have the exact location of students, but their residential zip codes (for individual data security compliance). By using the 'zips on route' list we verify if a user is on the route by checking on a list that is a dozen to hundreds of times smaller than if we were checking on the list of route's nodes, which also speeds up the processing time. The advantage of not computing the routes for each driver saves an average of 2 minutes per route (longer distances take longer time). A 1500 set of users with pre-processed routes can form carpools with high social compatibility in less than one minute.

3 Data Analysis

3.1 Data Collection

To create the model, we used the class enrolment dataset from Spring/22. The dataset consists of all the in-person classes with classes for all majors and academic levels offered during the mentioned semester. Each class has its schedule information that consists of the term (all from Spring/22), specific information on the class itself such as subject, catalog, and section, the meeting pattern of which weekdays they occur, the start and end time, and which student are enrolled in that class. The student information is unidentified. It has a number that is not the student number from the school system, and general information of their profile, such as gender, age, academic level, major, and zip code of residency. The dataset has 33417 entries, which corresponds to all seats of all in-person classes offered this semester. It consists of 306 classes offered with 86 different sections and 12 different meeting patterns to 10,292 students.

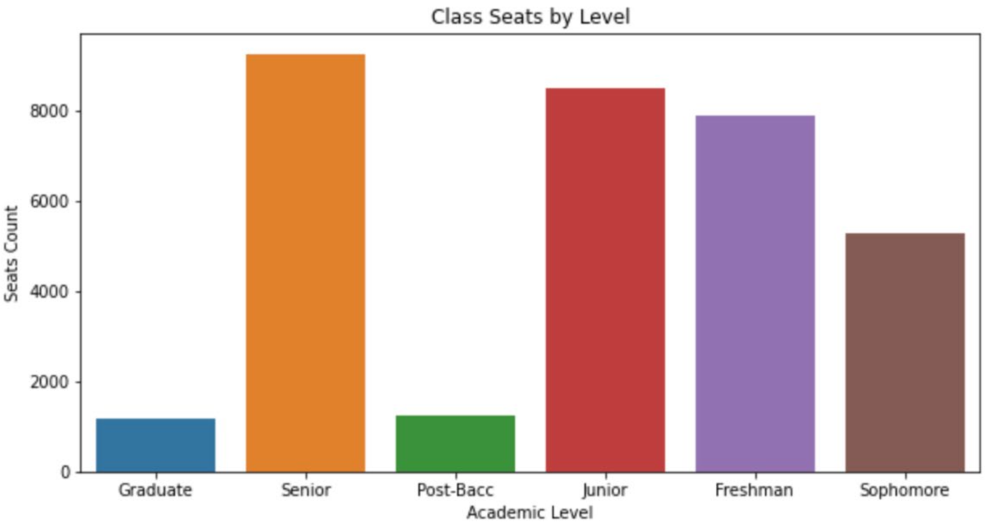


Figure 4: Bar Char of Seat Count per Academic Level
The college that offers more seats is the CHABSS, with almost the same number of seats of CEHHS and COBA combined.

Analyzing the seats per academic level, the number of freshman and specially sophomore students is less than juniors and seniors, as we can see on figure 4. We may assume that the lockdown and the transfer of all classes to on-line environment had an impact on new students' enrolment for the university. We created a new column for college and assigned it based on the student's major. Students with undeclared major were not assigned to any college, so we created a 'undeclared' college. Figure 5 shows the number of class seats per college. The CHABSS – College of Humanities, Arts, Behavioral and Social Sciences offers more seats than other colleges, followed by CSTEM – College of Sciences Technology, Engineering, and Mathematics, CEHHS – College of Education and Human Health Services, and COBA – College of Business Administration.

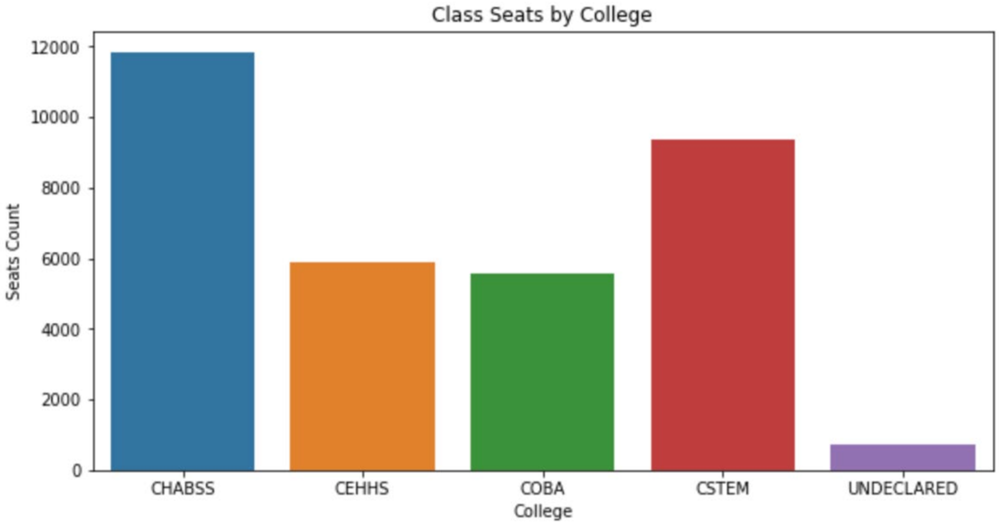


Figure 5: Bar Char of Set Count by Academic Level
Graduate and Post-Baccalaureate together have less than half of the seats' count than the lowest count of the undergraduate level, which is Sophomore. We can see the decline on the number of seats over the past two years, most likely, due to the change to online classes during the covid lockdown.

3.1.1 Data Cleaning

For the purpose of our project, we need to identify the student schedule for each day of the week with arrival and departure, where they come from, and set their preferences based on their profile. We don't need to know the specific classes they attend or which specific major they have. So, we created another dataset using the classes dataset as source. We compiled their arrival and departure time for each weekday. Age, gender, academic level, and postal code of residency were kept. The postal code had some missing values, and we had to discard those students.

3.2 STUDENT PROFILE

Out of 10,102 students, almost six thousand are female, and only nine identify as non-declared. The pie chart on figure 6 show the percentage of female, male, and non-binary students during the Spring/22 semester.

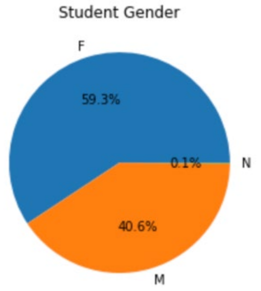


Figure 6: Gender Pie Chart
Almost 60% of the students are female

Figure 7 shows a bar plot of students count by age with a left-skewed distribution, with higher concentration of students between 18 and 25 years old.

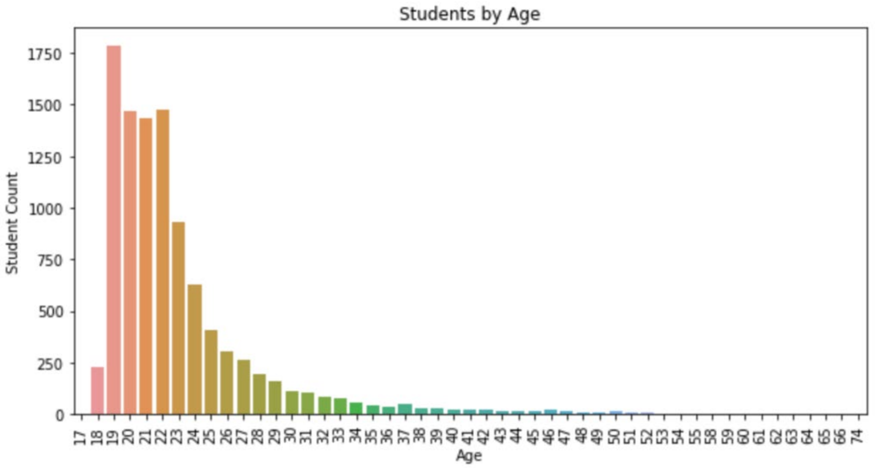


Figure 7: Student Age Distribution
The youngest student is 17 years old and the oldest is 74. The mean age is 22 and 75% of the students are 24 years old or younger

It's interesting to notice that, although female students are almost 60% of the students, they are not evenly distributed among colleges. While women count for 71% of the students enrolled in health and humanities, 59% of the students enrolled on Business and 58% of the students on Stem are man, see figure 8.

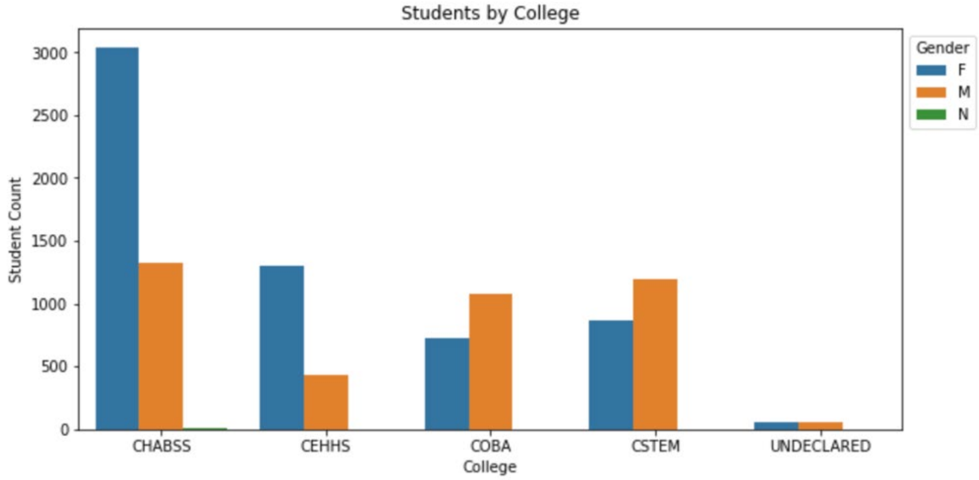


Figure 8: Student Gender Distribution by College

CHABSS has a ratio of 7 women to 3 men, 3 females to 1 male on CEHHS, but more than 3 men per 2 women for COBA and CSTEM. Undeclared majors are evenly distributed among genders.

As you can see on figure 9, some students had zip codes from out of the state of California, and even out of the country. We assume that those students moved into campus and are not commuting to go to and from classes.

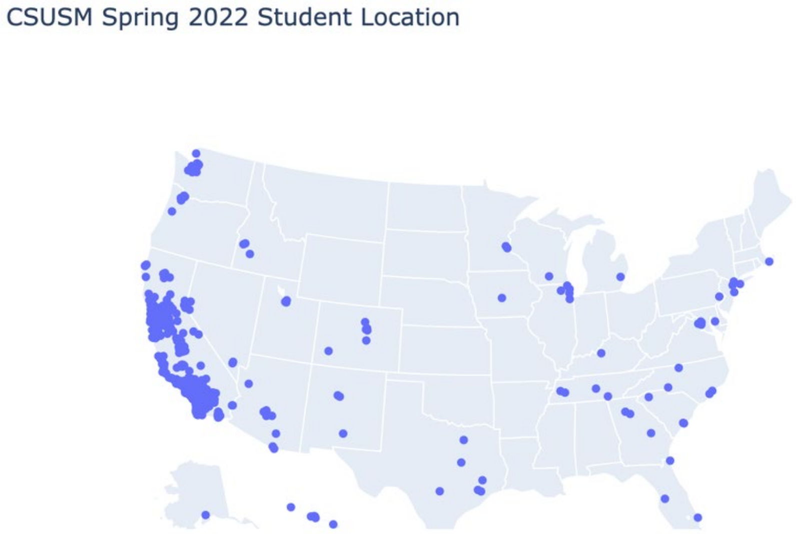


Figure 9: USA Map with Student Zip Code Location
Many students are from within the state of CA, but there are zip codes from many other states.

After discarding those students, our data set has 9953 students with 763 different California postal codes. We used the Python pgeocode library and US ZCTA file downloaded from the US Census website with CA zip boundaries and other geographical information to retrieve geolocation of the postal codes, finding 761 locations. Now we can determine the distribution of students per county, shown on figure 10 [30].

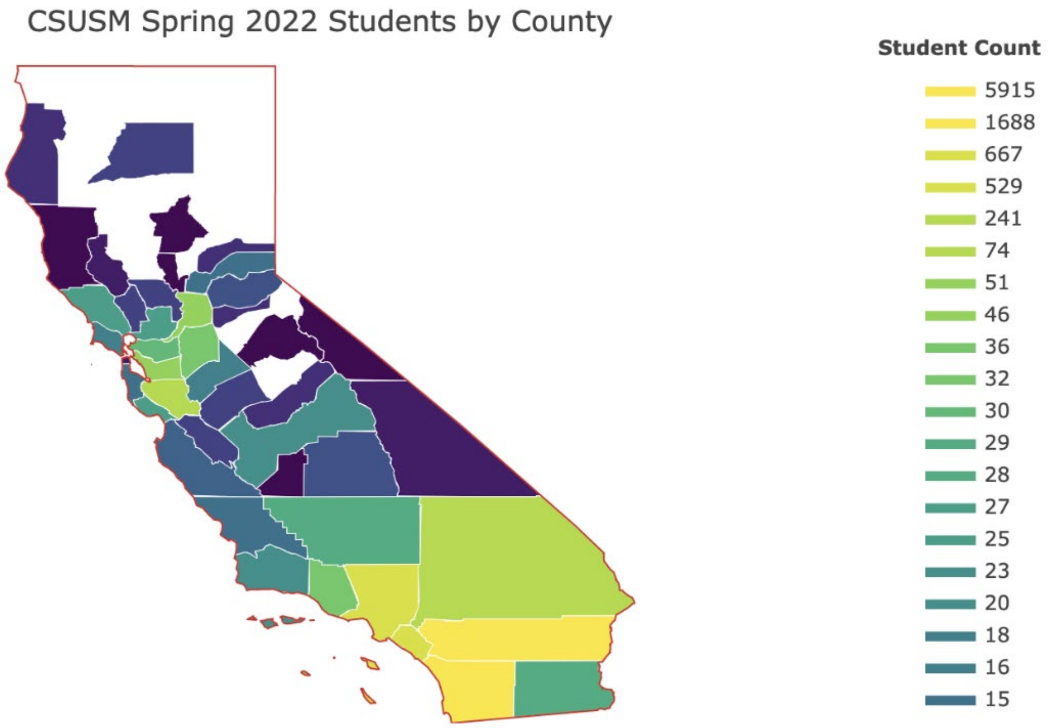


Figure 10: California Map with Student's Count per County
Higher concentration of students in San Diego and Riverside counties, followed by Orange, Los Angeles, and San Bernardino counties.

CSUSM is in San Diego County, at California Southwest region, about 40 miles away from the border with Mexico on South. We can assume that students that live too far from school would move closer, since commuting is not feasible.

Analyzing how many students are within a radius distance from campus location in San Marcos, we can see that half of the students live within 25 miles from school, and 80% live within 50 miles (figure 11). We assume that those almost eight thousand students commute to school.

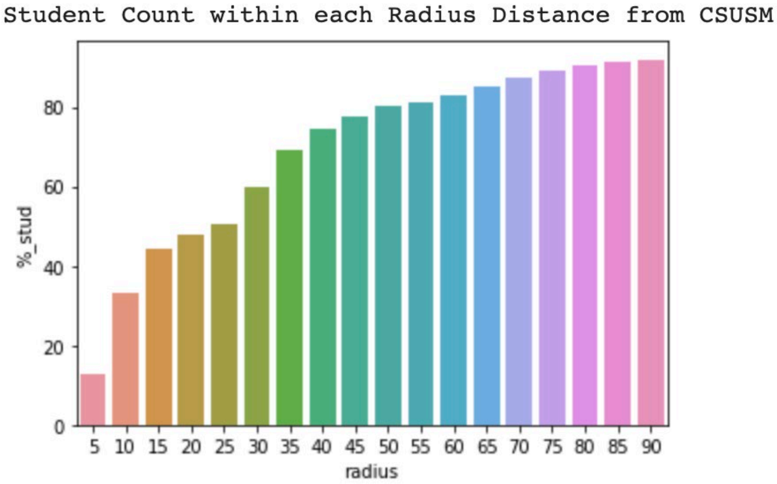


Figure 11: Student Count within each Radius Distance from CSUSM
50% of the students live within 25 miles, 80% within 50 miles radius distance

Analyzing location by zip code give us a better spacial distribution of the students. On figure 12, the legend shows the student range count. There is a sparse distribution above Los Angeles area, with higher concentration of students on the zip codes surrounding the main campus in San Marcos, San Diego County.

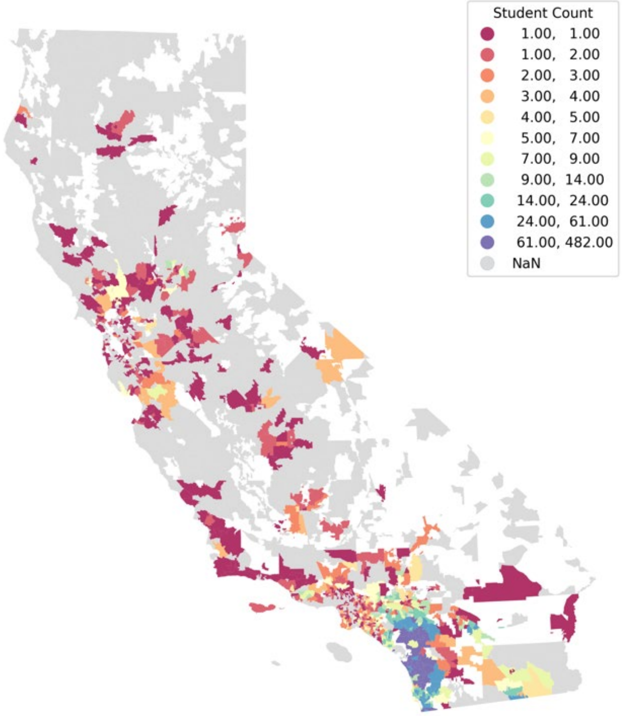


Figure 12: Student Count by Zip Code
Higher concentration of students on areas surrounding where CSUSM is located, in San Diego County

4 System Design

4.1 Current Manual Process

As briefly explained in the Introduction section, the current process of matching students is manual. After submitting a form, a person from the Parking Services department checks the student schedule, find other students with similar schedules, and look their locations on Google Maps to see if they are on similar routes. Then, the students are contacted by the department to receive a list of possible carpool candidates and their contact information. Because there are few students enrolled, it's feasible to do this process manually as it is.

4.2 Proposed Automated Process

The system proposed on this project automates the matching process of students to form carpools based on their location, schedule, and preferences, with the goal of increasing the efficiency to bring more students to the program.

4.2.1 Requirements

When a student submits a form to get help finding a carpool partner, they must enter some basic information about themselves and answer questions on their schedule and preferences. Their answers are used in the system to search for good matches. To create the model, we used the students' data from the class enrolment. Some required fields on the carpool form had to be included on the student's dataset. The new assigned values were based on the Transportation Survey and on the Carpool Program enrolment.

4.2.1.1 Constraints

To form the carpool groups, we need to filter student's locations and schedules to find matches. If students reside in the same neighborhood or on the same route to school, and have similar schedules, they are potential ride mates. Our main constraints are the user origin location with their arrival time in school, and the destination location with departure time from school.

Each carpool needs at least one driver. Since our dataset does not have this information, we assigned passenger and driver roles to students. Based on the form submission through the carpool link at Parking Services webpage, 9% of the students chose to be only a driver, while 49 % chose to be only a passenger, with the remaining 42% choosing either. Respecting these proportions, we randomly selected the role each student has. The same strategy was used to assign values for all new features.

Another constraint we have for our problem is the vehicle capacity. If a driver already drives with a passenger in the car or has a truck with one row of seats, the number of available seats is not going to be the standard five seats. We created a new column for available seats and assigned random values from one to three, limiting the total number of people to four per vehicle for comfort. We determined 14% for one seat, 11% for two seats, and the remaining 75% has the standard three seats available.

Ideally, we should form round trip carpools for all weekdays, but if it's not possible, we should have different carpool groups for different days of the week.

Table I: Constraint Summary Table

Constraints

Location	Zip code of residency
Schedule	Arrival and departure times for each day of the week
Role	Driver, passenger, or either
Number of Seats	How many available seats in the vehicle

4.2.1.2 Preferences

The goal of this project is to be able to form carpool groups based on their preferences, which tells the system what the preferred carpool partner characteristics are. Again, we had to create new features to establish those preferences. Based on our research and the collected data from the carpool link, we created preferences for non-smoker, same age group, same gender, and same academic status.

When filling up the form, students can select the characteristics they would like in their ride-mates. Around 40% of the students do not have any preference, 30% of the non-smoker students chose non-smoker ride-mates, 12% of the female students chose same gender, and 25% of the students chose the same academic status.

Since we do not have a smoker field in our original dataset, we used the information from a news article from CSUSM to determine that 10% of the students are smokers and assign it randomly to our dataset [30]. For their preferences, we assigned 30% of the non-smoker students to choose non-smoker ride-mates.

The class enrolment dataset has student's age, academic level, college, and gender. To reduce the computational time, instead of having too many values for each category, we created a new variable for 'under 25' and assign all students with age under 25 to this category. For academic level, we created 'undergrad status' and assigned all freshman, sophomore, junior, and senior to this. After that, we included preferences for the same age group and same status for 25% randomly selected students on both new categories of same age and same status. For the same gender category, we assigned this preference for 12% of the non-male students.

Table II: Preference Summary Table

Preferred Characteristics of the Carpool Partner

Non-Smoker	A non-smoker ride-mate
Same Age Group	A ride-mate in the same age group
Same Gender	A ride-mate with the same declared gender
Same Status	A ride-mate in the same academic status

Influenced by the good results achieved by Aydin et al. on their research, we compute a compatibility score to match driver with their passengers [27]. When searching for carpool mates, after respecting the constraints, we apply a formula to compute the compatibility score of the students. The base value may have three different values: -1, 0, or 1. If both students have the same values on a feature, the base value is one. When both students have different characteristics in the same feature, the base value can be 0 or -1, depending if there is a preference selection on this feature by any or both students. If no student selects this feature as preference, the base value is zero, otherwise the base value is negative one. When having a zero value for a feature where no preference is selected, we ensure that this feature does not impact the final score. For each student preference, a weight factor is applied to the feature. If the preference is not selected the weight is one. We assign a system defined factor of 2 to the preference selected by each student. Since the same college is not included in the preferences, it is used only to increase the final compatibility score when both students are from the same college. The system computes the compatibility score *CS* as the sum of all base features *b* multiplied by the weight of the feature chosen by the student A w^{A} and multiplied by the weight of the feature chosen by the student B w^{B} (equation 1).

$$CS = \sum b \cdot w^{A} \cdot w^{B}$$

Equation 1: Compatibility Score

Table III shows an example on how to apply the compatibility score between student A and student B. In this case, Student A and Student B are non-smokers, so their base score for this feature is 1. Both prefer a non-smoker rider, so both have weight 2 for non-smoker preference. When we multiply base 1 by student A weight factor of 2 and multiply it by the student B weight factor of 2, the result is 4 for the non-smoker feature. Following the same steps, we compute base value of 1 for gender, since both students are female, and only student B prefers to ride with the same gender, so only student B weight factor of 2 multiplies the base value. This gives a score of 2 for the gender feature. Students A and B are also in the same age group, but none of the students selected same age group as their preference. In this case, the base score is one, which hives a final score for age group of 1. On the last two categories, students A and B don't have the same characteristics nor chose these preferences, resulting on a score of 0 for features status and college. Finally, we sum the scores of all features and get the final compatibility score of 7 between student A and student B.

The advantage of using the compatibility score is to not use preference as a constraint. When forming carpool groups, the members with higher scores are selected. However, if the options are sparse, groups with lower score can still form a carpool. The compatibility score range is [-17, 17], with the minimum value occurring if both students have opposite characteristics on all features and select all possible preferences. The maximum score occurs when both students have the same characteristics and chose all possible preferences. We expect to have most of the scores ranging from [0,5], which is the range of scores that the system gets computing the base value of features, since almost 40% of students don’t select any preference. We interpret any positive score as a compatible match between driver and rider.

Table III: Compatibility Score Example

	STUDENT A		STUDENT B					
FEATURE	**Characteristic**	**Preference**	**Characteristic**	**Preference**	**b**	**w^A**	**w^B**	**Total**
NON-SMOKER	✓	✓	✓	✓	1	2	2	4
GENDER	Female		Female	✓	1		2	2
AGE GROUP	Under 25		Under 25		1			1
STATUS	Undergrad		Grad		0			0
COLLEGE	CSTEM		CEHHS		0			0
							FINAL COMPATIBILITY SCORE	**7**

When implementing this system, we can adapt the weights for level of importance of each feature. In the future we could have three factors: 1 for no preference, 2 for low level of preference, and 5 for high level of preference. The decision to keep a factor of two at the beginning of the implementation is based on the low level of enrolment, so the system has a higher influence on the final score. When the candidates pool increases, we can adapt to form groups based on more relevant features, giving the decision power to student’s choices.

4.2.2 Process Flow

A carpool exists only if there is a driver. Thus, the program starts selecting a student with ‘driver’ or ‘’ role from the pool. It then searches for students with ‘passenger’ or ‘’ role within the same location to find matches that respects the schedule constraint. The compatibility score is computed for all students in this cluster. The system then checks the capacity of the vehicle and selects the ones with highest scores, one by one, until it fills up all the available seats. The selected students to the carpool are withdrawn from the pool. If there are still available seats in the vehicle, the system checks for more students in the route to campus with a new search, otherwise, it returns the carpool and reinitiates the process until there are no more students in the pool. A diagram with the process flows of the carpool formation is presented on figure 13.

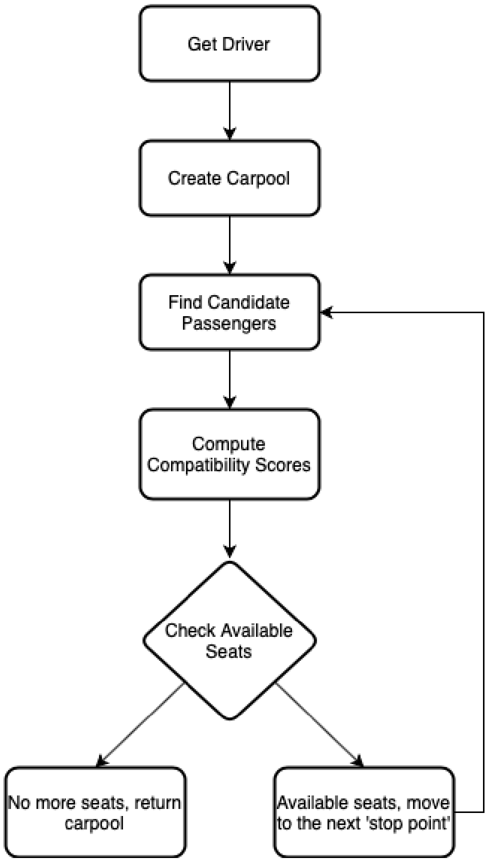

Get Driver
Create Carpool
Find Candidate Passengers
Compute Compatibility Scores
Check Available Seats
No more seats, return carpool
Available seats, move to the next 'stop point'

Figure 13: Flowchart Diagram of the Carpool Formation

First, a driver is selected from the pool to create a carpool, then search for passengers, and compute the compatibility score. If there are still available seats, restart the search for passenger on the next location, otherwise, save carpool and start a new one.

5 IMPLEMENTATION

5.1 ARCHITECTURE

The system has a client-server architecture, where the client interacts with a web portal, the server processes and responds the requests, and the data is stored in the database. The Parking Services department has access to a web portal to upload student's information to a database, the server runs an algorithm for clustering students based on constraints and preferences, and it returns a report to the web portal with the carpool information. The system architecture diagram shown below, on figure 1, represents the Model-View-Control, where the data processing and control, data storage, and views are decoupled. Following this architecture allows the program to scale up and introduce new features.

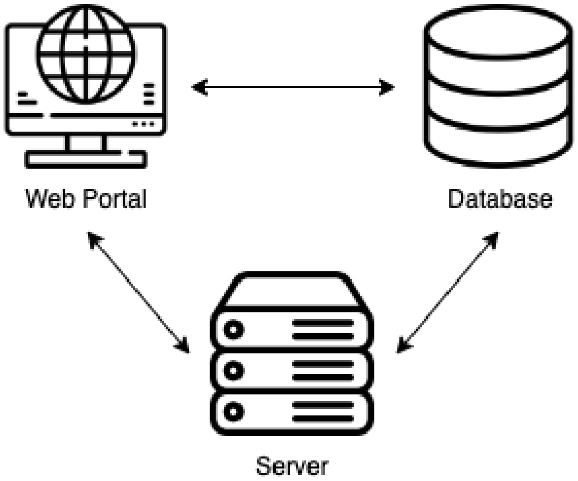


Figure 14: System Architecture [31]
Server – Web Portal – Database circular communication

5.2 LANGUAGE AND ENVIRONMENT

The program is written in Python, an open-source language with a variety of freely distributed libraries that can be used on application's back end or front-end. We also made use of SQL to manipulate the data stored in the database, and HTML/CSS on the creation of the webpages.

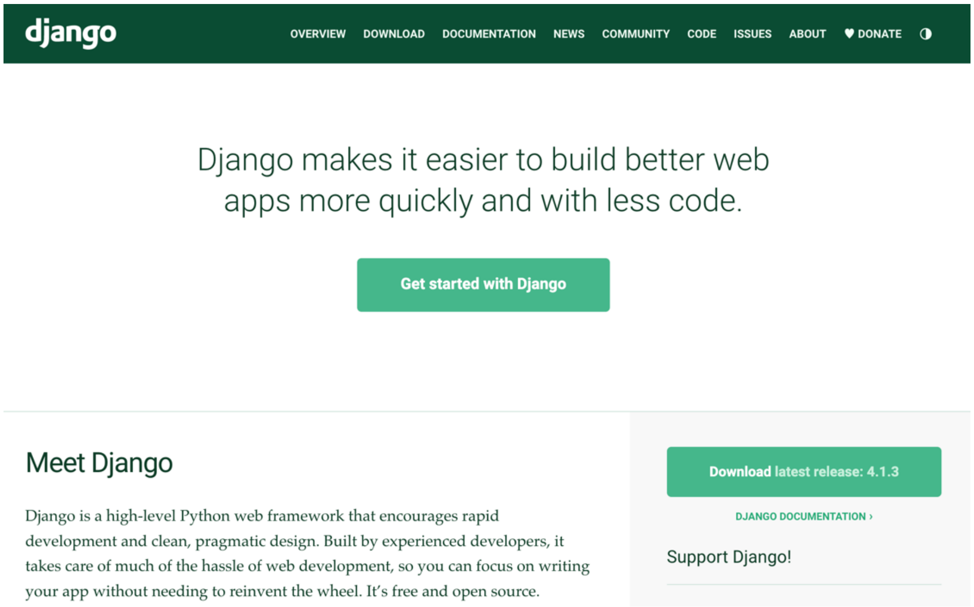
The carpool algorithm was written using Google Colab, a web application with interactive notebooks and free GPU access [34]. Because of its notebook style, it is easy to write and debug code. It also improves the quality of the code, since every new function can be extensively tested and validated. All experiments were done on Google Colab. After the program was created, the code was transferred to Visual Studio Code, and tests were done on a virtual environment with all necessary libraries installed in it. This procedure ensures the portability of the program to any server or web host. All dependencies within this project are saved in a file called 'requirements.txt' that can be used to install the required libraries and packages by running the command `pip install -r requirements.txt` on the terminal.

5.2.1 Server

The server is a Macintosh with a 3.1 GHz Dual-Core Intel Core i5 processor and 8 GB memory. Using the Visual Studio Code IDE, a virtual environment was created to install all necessary libraries and modules for creating and testing the program. The carpool program server-side uses Django, a Python web framework, to manage the database and the web portal [33] (figure16).

Django provides three layers: the model layer, the view layer, and the templates layer. The model layer is the object-relation mapper (ORM) that interacts with the database through its classes and function to create, read, update, and delete data. The view layer processes all requests from users and returns the responses. The template layer renders what needs to be presented to the user. Besides these three main layers, it also offers an admin interface, provides useful tools for developing and testing applications, security features, and so on.

The ORM is an API that abstracts the database, so the program can use any relational database to store the data. This feature allows the program to interact with the database through objects, making it possible to transfer the data to a different database in the future if desired.



Django makes it easier to build better web apps more quickly and with less code.

Meet Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

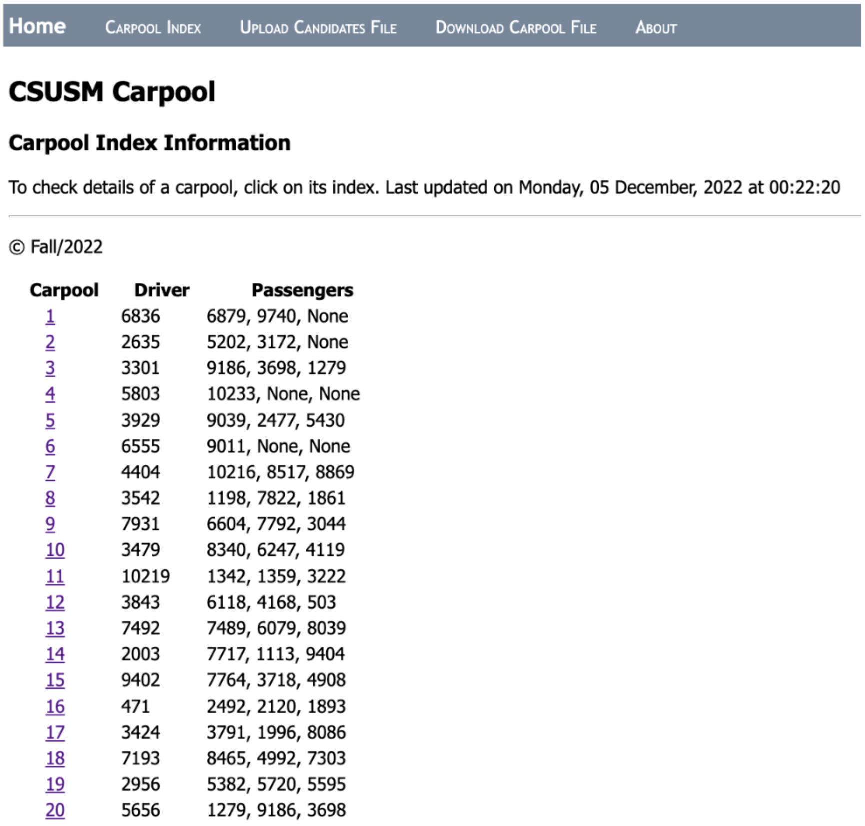
Support Django!

Figure 15: Django Web Framework
Web page of the Django framework for download and reference

5.2.2 Web Portal

The web portal has five webpages: Home, Carpool Index, Details, Upload File, and About. The pages were written in HTML and CSS using the Django tutorial on VS Code templates [34]. The Home page presents an overview of the web portal, the Carpool Index page has a list with the carpools in the database showing driver and passengers IDs with a hyperlink to show details of each carpool (figure 15).

When clicking on the hyper link, the user gets redirected to the details page of a specific carpool. As you can see on figure 18, it contains information on the origin of the carpool, which is the driver's zip code, how many seats this driver has available, a list of the passengers with their compatibility scores and schedules, and a list of Park and Ride locations within 3 miles of each passenger zip code as suggested meeting points. The about page explains the purpose of the web portal (figure 16).



CSUSM Carpool

Carpool Index Information

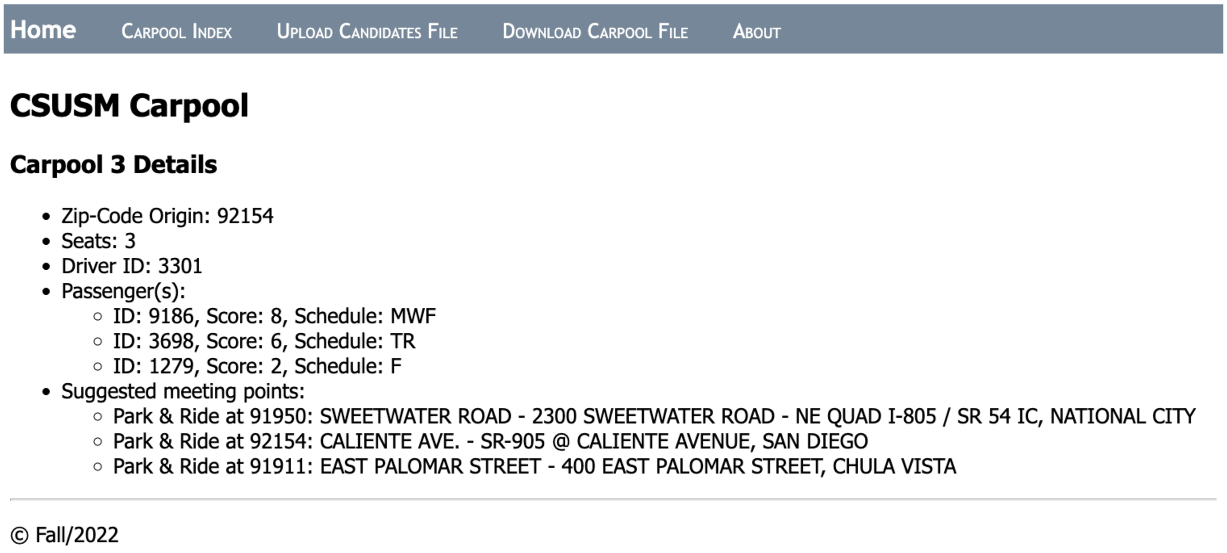
To check details of a carpool, click on its index. Last updated on Monday, 05 December, 2022 at 00:22:20

© Fall/2022

Carpool	Driver	Passengers
1	6836	6879, 9740, None
2	2635	5202, 3172, None
3	3301	9186, 3698, 1279
4	5803	10233, None, None
5	3929	9039, 2477, 5430
6	6555	9011, None, None
7	4404	10216, 8517, 8869
8	3542	1198, 7822, 1861
9	7931	6604, 7792, 3044
10	3479	8340, 6247, 4119
11	10219	1342, 1359, 3222
12	3843	6118, 4168, 503
13	7492	7489, 6079, 8039
14	2003	7717, 1113, 9404
15	9402	7764, 3718, 4908
16	471	2492, 2120, 1893
17	3424	3791, 1996, 8086
18	7193	8465, 4992, 7303
19	2956	5382, 5720, 5595
20	5656	1279, 9186, 3698

Figure 16: Carpool Index Webpage

The webpage has instructions on how to load the details of a carpool, along with the last update, and general carpools' information



CSUSM Carpool

Carpool 3 Details

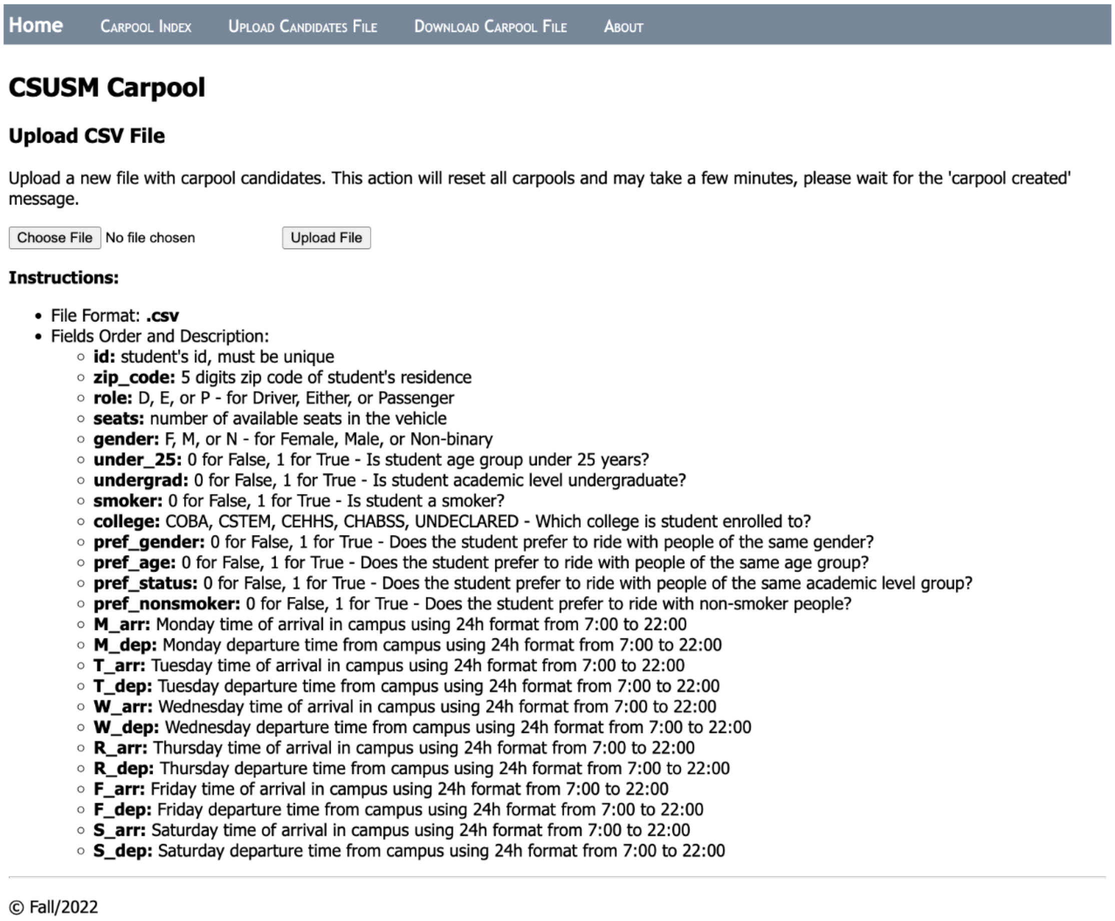
- Zip-Code Origin: 92154
- Seats: 3
- Driver ID: 3301
- Passenger(s):
  - ID: 9186, Score: 8, Schedule: MWF
  - ID: 3698, Score: 6, Schedule: TR
  - ID: 1279, Score: 2, Schedule: F
- Suggested meeting points:
  - Park & Ride at 91950: SWEETWATER ROAD - 2300 SWEETWATER ROAD - NE QUAD I-805 / SR 54 IC, NATIONAL CITY
  - Park & Ride at 92154: CALIENTE AVE. - SR-905 @ CALIENTE AVENUE, SAN DIEGO
  - Park & Ride at 91911: EAST PALOMAR STREET - 400 EAST PALOMAR STREET, CHULA VISTA

© Fall/2022

Figure 17: Carpool Details Webpage

Origin, available seats, riders' IDs, scores, schedules and suggested meeting points are provided

The web portal is for the Parking Services department usage. There is a submit button to upload a csv file with the carpool candidates to send the student's information to the database. The file must follow the format and sequence of columns of the database, which is described on the webpage (figure 19). After submitting a new file, the candidates table in the database is updated with the new candidates' information and the carpool program resets to create new carpools.



CSUSM Carpool

Upload CSV File

Upload a new file with carpool candidates. This action will reset all carpools and may take a few minutes, please wait for the 'carpool created' message.

Instructions:

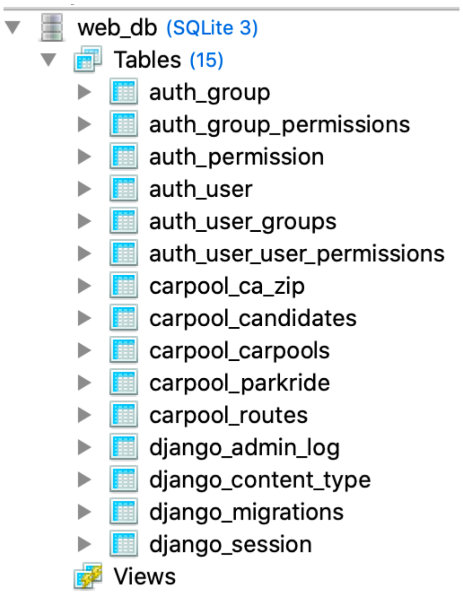
- File Format: **.csv**
- Fields Order and Description:
  - **id:** student's id, must be unique
  - **zip_code:** 5 digits zip code of student's residence
  - **role:** D, E, or P - for Driver, Either, or Passenger
  - **seats:** number of available seats in the vehicle
  - **gender:** F, M, or N - for Female, Male, or Non-binary
  - **under_25:** 0 for False, 1 for True - Is student age group under 25 years?
  - **undergrad:** 0 for False, 1 for True - Is student academic level undergraduate?
  - **smoker:** 0 for False, 1 for True - Is student a smoker?
  - **college:** COBA, CSTEM, CEHHS, CHABSS, UNDECLARED - Which college is student enrolled to?
  - **pref_gender:** 0 for False, 1 for True - Does the student prefer to ride with people of the same gender?
  - **pref_age:** 0 for False, 1 for True - Does the student prefer to ride with people of the same age group?
  - **pref_status:** 0 for False, 1 for True - Does the student prefer to ride with people of the same academic level group?
  - **pref_nonsmoker:** 0 for False, 1 for True - Does the student prefer to ride with non-smoker people?
  - **M_arr:** Monday time of arrival in campus using 24h format from 7:00 to 22:00
  - **M_dep:** Monday departure time from campus using 24h format from 7:00 to 22:00
  - **T_arr:** Tuesday time of arrival in campus using 24h format from 7:00 to 22:00
  - **T_dep:** Tuesday departure time from campus using 24h format from 7:00 to 22:00
  - **W_arr:** Wednesday time of arrival in campus using 24h format from 7:00 to 22:00
  - **W_dep:** Wednesday departure time from campus using 24h format from 7:00 to 22:00
  - **R_arr:** Thursday time of arrival in campus using 24h format from 7:00 to 22:00
  - **R_dep:** Thursday departure time from campus using 24h format from 7:00 to 22:00
  - **F_arr:** Friday time of arrival in campus using 24h format from 7:00 to 22:00
  - **F_dep:** Friday departure time from campus using 24h format from 7:00 to 22:00
  - **S_arr:** Saturday time of arrival in campus using 24h format from 7:00 to 22:00
  - **S_dep:** Saturday departure time from campus using 24h format from 7:00 to 22:00

© Fall/2022

Figure 18: Upload File Webpage
Webpage with instructions of the order and format of the data to upload a file

5.2.3 Database

We used SQLite, a self-contained database system, hosted on a local machine. Figure shows all tables in the database. The tables with names starting with 'django' and 'auth' are automatically created by the system to store information such as user's profile, login, password; and systems' logs such as changes in the database. The carpool program has four tables: Candidates, Routes, ParkRide, and Carpools. We store student's information on Candidates table, with student id, along with zip code, profile information and preferences. On the Routes table we store the pre-processed routes from most common student's zip code as origin to campus. The results of the clustering are stored in the Carpools table, and each carpool has information of its driver's ID's, each passenger IDs with their respective compatibility score, schedule, and direction. Park and ride information from Caltrans is stored on ParkRide table, and the CA_zip contains the Census information for California zip codes with coordinates of the central point of each zip code area along with its county.



```
web_db (SQLite 3)
  Tables (15)
    auth_group
    auth_group_permissions
    auth_permission
    auth_user
    auth_user_groups
    auth_user_user_permissions
    carpool_ca_zip
    carpool_candidates
    carpool_carpools
    carpool_parkride
    carpool_routes
    django_admin_log
    django_content_type
    django_migrations
    django_session
  Views
```

Figure 19: Tables' List in the Database
Tables to store auth, carpool, and Django logs and data

To access the data in the database we used the Django ORM, so it's possible to migrate the data to a different database in the future.

5.3 IMPLEMENTATION

During the data collection, data cleaning, and data preparation, we mainly used Panda's library and data visualization tools to understand the student's profile and creating the missing characteristics and preferences. To create the routes from student's origin to school, we created a base road network graph using a bounding box around California State University San Marcos that could include all zip codes within 90 miles from campus. The bounding box limits are San Antonio Heights in San Bernardino County on

North, Long Beach in Los Angeles County on West, Imperial Beach in San Diego County on South, and Borrego Springs in San Diego County border with Imperial County on East (figure 20).

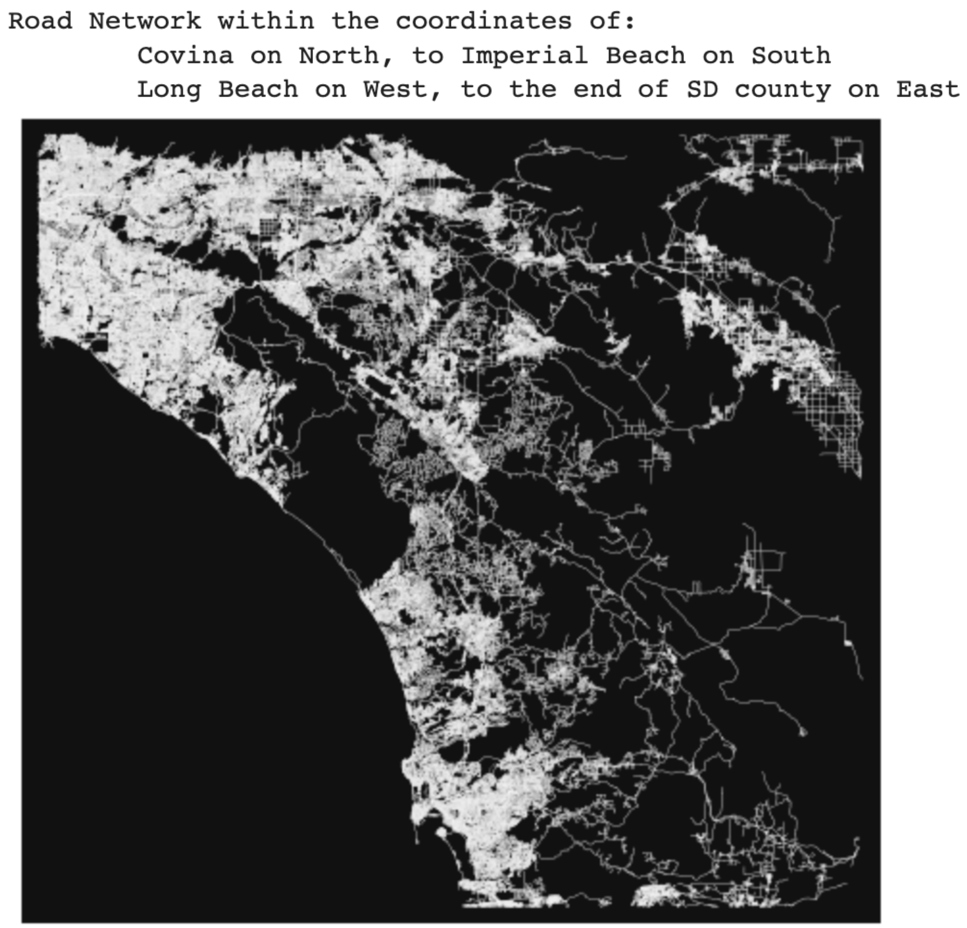

Road Network within the coordinates of:
Covina on North, to Imperial Beach on South
Long Beach on West, to the end of SD county on East

Figure 20: Bounding Box Road Map
From Los Angeles County on West, San Bernardino County on North, Imperial County on East, and Mexico border on South

We used OSMnx for Python, that is built on top of the Networkx library and the Open Street Map [32]. The Open Street Map is a free open-content map with information from all around the world that is constantly being updated by volunteers and contributors. Using this base map, we run the program to create the routes from the most common zip codes found on the student's dataset. Each node from the route contains information about its location, so we can retrieve all zip codes that are crossed on the route using a reversed function from the Geopy geocoder to find the zip code with the coordinates. We store all route's information on the database.

The process of creating the road network graph and finding the routes from origin to destination takes time. Thus, to speed up the process, routes from common zip code locations found on the student's database with CSUSM destination are processed before the clustering algorithm for a faster look up in the database. When a new student is selected from the pool to start a new carpool cluster, we just need to check the student's location and retrieve the route from the database. If the location is not found in the database, then the program needs to process the route, which is then stored in the database for future reference.

Table IV presents the main Python packages used on this project with a brief description of its purpose. During the data analysis of the classes and students' data, Pandas dataframes were used to store and manipulate the data. Seaborn and matplotlib were used to plot profile feature charts. Google, pgeocode, geopy, pyzipcode were used to get geocode information from zip codes. Osmnx and networkx were used to build the road network base map and create the routes. Folium, Plotly, Geopandas, and matplotlib were used to plot maps and map charts. Django framework was used to create the server-side management of the web portal and database.

Table IV: Python libraries, modules, and frameworks

Library/ Module	Description
Pandas	Data analysis and manipulation
Matplotlib, Seaborn, and Plotly	Data visualization
Datetime, Time	Manipulation of date/time objects
Google	Specific tools for use on Google Colab notebooks and access Google Cloud services
Pgeocode, Geopy	Retrieve locations' information from postal codes or partial address using Nominatim geocoder
Geopandas	For working with geospatial data
PyZipCode	Retrieve information from zip code
Folium	Data visualization on an interactive map
OMSnx	Road network graphs with data from OpenStreetMap
Networkx	Complex network analysis
Django	Web framework

To choose ride mates the system uses a greedy approach. It randomly chooses a driver from the pool and finds students who matches the constraints of schedule and location, and then it chooses the ones with higher compatibility score. The driver and the passengers that form a carpool are withdrawn from the pool. If the driver still has available seats, the system continues the search for matches through the driver's route to school until all seats are occupied or they arrive on campus. When creating the routes to campus, beside storing the routes node by node, the program saves all unique zip codes from the nodes on the way. This 'zips on the route' list is used to search for students within 5 miles of each zip. This is the radius of maximum detour the system finds reasonable for the driver and the rider to find a meeting point. Once the pool of candidates is large enough, we can adjust it to a smaller radius. This search occurs on each "stop" for the schedule's possible matches for all weekdays, group days, and individual days. For example, when a driver is selected, the system searches for candidates in the same zip code, check their schedules, saving students in separate lists for each schedule. Next, the system computes the scores and sorts the candidates' list, starting with the schedule list for all weekdays, from Monday to Saturday. It removes the candidate with higher score and place them in the carpool, one by one, until all seats are occupied, or the list is empty. If there are no matches or not enough matches for all weekdays, then the system investigates the Monday-Wednesday-Friday list, selecting the highest score candidates from this list. After, it searches the Tuesday-Thursday and follows the same pattern. Only after all the weekdays and group days have been searched, and if there are still seats available, the system starts looking on each weekday to fill up the seats with the highest score candidates. This process occurs on each zip code of the path until all seats are occupied or the driver arrives in campus.

6 EVALUATION AND DISCUSSION

6.1 EXPERIMENTS

The full dataset has students from all locations, so we selected 7893 students within the radius of 50 miles from school. Our analysis of the student profile shows that 80% of the students reside within this radius, and we assume that students that live further than 50 miles would move closer to school during classes. From this pool of students, we select random sets of different sizes from 100 to 2500 students to run the experiments. To test our implementation, we ran four experiments on how to create the carpools: limiting the radius search from the driver's origin to find riders; increasing the number of days to match driver's schedule with candidate's schedule, keeping the passenger in the pool after finding a match for some day of the week to allow passengers to ride with other drivers on different days, and alleviating the schedule constraint up to 30 min. All experiments were done on Google Colab using Python 3 Google Compute Engine backend with no GPU.

6.1.1 Radius

From the original location of the driver, we do a search of n miles radius to do a search for candidates on the surroundings. For each zip code that the driver crosses along the way to school, we do another search with the same n radius. Because our dataset does not have the exact location of the students, but their zip code, our search is based on the geographical center of the zip code area, and it searches in the selected radius from the center of the area. In more populated areas, the zip code area is smaller, and on less populated areas the zip code area is bigger. When choosing the radius, if it is too small, it may stay inside a zip code in an unpopulated region, whereas if it is too big, it may cover many different zip codes in the overpopulated areas.

6.1.2 Weekday Schedule

In the search for passengers that matches driver's schedule, we have the options to constraint the search for all weekdays, or break it down into groups of days, and each day separately. On each stop on search for candidates, the system searches for students with matching schedules, compute their scores, and select the ones with highest scores. We tested the options of the matching schedule for all weekdays from Monday to Saturday, then we tried all weekdays with two more lists of students with matching schedules for group days of Monday-Wednesday-Friday, and Tuesday-Thursday. Finally, we experiment searching for matches on all individual days as well, but only after finishing the search for all weekdays and group days.

6.1.3 Passenger with more than one driver

On the search for ride-mates, first selects a driver that is withdraw from the pool, followed by withdrawing passengers that forms a carpool group with the forementioned driver. To allow the opportunity of passengers finding rides with different drivers for different days of the week, we can keep all the candidates with passenger role in the pool until all drivers are exhausted.

6.1.4 Time Constraint Relaxation

Finding candidates within a restrict location is a limitation itself and adding the schedule constraint to it reduces the pool even more. The classes schedule is not consistent on start time and end time, with options every 5 minutes, so students may have a similar schedule on a day but with a time delta of a few minutes due to this variety of class schedules. For instance, a student A may have to be in campus for a class starting at 9:20 and leave campus after the end of a class at 11:45. If student B needs to be in campus by 9:30 and has class until noon, even if student A and student B live close, they are not considered as potential ride-mates because their schedule do not match exactly. To allow a wider range of matches, we can relax the time constraint for a range up to 30 minutes when matching schedules. To avoid making the driver wait for passengers, we search for candidates with start time up to 30 minutes after the driver's start time, and end time up to 30 minutes before the driver's end time.

6.2 Results

6.2.1 Radius

The change of values in the radius from 0 to 5 miles for searching for riders on the driver's path can have an impact of up to 40% in the search time. Radius and time are inversely proportional, so increasing the radius decreases the processing time. In our experiments, that translates to a difference of up to 20 seconds on a dataset of 1500 students. Moreover, increasing the radius improves the efficacy of the system, increasing the matching riders by almost 10% when comparing radius zero to radius five. Table V shows the testing results on a 1500 students pool using the any day schedule.

Table V: Radius in miles on a pool of 1500 students

Radius (miles)	Carpools	Total Unmatched		Total Matched		Time (sec)
0	650	448	30%	1052	70%	59
1	653	440	29%	1060	71%	56
2	676	408	27%	1092	73%	64
3	678	349	23%	1151	77%	53
4	697	315	21%	1185	79%	49
5	702	296	20%	1204	80%	49

6.2.2 Weekday Schedule

Our results show that finding students that matches the schedule for all days of the week is computational expensive. It takes more than twice of the time that is needed when grouping days of the week like Monday-Wednesday-Friday and Tuesday-Thursday. And it takes almost 10 times longer than searching for each day individually. These results are consistent for different dataset sizes. The tables bellow shows the results considering the search radius of 5 miles.

Table VI: Schedule all weekdays (MTWRFS)

Dataset Size	Carpools Created	Total Unmatched		Total Matched		Time (sec)
2500	178	2058	82%	442	18%	128
2000	134	1617	81%	329	16%	92
1500	81	1301	87%	199	13%	69
1000	48	885	89%	115	12%	44
500	15	464	93%	36	7%	19
400	9	379	95%	21	5%	17
300	4	290	97%	10	3%	17
200	4	191	96%	9	5%	7
100	2	95	95%	5	5%	4

In the search for matching the schedule for all weekdays, the system chooses a driver and searches in the pool for students that have the same schedule of arrival to campus and departure from campus. If it doesn't find it, it chooses another driver and repeats the process, until there are no more drivers in the pool. With this approach, as expected, more carpools can be formed if there are more candidates in the pool. However, even with a pool of 2500 students, only 18% of the students can find a partner with the same schedule, and only 5% for a small pool of 100 students.

When broadening the search for grouping days, the system starts trying to find students that have the same schedule for all days of the week. However, in not finding matches or if there are still seats available in the car, it then groups Monday-Wednesday-Friday, and Tuesday-Thursday and searches for students to fill up empty seats if there are still any available. With this approach, we increase to 48% of matched students if the pool has 2500 candidates, but we still have only 14% matches if the pool is small as 100 students. With the grouping search, we ensure that students can find ride mates for at least 2 days a week, but as the pool must be greater than 2500 students to find riders for half of the students.

Table VII: Schedule grouping weekdays (MTWRFS, MWF, TR)

Dataset Size	Carpools	Total Unmatched		Total Matched		Time (sec)
2500	432	1293	52%	1207	48%	113
2000	322	1096	55%	904	45%	85
1500	222	890	59%	610	41%	66
1000	133	645	65%	355	36%	42
500	59	356	71%	144	29%	19

400	43	300	75%	100	25%	20
300	25	242	81%	58	19%	12
200	15	164	82%	36	18%	7
100	6	86	86%	14	14%	3

Searching first for all weekdays, then narrowing it down for group days, and adding a new search for each day to fill up empty seats is the fastest and most efficient search, ensuring that 80% of the students can find a partner to carpool at least one day a week for big pools above 2000 students.

Table VIII: Schedule Grouping Any Day (MTWRFS, MWF, TR, M, T, W, R, F, S)

Dataset Size	Carpools	Total Unmatched		Total Matched		Time (sec)
2500	632	419	17%	2081	83%	57
2000	496	381	19%	1619	81%	41
1500	357	369	25%	1131	75%	33
1000	231	288	29%	712	71%	25
500	114	173	35%	327	65%	14
400	94	146	37%	254	64%	12
300	60	143	48%	157	52%	9
200	38	104	52%	96	48%	6
100	16	60	60%	40	40%	3

6.2.3 Passenger with more than one driver

The above experiments have an implementation where driver and passengers are taken out of the pool when they start a carpool. We assessed the option of keeping the passengers in the pool to try to find rides with other drivers for different days of the week. The results don't improve much on big pools above 2000 students, with a more robust increase of almost 10% in smaller pools. The tests are for the schedule group of any day (MTWRFS, MWF, TR, M, T, W, R, F, S) with 5 miles radius.

Table IX: Keeping Passengers in the Pool

Dataset Size	Carpools Created	Total Unmatched		Total Matched		Time (sec)
2500	1216	392	16%	2108	84%	93
2000	951	336	17%	1664	83%	75
1500	702	295	20%	1205	80%	54
1000	466	228	23%	772	77%	47
500	215	139	28%	361	72%	21
400	169	126	32%	274	69%	17
300	109	125	42%	175	58%	13
200	66	92	46%	108	54%	9
100	29	51	51%	49	49%	4

6.2.4 Time Constraint Relaxation

If we consider the time constraint as a time window, where we could match drivers and candidates within a time window of 30 minutes, we improve the matching rate by 5% on average. The tests are for the schedule group of any day (MTWRFS, MWF, TR, M, T, W, R, F, S) with 5 miles radius.

Table X: Time Window of 30 Minutes

Dataset Size	Carpools Created	Total Unmatched		Total Matched		Time (sec)
2500	1227	375	15%	2125	85%	126
2000	998	229	15%	1701	85%	85
1500	698	263	18%	1237	82%	61
1000	470	189	19%	811	81%	43
500	234	129	26%	371	74%	27
400	192	104	26%	296	74%	24
300	121	98	33%	202	67%	14
200	82	67	34%	133	67%	19
100	33	48	48%	52	52%	5

7 CONCLUSION AND FUTURE WORK

The carpool program at CSUSM can implement the system using the radius of 5 miles and the search for matching any weekday schedule. We define the system default radius in 5 miles, which is the maximum detour for a driver, or the maximum passenger route length to meet a driver, this increases the matching efficacy in 10%, and it is still feasible for drivers and riders to find a meeting point in this area radius. The any day schedule option ensures that almost 50% of the students would be able to find a ride for at least one day a week. And the relaxation of the time constraint on searching for candidates in a time window of 30 minutes also improves the results in 5% on small pools. Our results show that for pools of 1500 candidates we could form carpools in less than 1 minute. The only vanpool research paper we found has a simulated dataset of 600 users and finds solutions in up to 15 minutes.

7.1 FUTURE WORK

The carpool tool is meant to be used by the Parking Services department; however, we think that it would have a higher adoption if it was a mobile application. In the future, we would like to extend this tool to the mobile environment. This way, students could enroll straight into the mobile platform, that could have chatted and rating features for user's interaction. This approach students could find carpool candidates themselves, forming long-term carpools or finding rides on the go for any given day.

8 REFERENCES

[1] C. Gros, "CBS8," 04 10 2022. [Online]. Available: https://www.cbs8.com/article/traffic/gas-prices/san-diego-county-gas-price/509-a12805c9-14de-4dd3-8f4c-9d5afe759815. [Accessed 05 11 2022].

[2] S. Shaheen, A. Cohen and A. Bayen, "The Benefits of Carpooling," *UC Berkley: Transportation Sustainability Research Center,* 22 10 2018.

[3] CSUSM, "Sustainability Master Plan," 12 2018. [Online]. Available: https://www.csusm.edu/sustainability/docs/sustainabilitymasterplan.pdf. [Accessed 05 11 2022].

[4] CSUSM, "Carpool Program," 2022. [Online]. Available: https://www.csusm.edu/parking/commuter/carpoolprogram.html. [Accessed 05 11 2022].

[5] N. D. Chan and S. A. Shaheen, "Ridesharing in North America: Past, Present, and Future," *Transport Reviews,* vol. 32, no. 1, pp. 93-112, January 2012.

[6] K. H. Ting, L. S. Lee, S. Pickl and H.-V. Seow, "Shared Mobility Problems: A Systematic Review on Types, Variants, Characteristics, and Solution Approaches," *Appl. Sci.,* vol. 11, p. 7996, 2021.

[7] A. Aguiléra and E. Pigalle, "The Future and sustainability of Carpooling Practices. An Identification of Research Challenges.," *Sustainability,* vol. 13, no. 11824, p. 11824, 2021.

[8] L. E. Olsson, R. Maier and M. Friman, "Why do They Ride with Others? Meta-Analysis of Factors Influencing Travelers to Carpool.," *Sustainability,* vol. 11, p. 2414, 2019.

[9] R. Guidotti, A. Sassi, M. Berlingerio, A. Pascale and B. Ghaddar, "Social or Green? A data-driven approach for more enjoyable carpooling," in *IEEE 18th International Conference on Intelligent Transportation Systems*, 2015.

[10] S. Adelé and C. Dionisio, "Learning from the real practices of users of a smart carpooling app.," *European Research Review,* pp. 12-39, 2020.

[11] S. Liakopoulou, M. M. Kakana, P. Avtji, E. Genitsaris and A. Naniopoulus, "Investigating the preferences of students towards the creation of a carpooling system serving the academic bodies of Thessaloniki city," in *3rd Conference on Sustainable Urban Mobility*, Volos, Greece, 2016.

[12] F. Galatoulas, S. Koutra, P. Rycerski, M. I. L. Candanedo and C. S. Ioakimidis, "A Comparative Study on User Characteristics of an E-car Pooling Service in Universities in Europe," in *6th International Conference on Smart Cities and Green ICT Systems*, 2017.

[13] D. Madubuike, "Changing Car Culture Towards Carpooling: A Case Study in Binghamton University (Part I: Analysis and Measurement of Potential)," *Alpenglow: Binghamton University Undergraduate Journal of Research and Creativity,* vol. 3, no. 1, pp. 1-26, 2017.

[14] CSUSM, "Enrollment Summary," [Online]. Available: https://tableau.csusm.edu/views/StudentProfile_0/EnrollmentSummary?%3Aembed=y&%3AshowShareOptions=true&%3Adisplay_count=no&%3AshowVizHome=no. [Accessed 06 11 2022].

[15] A. Mourad, J. Puchinger and C. Chu, "A survey of models and algorithms for optimizing shared mobility.," *Transportation Research Part B: Methodological,* vol. 123, pp. 323-346, 2019.

[16] M. Bouzid, I. Alaya and M. Tagina, "A Bee Colony Optimization Algorithm for the Long-Term Car Pooling Problem.," in *15th International Conference on Software Technologies*, 2020.

[17] W. Peng and L. Du, "Investigating Optimal Carpool Scheme by a Semi-Centralized Ride-Matching Approach," *Transactions on Intelligent Transportation Systems,* 2021.

[18] J. Xia, K. M. Curtin, W. Li and Y. Zhao, "A New Model for a Carpool Matching Service," *PLoS One,* vol. 10, no. 6, p. e0129257, 2014.

[19] G. Kutiel and D. Rawitz, "Local Search Algorithm for the Maximum Carpool Matching Problem," in *25th Annual European Symposium on Algorithms*, 2017.

[20] V. T. N. Nha, S. Djahel and J. Murphy, "A Comparative Study of Vehicles' Routing Algorithms for Route Planning in Smart Cities," in *First International Workshop on Vehicular Traffic Management for Smart Cities (VTM)*, 2012.

[21] F.-S. Hsieh and F.-M. Zhan, "A Discrete Differential Evolution Algorithm for Carpooling," in *42nd IEEE International Conference on Computer Software and Applications*, 2018.

[22] M. A. A. Santos, C. M. Santos, S. I. Martínez, J. A. C. Rocha, J. L. Menchaca, J. D. T. Villanueva, M. G. T. Berrones, J. P. Cobos and E. C. Rocha, "A Comparison of Machine Learning Techniques in the Carpooling Problem," *Journal of Computer and Communications,* vol. 8, pp. 159-169, 2020.

[23] S. El Salhi, F. Farouq, R. Obeidallah, Y. Kilani and E. Al Shdaifat, "Real-Time Carpooling Application based on k-NN Algorithm: A Case Study in Hashemite University," *International Journal of Advanced Computer Science and Applications,* vol. 10, no. 12, pp. 251-257, 2019.

[24] C. M. Boukhater, O. Dakroub, F. Lahoud, M. Awad and H. Artail, "An Intelligent and Fair GA Carpooling Scheduler as a Social Solution for Greener Transportation," in *17th IEEE Mediterranean Electrotechnical Conference*, 2014.

[25] S. Yan and C.-Y. Chen, "A model and a solution for the carpooling problem with a pre-matching information," *Computers & Industrial Engineering,* vol. 61, pp. 512-524, 2011.

[26] M. Brugliere, D. Ciccarelli, A. Colornia and A. Luè, “PoliUniPool: a carpooling system for universities,” *Procedia Social and Behavioral Sciences,* vol. 20, pp. 558-567, 2011.

[27] O. f. Aydin, I. Gokasar and O. Kalan, “Matching algorithm for improving ride-sharing by incorporating route splits and social factors,” *PLoS ONE,* vol. 15, no. 3, p. e0229674, 2020.

[28] F. Martelli and M. E. Renda, “Comparison of Trip Matching Algorithms for Mobility Sharing Applications,” in *IEEE 22nd International Symposium on a World Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2021.

[29] F. Librino, M. E. Renda, P. Santi, F. Martelli, G. Resta, F. Duarte, C. Ratti and J. Zhao, “Home Working for Social Mixing,” *Transportation,* 2019.

[30] CSUSM, “CSUSM Will Become Smoke-Free Campus in Fall,” 13 April 2017. [Online]. Available: https://news.csusm.edu/csusm-will-become-smoke-free-campus-in-fall/#:~:text=Cal%20State%20San%20Marcos%2C%20which,cigarettes%20will%20also%20be%20banned.. [Accessed 07 10 2022].

[31] FlatIcon. [Online]. Available: https://www.flaticon.com/free-icons. [Accessed 11 2022].

[32] G. Boeing, “OSMnx: Python for Street Networks,” [Online]. Available: https://geoffboeing.com/2016/11/osmnx-python-street-networks/. [Accessed 11 2022].

[33] F.-S. Hsieh, “Trust-based Recommendation for Shared Mobility Systems Based on a Discrete Self-Adaptive Neighborhood Search Differential Evolution,” *Electronics,* vol. 11, p. 776, 2022.

[34] N. Alisoltani, M. Ameli, M. Zargayouna and L. Leclercq, “Space-time clustering-based method to optimize shareability in real-time ride-sharing.,” *PLoS ONE,* vol. 17, no. 1, p. e0262499, 2022.

[35] Y. Sun, Z.-L. Chen and L. Zhang, “Nonprofit peer-to-peer ridesharing optimization,” *Transportation Research Part E,* vol. 142, p. 102053, 2020.

[36] Z. Yu, Y. Guo and Y. Chen, “Learning Trajectory Routing with Graph Neural Networks,” *ICBDC,* pp. 121-126, 2020.

[37] N. Masoud and R. Jayakrishnan, *Transportation Research Part B,* vol. 99, pp. 1-29, 2017.

[38] M. Tang, S. Ow, W. Chen, Y. Cao, K.-w. Lye and Y. Pan, “The Data and Science Behind GrabShare Carpooling,” in *International Conference on Data Science and Advanced Analytics*, 2017.

[39] M. Schreieck, H. Safetli, S. A. Siddiqui, C. Pflügler, M. Wiesche and H. Krcmar, “A Matching Algorithm for Dynamic Ridesharing,” *Transportation Research Procedia,* vol. 19, pp. 272-285, 2016.

[40] I. Hussain, L. Knapen, S. Galland, A.-U.-H. Yasar, T. Bellemans, D. Janssens and G. Wets, “Organizational-based model and agent-based simulation for long term carpooling,” *Future Generation Computer Systems,* vol. 64, pp. 125-139, 2016.

[41] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks.," *PNAS,* vol. 111, no. 37, pp. 13290-13294, 2014.

[42] "The Vanpool Assignment Problem: Optimization models and solution algorithms," *Computers & Industrial Engineering,* vol. 66, pp. 24-40, 2013.

[43] I. Zidi, K. Mesghouni, K. Zidi and K. Ghedira, "A multi-objective simulated annealing for the multi-criteria dial a ride problem," *Engineering Applications of Artificial Intelligence,* vol. 25, pp. 1121-1131, 2012.

[44] H. Tenkanen, V. Heikinheimo, H. W. Aagesen and C. Fink, "Automating GIS Processes," 2022. [Online]. Available: https://autogis-site.readthedocs.io/en/latest/course-info/general-information.html. [Accessed 08 11 2022].

9 APPENDIX

9.1 LITERATURE REVIEW

Table XI: Summary of Social Factors on Carpooling

Paper	Theme	Method
Aguiléra, 2021 [7]	Carpooling practices	Extensive research of academic literature of 97 papers from 2010 to 2021
Adelé, 2020 [10]	Survey of real users of carpooling app	Summary of literature review on human factors and carpooling, analysis of a carpool app usage data, and interviews
Olsson, 2019 [8]	Carpoolers motivation, worries, and characteristics	Survey of 17 studies from 2014 and 2018
Shaheen , 2018 [2]	Carpooling benefits	Environmental and socio-economic analysis
Galatoulas, 2017 [12]	E-carpooling user characteristics	On-line survey of students in two universities in Spain and Belgium to find their preferences and practices towards electric mobility carpooling
Madubuike, 2017 [13]	Changing culture from single driver to carpooling	Online survey of Binghamton University (NY) to understand needs that would lead to creating a successful carpooling system
Liakopoulou, 2016 [11]	Investigating preferences	Extensive web-based and on field questionnaire survey addressed to students of 3 universities in Thessaloniki city
Guidotti, 2015 [9]	Data-driven enjoyable carpool	Optimization model to minimize number of cars, greedy approach to maximize enjoyability

Table XII: Summary of Solution Approaches to the Shared Mobility Problems

Paper	Problem	Method
Hsieh, 2022 [38]	Trust on Shared-mobility Systems	Trust-awareness recommender system using self-adaptive neighborhood search with differential evolution
Martelli, 2021 [28]	Comparison of trip matching algorithms	Cardinality (maximize matched users), Saved Distance (maximize users shared distance), Time (to favor social integration), Proximity (physical proximity of users' trips) algorithms
Peng, 2021 [17]	Carpool Problem with semi-centralized ride-matching approach SCM	Combination of decomposition and network flow algorithms, using a combination of carpooling chance graph within a community, analysis of connections using CTA Carpooling Tree Generation Algorithm, and then apply a greedy algorithm to find an optimal solution with the goal to serve more riders with min vehicles
Alisoltani, 2022 [39]	Real-Time Ridesharing	Space-time clustering-based method
Bouzid, 2020 [16]	Long-Term Car Pooling	Bee Colony Optimization Algorithm
Aydin, 2020 [27]	Dynamic ride-sharing matching with route splits and social factors	Use of social compatibility score, split the unmatched route to match driver with another rider. Optimization model with objective functions to maximize social compatibility or saved distance, heuristic solution using both. Use of Needleman-Wunsch algorithm to check route feasibility. Dijkstra's algorithm to calculate shortest path.
Librino, 2019 [29]	Carpooling for Social Integration	4 different algorithms: Maximum Cardinality, Maximum Saved Distance, Maximum Mingling Time, Maximum Social Mixing
Sun, 2020 [40]	Peer-to-peer ridesharing	Consider static and dynamic versions of the problem. Use of exact solution approach
Yu, 2020 [41]	Trajectory-based route planning	Long-short Term Memory Graph Network LSTM-GN limiting the learning space and using a dynamic weighted loss function to strengthen the positive class
Santos., 2020 [22]	Carpooling Problem	Comparison of the most used ML Techniques: K-Nearest Neighbors - KNN (using Manhattan distance), K-means (using Euclidean distance), Decision Tree ID3, Bayesian Networks, Naïve Bayes, Artificial Neural Networks - NN
El Salhi, 2019 [23]	Carpooling App based on kNN	Drivers are evaluated using kNN and passengers receive a list of potential matches regarding their preferences, with rating and chat features
Hsieh, 2018 [21]	Carpooling Problem	Meta-heuristic approach using many variants of discrete differential evolution algorithms with binary decision variables to find a solution using bids.

Paper	Problem	Method
Kutiel, 2017 [19]	Maximum Group Carpool Matching Problem	Local Search Algorithm with similarity function. It analyses approximation algorithms, the impact of capacity constraint with mathematical proof for each variant
Masoud, 2017 [42]	Peer-to-peer multi-hop ridesharing	Decomposition algorithm, use pre-processing to reduce the size of the problem and iteratively solving the subproblems. Use of optimization and heuristic solutions
Tang., 2017 [43]	Dynamic Ridesharing - Data Science behind GrabShare service	Extract insights from data to improve passenger experience.
Schreieck, 2016 [30]	Dynamic Ridesharing Problem	Matching algorithm. Finds shortest path with Dijkstra's algorithm and save rider offers for each node using inverted index data structure
Xia, 2014 [18]	Carpool Problem for 3+ person	Use of optimal (linear solution, and brute force enumerated procedure), and Heuristic (simulated annealing, and tabu search) solutions.
Hussain, 2016 [44]	Long-Term Carpooling Problem	Organizational-based model and agent-based simulation
Boukhater, 2014 [24]	Long-Term Carpool Problem	Genetic Algorithm with a customized fitness function. Includes objective function with fairness, and costs for distance, time, and preferences (# users, smoker, blacklist)
Santi, 2014 [45]	Quantifying benefits of 'sharing' vehicles	Mathematical framework to compute the trade-off of collective benefits and passenger discomfort
Kaan, 2013 [46]	Vanpool assignment problem	One-stop and Two-stop models using Integer Linear Program to find optimal solution minimizing the total cost applying different heuristics
Nha, 2012 [20]	Comparison of vehicle's routing algorithms	Initiate route using shortest path and test algorithms' capacity of adapting to events using traffic simulator. Optimal (Dijkstra, Incremental Graph), Heuristic (A*, Tabu, GA, ANT colony) and Hybrid (Dijkstra + GA).
Yan, 2011 [25]	Pre-matching carpooling problem	Network-flow model using Lagrangian Relaxation to find close-optimal solution
Zidi, 2012 [47]	Dial-a-Ride Problem	Multi-objective simulated annealing
Brugliere, 2011 [26]	Carpooling for Universities	Guided Monte Carlo method with matching matrix and objective function.