MAKE AMERICAN TRANSPORTATION GREAT AGAIN:
AUTONOMOUS TAXI FLEET MANAGEMENT STRATEGIES

SHIRLEY ZHU
ADVISOR: PROFESSOR ALAIN L. KORNHAUSER

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN ENGINEERING
DEPARTMENT OF OPERATIONS RESEARCH AND FINANCIAL ENGINEERING
PRINCETON UNIVERSITY

MAY 2016
I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

______________________________
Shirley Zhu

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

______________________________
Shirley Zhu
Abstract

As autonomous vehicle technology is becoming more of a reality, we must start preparing for a future where autonomous vehicles are part of everyday life. Self-driving cars will change how people access mobility and change the mobility structures that exist today. With the mass adoption of self-driving cars, individual car ownership will become unnecessary and vehicles will be operated as fleets of autonomous taxis (aTaxis). aTaxis also bring incredible opportunities for ridesharing, which will decrease vehicle miles traveled, and in turn decrease vehicular congestion and environmental pollution. Managing fleets of aTaxis will be drastically different than traditional taxi fleet management.

Using a fleet of vehicles of varied sizes and a data set of synthesized travel demand for the state of New Jersey, this thesis analyzes the benefits of ridesharing for New Jersey and explores various fleet management strategies and the costs associated with these strategies. Ridesharing is able to increase average vehicle occupancy from 1 to 1.74 and reduces total vehicle miles traveled by 43%. Even in an upper bound case, the total number of vehicles needed to serve all of New Jersey’s travel demand is less than 50% of the number of vehicles currently on the road today in New Jersey, which would have significant benefits in terms of congestion and pollution. Additionally, the research in this thesis extends well to optimal fleet sizing for an aTaxi fleet.
Acknowledgements

First and foremost, my deepest gratitude goes to my advisor, Professor Alain Kornhauser, for all of his support and guidance over the past year. Prof. K., thank you for always being available and taking time to meet with me whenever I showed up to your office. Your enthusiasm and passion is infectious – I’ve learned so much from you over the past year and had a great time. This thesis would not have been possible without you.

Additionally, I would like to thank the generations of ORF467 classes before me, whose work made it possible to have the incredibly detailed data set that was used for this thesis. In particular, I would like to thank Hana Ku, not only for being my ORF467 partner for the project that led to this thesis, but also for being a great pset buddy and an amazing friend.

My Princeton experience has been marked by the incredible people that I’ve been fortunate enough to call my friends. My best friends on campus since day 1: Hannah, Danny, Ilya, Eric, thanks for making me laugh just about every day for the last four years. I couldn’t have asked for a better group of people to have spent these last four years with. Rachel Marek, I’m so glad that Charter gave us the opportunity to meet each other. Thank you being an wonderful friend and thesis fairy. You’re the best! To the original pset crew: Hana, Cailin Hong, and Eric Li, thank you for being a part of this journey with me. I can’t imagine what my ORFE experience would have been like without you guys.

Last, but most definitely not least, to my family — Mom, Dad, Alex, Vanessa — thank you for being so incredibly encouraging and supportive in all of my endeavors over the years. Mom and Dad, words cannot express my gratitude for all that you’ve done for me. Thank you for always believing in me and always pushing me to be a better student and a better person. I would not be where I am today without you. This is for you.
Contents

Abstract ........................................................................................................................................ iii
Acknowledgements ....................................................................................................................... iv
List of Tables .................................................................................................................................. vii
List of Figures ............................................................................................................................... viii

1 Introduction ................................................................................................................................ 1
  1.1 Purpose ...................................................................................................................................... 1
  1.2 Motivation .................................................................................................................................. 2
    1.2.1 Self-Driving Cars ................................................................................................................. 3
    1.2.2 New Car Ownership Patterns .............................................................................................. 4
    1.2.3 Ridesharing ......................................................................................................................... 6
  1.3 Existing Research ...................................................................................................................... 7

2 Data Background .......................................................................................................................... 9
  2.1 Data Generation ......................................................................................................................... 9
    2.1.1 Resident and Trip Synthesis ................................................................................................. 10
    2.1.2 Trip Mode Split .................................................................................................................... 11
    2.1.3 Pixelization .......................................................................................................................... 11
    2.1.4 aTaxi Operation .................................................................................................................. 12
    2.1.5 Ridesharing Trips ............................................................................................................... 13
  2.2 Data Contents ............................................................................................................................ 14
  2.3 Data Integrity ............................................................................................................................. 15

3 Trip Analysis ................................................................................................................................ 16
  3.1 Trips Taken ................................................................................................................................ 17
  3.2 Trip Length ................................................................................................................................ 21
    3.2.1 Average Vehicle Occupancy and Average Departure Occupancy .................................... 22
    3.2.2 Intercounty and Intracounty Trips ....................................................................................... 25
  3.3 Intermediate Stops ..................................................................................................................... 27
  3.4 Summary .................................................................................................................................... 29

4 Baseline Repositioning .................................................................................................................. 30
  4.1 Lower Bound - Infinitely Fast Repositioning ......................................................................... 30
  4.2 Upper Bound - Naive Repositioning ....................................................................................... 32
  4.3 Repositioning Cost .................................................................................................................... 33
    4.3.1 Linear Program .................................................................................................................... 33
    4.3.2 Gravity Model for Trip Distribution .................................................................................... 36
    4.3.3 Results ................................................................................................................................ 37
# 5 Local Repositioning

5.1 Simple Local Repositioning ........................................... 40
5.1.1 Strategy ................................................................. 41
5.1.2 Algorithm ............................................................... 42
5.1.3 Results ................................................................. 43
5.1.4 Costs ................................................................. 44
5.1.5 Initial Distribution of Vehicles ..................................... 45
5.2 Extended Vehicle Search ............................................... 47
5.2.1 Strategy ................................................................. 48
5.2.2 Results ................................................................. 48

# 6 Summary and Next Steps

6.1 Summary ................................................................. 52
6.2 Limitations ............................................................... 53
6.3 Next Steps ................................................................. 53
6.3.1 Optimal Fleet Sizing .................................................. 53
6.3.2 Improvements on Ridesharing Methodology ................. 56
6.3.3 Fleet Mix ............................................................... 57
6.3.4 County-Level Management ....................................... 58
6.3.5 Taxi Source Placements ............................................. 58

# A Selected R Code

A.1 Naive Repositioning Code ............................................ 61
A.2 Gravity Repositioning Code ......................................... 63
A.3 LP Repositioning Code ................................................ 66
A.4 Local Repositioning Code ........................................... 67
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Person Trips and Vehicle Trips originated statewide</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Average vehicle occupancy and average departure occupancy statewide</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Baseline Fleet Size Summary</td>
<td>33</td>
</tr>
<tr>
<td>4.2</td>
<td>Empty repositioning miles in naive case</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>Fleet size comparison for various repositioning strategies</td>
<td>42</td>
</tr>
<tr>
<td>5.2</td>
<td>Average wait time (in minutes) for passengers waiting longer than 5 minutes</td>
<td>49</td>
</tr>
<tr>
<td>5.3</td>
<td>Maximum wait time (in minutes) for passengers waiting longer than 5 minutes</td>
<td>49</td>
</tr>
<tr>
<td>5.4</td>
<td>Total wait time (in minutes) for passengers waiting longer than 5 minutes</td>
<td>50</td>
</tr>
<tr>
<td>6.1</td>
<td>Fleet operation costs</td>
<td>54</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Uber traffic volume in San Francisco before and after introduction of Uber-POOL .......................................................... 7

2.1 Pixelation Map .......................................................... 12

3.1 Person Trips originated by vehicle type in each county ............... 18
3.2 Vehicle Trips originated by vehicle type in each county ............... 19
3.3 Spatial distribution of person trips originated in each pixel ........... 20
3.4 Average Person Trip length of trips originated by vehicle type in each county 22
3.5 Average Vehicle Trip length of trips originated by vehicle type in each county 23
3.6 Average vehicle occupancy by vehicle type in each county .......... 24
3.7 Average departure occupancy by vehicle type in each county ...... 25
3.8 Percentage of Intracounty Person Trips originated by vehicle type in each county .......................................................... 26
3.9 Percentage of Intracounty Vehicle Trips originated by vehicle type in each county .......................................................... 27
3.10 Average number of intermediate stops by vehicle type in each county .......................................................... 28

4.1 Number of vehicles on the road by vehicle type .......................... 31
4.2 Visualization of pixel and super pixel .................................. 34
4.3 Two possible cases of super pixel to super pixel repositioning ....... 35
4.4 Naive strategy repositioning distances traveled, super pixel to super pixel, linear programming approach .............................................. 38
4.5 Naive strategy repositioning distances traveled, super pixel to super pixel, gravity model approach .............................................. 39

5.1 Pixels that can be reached from the departure pixel within 5 minutes .... 41
5.2 Voxelization of pixels in local repositioning ............................ 43
5.3 Local repositioning fleet usage by vehicle type, 50% case .......... 44
5.4 Percentage of unserved trips for various fleet sizes .................... 45
5.5 Empty miles comparison for various fleet sizes, simple local repositioning 46
5.6 Percentage of unserved trips with different initial vehicle distributions .... 47
5.7 Empty miles comparison for various fleet sizes, local repositioning with extended search .............................................. 51

6.1 Dominating Set .......................................................... 59
Chapter 1

Introduction

Transportation, at its core, is about improving an individual’s utility. An individual chooses to spend their time moving from one place to another because his or her utility can improve by moving. Technology facilitates transportation by making it faster, easier, and cheaper. Today, transportation is becoming more and more autonomous and new advancements in autonomous vehicle technology have made self-driving cars ever more of a reality.

1.1 Purpose

The goal of this thesis is to explore fleet management strategies for ridesharing autonomous taxis, or aTaxis, as a statewide operator in New Jersey. In this scenario, aTaxis, in conjunction with existing forms of public transportation, serve all of the personal mobility demand of New Jersey. Using a fleet of vehicles with varied passenger capacities, we explore ways to meet the travel demand of New Jersey. We analyze different fleet operation and repositioning strategies to see what effects this has on vehicle usage.

In this chapter, we provide an overview of self-driving cars, the need for ridesharing, and existing research. In Chapter 2, we give a more comprehensive overview of the data set used in this thesis, how the data was synthesized, and how the aTaxi system operates. Next, Chapter 3 analyzes overall trends in trips taken and provides an understanding of the trends in ridesharing trips. Chapter 4 creates a baseline of minimum and maximum
fleet sizes and costs needed to operate the aTaxi system needed to serve of New Jersey’s mobility demands. Chapter 5 looks at methods of locally repositioning vehicles during the day to reduce the fleet size from the maximum fleet size needed. Finally, Chapter 6 details next steps for further research.

1.2 Motivation

Since the mass production of automobiles by Henry Ford in the early 20th century, cars have transformed personal transportation, giving people on-demand transportation and independence. People were able to travel across large distances easily, reliably, and quickly, causing the personal automobile to become the dominant form of transportation in the United States.

However, in the over 100 years since the automobile was first introduced to the public, not much has changed fundamentally about automotive transportation. There have been many new safety, efficiency, and convenience features introduced, but the automobile still requires a human to manually operate it for the duration of the trip. Because of this, automobile accidents are still a major problem. Despite the improved safety features in modern cars, they are not the indestructible machines that we would like for them to be. According to the National Highway Traffic Safety Administration (2008),, though the number of fatalities due to motor vehicle accidents has been steadily decreasing each year, motor vehicle accidents are still the leading cause of death for 4-34 year olds in the US 1.

About 90% of road accidents are caused by human error 2 and even the most advanced safety features can only do so much to mitigate human error. Many of these new safety features, such as collision avoidance and automatic braking, involve autonomous detection of dangerous situations and reacting accordingly. More and more of these autonomous technologies are being incorporated into new cars. Fully autonomous vehicles would bring numerous benefits, including increased safety and efficiency. Furthermore, fully autonomous vehicles could significantly reduce, if not completely eliminate, auto accidents due to human error.

1KPMG & Center for Automotive Research (2012)
1.2.1 Self-Driving Cars

Autonomous vehicle technology, particularly self-driving cars, has advanced significantly in the past few years. However, fully autonomous vehicles, those that operate without any input from a human driver, are still far from ready for mass consumption.

Currently, the most notable self-driving car project is the Google Self-Driving Car. Google has been working on their Self-Driving Car Project since 2009, first adapting existing cars, specifically the Lexus RX and Toyota Prius, to drive autonomously. This involved retrofitting these cars with a laser remote sensing system, called LIDAR. These vehicles were able to drive fully autonomously, without any input from a human driver. After testing these cars extensively, Google developed their own fully autonomous car without a steering wheel, gas pedal, or brake pedal, prototypes of which started to be driven on public roads in 2015. As of February 2016, Google’s self-driving cars have logged almost 1.5 million miles in autonomous mode.\(^3\)

In 2015, the peer-to-peer ride-hailing company Uber partnered with Carnegie Mellon University to establish a research center focusing on autonomy technology and Delphi, an automotive supplier, drove a car from San Francisco, California to New York City 99% autonomously. The trip covered 3,400 miles and lasted 9 days\(^4\). Additionally, in late 2015, Tesla Motors introduced the “Autopilot” feature in its Model S electric cars. This was a software update that was capable of autonomously driving the car on highways as well as change lanes autonomously.

However, there are also challenges that self-driving cars still face. While Autopilot is able to drive the car without any input from the driver under certain conditions, the driver must be alert and ready to take over control at all times. Google’s self-driving cars navigate using a highly detailed and very accurate set of onboard 3D maps, which includes locations of road signs and traffic lights. This limits cars to roads which have been thoroughly mapped. Furthermore, it is extremely challenging to keep these maps up to date, as road features can change frequently. Additionally, they are not yet able to drive in inclement weather.

Beyond technological challenges, self-driving cars must also overcome trust issues and

---

\(^3\)Google Self-Driving Car Project Monthly Report, February 2016

\(^4\)“This Is Big: A Robo-Car Just Drove Across the Country,” Wired, April 3, 2015
legal hurdles before they are able to be brought to market. Initially, people may be skeptical of autonomous technology and find it hard to turn over full control of driving over to computers, as there is a large amount of uncertainty in the technology. Human driving is highly dependent on our intuition in making split second decisions. There are many people who do not yet trust computers and artificial intelligence to have the same intuition as humans in driving and are expecting accidents to happen. This is evidenced by the following news headlines when Google's self-driving car was involved in its first accident where the car bears some responsibility for the accident (previous accidents involving self-driving cars have been the fault of human drivers): “Google’s self-driving car finally at fault in accident,” 5 “Google admits its self driving car got it wrong,” 6 “Google’s self-driving car got into another accident - but this time it was the car’s fault.” 7

In addition, there are many legal and regulatory issues with self-driving cars that must still be addressed before self-driving cars are able to be widely used, particularly those of legal liability. Among the open issues are assigning responsibility for accidents involving self-driving cars or incidents where self-driving cars break road laws and licensing and training for drivers/passengers.

Yet, the benefits to self-driving cars are enormous. They give back valuable time to drivers, who are able to engage in other activities while moving from place to place. Self-driving cars also increase mobility for those who are unable to drive. This includes older adults, children and young adults who would not be able to get a driver’s license, as well as individuals who are visually or otherwise physically impaired. Being able to provide mobility to this group of people also give time back those who currently have to provide transportation for them.

### 1.2.2 New Car Ownership Patterns

Though mass adoption of self-driving cars is not yet imminent, it is worth considering how car ownership patterns may change with the mass availability of self-driving cars. Today, in the United States, cars are generally owned by individuals for private use. In

---

5 CNN Money, February 29, 2016
6 Daily Mail, March 14, 2016
7 Business Insider, March 1, 2016
a study by Sivak (2013), it was found that although car ownership has declined in recent years, we still had in the US, on average, 1.95 vehicles per household and 1.10 vehicles per driver in 2011.

In areas of the United States without comprehensive public transportation systems, i.e. outside of major cities, mobility is, for the most part, achieved through driving. For most people who live outside of major cities, personal car ownership is currently the most reliable and cost-efficient method of transportation. However, on a macro scale, this kind of car ownership is inefficient; cars tend to only be driven for short periods of time during the day and sit unused in parking lots for a majority of the day. This kind of car ownership model currently exists because cars require human operators. There would only be an opportunity for a car to be reused after, for example, a person drives to work in the morning, if the next person was at the workplace, or if there was a dedicated driver for the car, essentially making it a taxi.

With self-driving cars, this is no longer an issue. The car would be able to drive itself to the next person who needed a ride, which would allow for the car to be used more frequently during the day, increasing the car’s utilization ratio. Additionally, because this does not require a human driver, the cost of a ride in a self-driving car decreases dramatically compared to a ride in a traditional taxi, as there is no longer a human labor cost. Self-driving cars are able to operate on existing transportation infrastructure and though the technological costs are high now, as the technology becomes more mature, the technological costs will decrease as well. If the costs per ride decrease sufficiently with the introduction and on-demand availability of aTaxis, people who previously owned vehicles may find it is now more cost efficient to hail an aTaxi than to own a car. As this becomes the case, autonomous vehicles will be owned in aTaxi fleets by operators rather than individually owned.

Even today, we see that there is a large market for on-demand rides. Uber, which launched in 2009, and allowed users to instantly request a ride from drivers using their own cars nearby, was serving 2 million rides per day in 2015.\textsuperscript{8} As people eschew personal car ownership, it is likely that self-driving cars will be owned and operated in fleets.

\textsuperscript{8}“Uber is Even Bigger Than You Realize,” Fast Company, September 8, 2015
1.2.3 Ridesharing

Fleet ownership of self-driving cars also presents the opportunity to increase ridesharing — combining trips originating from similar locations and going to similar destinations in one vehicle instead of using multiple vehicles. There are fewer vehicles on the road and fewer vehicle miles traveled while serving the same number of trips, which would help to alleviate congestion, increase efficiency, and further decrease costs.

Ridesharing can be viewed as a horizontal elevator. In an elevator, a person or a group of people who are all going in the same direction, either up or down, share the same space as they are automatically transported in that direction. When the elevator reaches someone’s destination, they get off of the elevator, and perhaps other people get on. We can apply this methodology to vehicular travel, where groups of people going in the same direction are moved at the same time, rather than an individual person in a vehicle. In buildings, we generally do not see a private elevator for every individual, but this is essentially what we have with cars today.

Ridesharing is particularly significant as Americans are driving more and more — in 2015, Americans drove 3.15 trillion miles\(^9\). As the number of miles driven and the number of trips taken increases, the amount of vehicular congestion and environmental pollution also increases. The only way to reduce the number of vehicles on the road and in turn, the number of vehicular miles driven, is through ridesharing.

Uber has introduced this concept, called UberPOOL, in many cities, offering lower fares for riders willing to share their ride with people going in a similar direction. They found that, in fact, people were willing to use UberPOOL. The use of UberPOOL has helped to decrease traffic volume in high volume areas, shown in Figure 1.1.\(^10\)

Because ridesharing brings immense benefits and self-driving cars easily allow for ridesharing, the fleet operation in this thesis assumes ridesharing when possible.


\(^10\)From Travis Kalanick: Uber’s plan to get more people into fewer cars, TED Talk given February 2016
1.3 Existing Research

This thesis builds upon the work that has been done by Kornhauser & ORF467F15 (2016), which analyzed ridesharing opportunities for New Jersey. They also took a preliminary look at various repositioning strategies for an aTaxi fleet in New Jersey and attempted to develop a stochastic repositioning strategy by using New York City taxi trip data to forecast demand.

Fagnant & Kockelman (2015) analyzes dynamic ridesharing opportunities for Shared Autonomous Vehicles or SAVs, based on a synthetically generated set of trips for Austin, Texas, routed on road networks in Texas. A dynamic rideshare is one where new travelers can be added on the network as a vehicle is traveling to an existing passenger’s destination, provided that the new and existing passengers are not too inconvenienced. In their simulations, they found that dynamic ridesharing increased total shared miles up to 11%. They also determined an optimal fleet size of 2118 vehicles to serve the daily demand.

Pavone, Smith, Frazzoli, & Rus (2012) propose a robotic solution for vehicle rebalancing in mobility-on-demand systems. Their vehicular system consists of stations where customers and vehicles arrive and depart from. They model the system with stochastic arrivals of both vehicles and passenger trips, which determines the number of vehicles at each station. The
total number of vehicles in the system is constant, therefore, without rebalancing, the system is unstable and there will not be enough vehicles to serve all customers. In their rebalancing framework, they minimize the number of empty vehicles that need to be rebalanced and propose a methodology to optimally route the rebalance vehicles on the transportation network.
Chapter 2

Data Background

The data set used for this analysis is a synthetically generated set of trips taken in New Jersey by residents, as well as non-residents who work in New Jersey on an average weekday. The original trip data was created by Princeton University’s ORF 467: Transportation Systems Analysis Fall 2011 class, taught by Professor Alain Kornhauser. Subsequent sections of ORF 467 have built off of this data to create the data set used for analysis in this thesis.

2.1 Data Generation

The data that is the basis for this thesis was generated with the following steps, which closely follows the classic transportation planning process (Kornhauser & ORF467F11 (2012)):

- Generate a synthetic population of New Jersey with characteristics that closely resembles the actual population of New Jersey.
- Assign home, work, and school locations to the generated population.
- Assign trips to individuals
- Assign transportation modes to trips
- Analyze vehicle trips for ridesharing opportunities
2.1.1 Resident and Trip Synthesis

First, the 2010 census data was used to generate a population of individuals that closely resembles the population of New Jersey residents and non-residents working in New Jersey. Characteristics about each individual were also generated. This included age, gender, home location, occupation type — whether an individual was a in grade school, college, worked from home, commuted to work, lived in a nursing home, etc. Individuals were then grouped into households, again based on census information about household size. By creating individuals and households with these characteristics that reflect the true population of New Jersey, the travel demands can be better reflected in the generated data.

Next, based on the characteristics of the individuals, work or school locations were assigned. For workers, a work county and specific employer were assigned, as well as the start and end of the working day. School-aged children were probabilistically assigned to different schools and types of schools (public, private, homeschool, special) based on location, age, and school enrollment.

Once the home and work/school locations for each individual was assigned, trip tours were created for each individual. Trip tours describe the location, time, and purpose of the set of trips that an individual takes during the day. Trip tours contain exact latitude and longitude coordinates of trip origin and destination, as well as the departure and arrival times for each trip node. The times are assigned based on a probabilistic distribution of times that people arrive at and leave from work, based on their workday.

On an average weekday, an individual’s trips are assumed to start at home and end at home. For example, a resident of Princeton who works in Trenton would have a trip tour that is at least one trip to work from home and one trip home from work. A trip tour can also include additional trips between home and work, e.g. going to the grocery store after work.

Each trip is assumed to only have 1 person. The complete set of trips contains about 30.5 million individual trips with the average individual making 3.41 trips per day. The average trip length is 19.3 miles. (Kornhauser & ORF467F11 (2012))
2.1.2 Trip Mode Split

From the trip tours, modes of transportation were assigned for each individual trip. Based on length and destination, each trip is assigned to either: walk/bike, train, or aTaxi. Trips that are sufficiently short, approximately 1 mile or shorter in distance, are considered walk/bike trips. Trips originating or ending in New York or Philadelphia are satisfied by taking a New Jersey Transit train to the station closest to the New Jersey end of the trip. All remaining trips are served with aTaxis. (Kornhauser & ORF467F13 (2014))

2.1.3 Pixelization

For each trip, the locations are given by latitude and longitude coordinates. Because there are so many values that these coordinates can take, directly using the latitude and longitude coordinates in analyses would be too computationally intensive. In order to simplify computations, the state of New Jersey and external locations of interest are discretized. In this scenario, New Jersey is discretized by creating squares, or pixels, of 0.5 x 0.5 miles in dimension, with the bottom left corner, (0,0), being at (−75.6° E, 38.9° N), a point west and south of New Jersey. Though using this discretization method sacrifices the level of service available, i.e. it is not door-to-door service like Uber, the pixels are small enough such that walking within a pixel is reasonable. With an aTaxi stand at the center of each pixel, the longest walking distance is \(0.25\sqrt{2}\), or .35 miles. If we assume an average walk speed of 3 miles per hour, then the maximum walking time is about 7 minutes. As even Uber is not always perfectly door-to-door (passengers sometimes have to cross the street, walk to an intersection, etc), this is a reasonable discretization to use.

All latitude and longitude coordinates in the data set are converted to pixel numbers using the following formula (Kornhauser & ORF467F12 (2013)):

\[
X_{\text{coord}} = \text{floor}(108.907^*(\text{longitude} + 75.6))
\]

\[
Y_{\text{coord}} = \text{floor}(138.2^*(\text{latitude} − 38.9))
\]

Further analysis uses the pixel as the smallest geographical unit. The pixelation of New
Jersey can be seen in Figure 2.1a and the pixelation of Princeton, NJ can be seen in Figure 2.1b.

2.1.4 aTaxi Operation

The aTaxi system operates as follows: an aTaxi stand is placed at the center of each pixel. All aTaxi trips originate from and end at the aTaxi stand. Passengers will walk or bike to the stand from wherever they are in the particular pixel. Each aTaxi stand has the capacity to hold and dispatch as many vehicles as necessary.

For the purposes of this analysis, we assume that there is one aTaxi operator for the entire state. This means that regardless of where a vehicle originated from, it can be used
to fulfill a trip at its destination pixel. In a situation where there are multiple aTaxis
operators in the state, for example, by county, there may be situations where taxis that are
“owned” by one county may not pick up passengers in a different county and must return
to their home county empty. This is currently the case with taxis in many parts in and
around New Jersey. For example, a New York City taxi that takes a passenger to Newark
International Airport from New York is not allowed to pick up a passenger at Newark
Airport and must make an empty trip back to New York. Conversely, in Manhattan, only
yellow cabs are allowed to pick up passengers. This means that a Newark taxi taking a
passenger to Manhattan must make also make an empty trip back to Newark.

2.1.5 Ridesharing Trips

Once trip modes were assigned, a ridesharing analysis was conducted on the aTaxi trips.
When a passenger arrives at the aTaxi stand, they will wait up to a certain amount of time
(Departure Delay) before their vehicle departs. Their trip can also include people going to
the same destination pixel as them or people going to up to a certain number of destinations
(Common Destination) that are nearby or en route.

Various combinations of common destinations (CD = 1, 2, 3, 4) and Departure Delays
(DD = 1, 2, 3, 4, 5 minutes) were analyzed to determine what the effects of ridesharing
would be. CD represents the number of destinations that each departing aTaxi can stop
at and DD represents the amount of time that the aTaxi waits after the first passenger
arrives before departing. An additional constraint of 20% circuity is added, meaning that
the additional distance that any one passenger has to travel because of ridesharing cannot
be more than a 20% increase from the direct distance to their destination. Kornhauser &
ORF467F12 (2013) found decreasing marginal benefits in vehicle occupancy as CD and DD
increased. A complete ridesharing scenario for all synthesized aTaxi trips was constructed
for CD = 3 and DD = 5 minutes, or 300 seconds. This is the starting data set for this
thesis. Further details of the ridesharing methodology and implementation can be found in
Section 4.2 of Kornhauser & ORF467F12 (2013).

In the CD = 3, DD = 300, and max circuity = 20% scenario, each vehicle trip has no more
than three total destinations where passengers are being dropped off. The rideshare analysis
does not include picking up additional passengers after a vehicle has left its origin. This means that when a vehicle makes an intermediate stop, it does not pick up any passengers at that stop or reroute to pick up passengers that may be going in the same direction.

Furthermore, we assume average travel speeds of 30 miles per hour. This averages the speeds traveled on high speed motorways with speeds traveled on local roads, where a vehicles may be stopping more frequently. Based on this, new arrival times were assigned to the ridesharing trips, additionally assuming that there is no additional time required to discharge passengers any trip destination. Because we do not have the exact route that the vehicle takes, we approximate the distance traveled as \(1.2^*D_{cart}\), where \(D_{cart}\) is the cartesian distance between two points.

This data was then organized by county. In total, 10,479,382 ridesharing aTaxi trips were taken across New Jersey for \(CD = 3\), \(DD = 300\), and \(MaxCircuity = 20\%\).

### 2.2 Data Contents

The original rideshare trip files described above contained, for each vehicle trip taken:

- **oCounty**: Departure county
- **OriginX**: X pixel of departure county
- **OriginY**: Y pixel of departure county
- **DepartTime**: time of departure
- **TripNodeCount**: Number of destinations on the tour
- **Destination pixel and number of passengers on each leg of the trip**
- **TotalRiders**: sum of all passengers across all legs of trip
- **VehicleMiles**: Miles traveled by the vehicle
- **tripMiles** (which will be subsequently referred to as PersonMiles): Sum of the miles that each rider would have needed to travel if traveling individually. This is calculated as the sum of the distance from the origin to the destination of each passenger.
- **AVO**: Average Vehicle Occupancy - PersonMiles divided by TripMiles
• VehTourEnd: Destination county
• VehTourEnd_xPixel: X pixel of destination county
• VehTourEnd_yPixel: Y pixel of destination county
• VehTourEnd_time: Time of arrival at final destination. This is calculated using the assumption that on average, vehicles travel at a speed of 30 miles per hour.

2.3 Data Integrity

From the ridesharing trips, the first major task is to ensure that the ridesharing trips were constructed correctly – that is, all of the taxi trips that were synthesized were accounted for, all of the trips fell within the parameters, and trips times and distances were calculated correctly.

First, the total number of passengers and the total taxi trips synthesized were counted and were exactly equal: 30,125,587. This indicates that all of the trips were accounted for. Because PersonMiles is independent of ridesharing, the sum of PersonMiles calculated from the taxi trips should be the same as the sum of PersonMiles across all ridesharing trips. Both were calculated to be 475,051,588. A further breakdown and analysis of the trips can be seen in the next chapter.

The trip distances were verified to be calculated as $1.2^* D_{cart}$ and the trip time was calculated by assuming an average travel speed of 30 miles per hour.
Chapter 3

Trip Analysis

In our model, we assume that we have an operating fleet of vehicles of 4 sizes: 3 passenger, 6 passenger, 15 passenger, and 50 passenger. For simplicity, the number of passengers on the trip strictly dictates which vehicle they are assigned to, i.e. a trip with 5 people is always assigned to a 6 passenger vehicle, even if there are 3, 15, or 50 passenger vehicles available. For trips with more than 50 passengers, multiple 50 passenger vehicles are always used to fulfill the trip. Again, even if the excess number of passengers can be served with a different size vehicle or if there is a different size vehicle available, a 50 passenger car is used to serve the excess passengers.

We also make the assumption that the second day’s travel demand is exactly the same as the first day’s. Although this is not a particularly realistic assumption, because we are working with a synthetic data set of “typical” travel demand, based on probabilisitic distributions of characteristics and times, generating a second day’s worth of trip data would be very similar to the first day’s trip data, unless fundamental assumptions about the model are changed.

In this chapter, we provide an understanding of what the “typical” travel demand looks like in New Jersey, as well as the effects on ridesharing, broken down by county and vehicle type. For the purposes of the analyses in this chapter, the trips in each county refer to the trips that originate from that particular county. The following abbreviations are used for the counties in New Jersey:
• ATL - Atlantic County
• BER - Bergen County
• BUR - Burlington County
• CAM - Camden County
• CAP - Cape May County
• CUM - Cumberland County
• ESS - Essex County
• GLO - Gloucester County
• HUD - Hudson County
• HUN - Hunterdon County
• MER - Mercer County
• MID - Middlesex County
• MON - Monmouth County
• MOR - Morris County
• OCE - Ocean County
• PAS - Passaic County
• SAL - Salem county
• SOM - Somerset County
• SUS - Sussex County
• UNI - Union County
• WAR - Warren County

• EXT - External locations - includes New York City, NY; Bucks County, PA; Philadelphia, PA; Westchester County, NY; a cumulative point to the north of New Jersey; and a cumulative point to the south of New Jersey.

### 3.1 Trips Taken

We define a Person Trip as an individual’s trip — that is, each time an individual wants to go somewhere, that individual makes one Person Trip. A Vehicle Trip is one trip tour for a vehicle, from its departure pixel to its final destination pixel. A Vehicle Trip has 1 or more Person Trips, depending on the number of passengers in the vehicle. The number of
Person Trips per vehicle is the number of passengers in the vehicle at the time of departure. In the non-ridesharing case, the number of Vehicle Trips is equal to the number of Person Trips as each person takes their own car.

Table 3.1: Person Trips and Vehicle Trips originated statewide

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Trips</td>
<td>20,071,573</td>
<td>6,772,252</td>
<td>2,309,622</td>
<td>972,140</td>
<td>30,125,587</td>
</tr>
<tr>
<td>Vehicle Trips</td>
<td>8,659,171</td>
<td>1,526,478</td>
<td>257,149</td>
<td>36,584</td>
<td>10,479,382</td>
</tr>
</tbody>
</table>

In total, 30,125,587 Person Trips are taken in aTaxis across New Jersey. As expected, this is the same number of taxi trips synthesized. The majority of these trips are taken in 3 passenger vehicles, indicating that most trips have few ridesharing opportunities. The number of trips for which there are significant ridesharing opportunities, where there are
Figure 3.2: Vehicle Trips originated by vehicle type in each county

more than 15 combined trips, is much smaller. From the spatial distribution of Person Trips originated in each pixel, shown in Figure 3.3, we can see that the pixels with the most Person Trips originated are located near cities: New York City, Trenton, Atlantic City, and Philadelphia. We also see that while most pixels (95%) have fewer than 6,000 Person Trips per day, there are a few pixels that have significantly more Person Trips, which we can see from the large range in the number of trips for the top 1% of pixels.

Combined across all vehicles types, 10,479,382 Vehicle Trips were taken, with the vast majority in 3 passenger cars, which corresponds to what was seen in the number of Person Trips taken across the state. More Person Trips would require more vehicles to satisfy those trips. The number of Person Trips and Vehicle Trips taken statewide is summarized in Table 3.1. The breakdown of Person Trips and Vehicle Trips originated by county is shown in Figures 3.1 and 3.2, respectively. We see that on the county level, Person Trips
Figure 3.3: Spatial distribution of person trips originated in each pixel. The breaks represent the 10th, 25th, 50th, 75th, 95th, and 99th percentiles of the data. 95% of the pixels have fewer than 5,913 Person Trips originated.
and Vehicle Trips are also correlated, which is expected — more Person Trips also means more Vehicle Trips. For all vehicle types, Essex County had the either the highest or among the highest number of Person Trips. The pixel with the most Person Trips in all of New Jersey, which is Newark International Airport (Kornhauser & ORF467F12 (2013)), is in Essex County. Other counties with high numbers of Person Trips include Middlesex County, Hudson County, and Union County.

In larger vehicles, Essex County and Hudson County have significantly higher numbers of Person Trips than other counties. This shows that there is a much larger opportunity for ridesharing in these two counties, which is a result of total trip volume in these counties. As the volume of trips increases, the likelihood that there will be more than one person going to a particular location increases. Additionally, these counties are very close to New York City and it is likely that individuals who commute to and from work in New York City would be traveling to and from work at around the same time during the day, further increasing the opportunity for ridesharing.

### 3.2 Trip Length

By comparing the average length of both Person Trips and Vehicle Trips, we can get a sense of how efficient this particular ridesharing implementation is. In Figures 3.4 and 3.5, we have the average Person Trip length and average Vehicle Trip length of trips originating in each county. The longest average Person Trips and Vehicle Trips are from external locations, telling us that trips within New Jersey tend to be of shorter distances.

The shortest average Person Trips lengths were for trips originating in 50 passenger Vehicles. As the vehicle sizes increase, the average Person Trip length decreases. Taking Essex County as an example, the average Person Trip length for trips in 3 passenger vehicles is about 12 miles, in 6 passenger vehicles is 10 miles, in 15 passenger vehicles is 7 miles and in 50 passenger vehicles is 5 miles. From this, we can see that it is much easier to find ridesharing opportunities for passengers traveling short distances and that passengers traveling short distances tend to travel to similar locations.

However, there is an interesting phenomenon here: there is no significant change to the
average Vehicle Trip length as the vehicle size increases, even though the average Person Trip length is decreasing. In many cases, the average Vehicle Trip length is much longer than the average Person Trip length, especially for larger sized vehicles. This indicates that though most trips for the larger vehicles are short trips, there are large vehicles traveling long distances for a few passengers, which may not be the most efficient use of vehicles.

3.2.1 Average Vehicle Occupancy and Average Departure Occupancy

The efficiency of ridesharing can also be seen in the Average Vehicle Occupancy and Average Departure Occupancy. Average Vehicle Occupancy, or AVO, is defined as the Person Miles divided by the Vehicle Miles. This is one measure of how much vehicle travel can be reduced with ridesharing. Without ridesharing, the AVO, is 1, as each person takes a car alone. This is also the minimum AVO, not accounting for empty vehicle miles. When
there is ridesharing, the number of Person Miles does not change, but the number of Vehicle Miles is reduced, as people who are going in the same direction are sharing vehicles. This results in fewer vehicles being used to satisfy the number of trips and the same number of Person Miles. An AVO of 2 tells us that the number of miles traveled by aTaxis in ridesharing is reduced by one-half compared to the miles traveled by aTaxis without ridesharing.

Average Departure Occupancy, or ADO, is the number of people in the vehicle when it departs. The ADO is found by dividing Person Trips by Vehicle trips. This is one measure of how efficiently different sized vehicles are being used. The ADO is correlated with the vehicle type and tells us how often we are filling vehicles to capacity.

In Table 3.2, we see the AVO and ADO for different vehicles sizes statewide. Overall, the AVO statewide is 1.74, telling us that this ridesharing implementation decreased the vehicle miles traveled by about 43%. The highest AVO, 8.15, achieved was in the 50 passenger case,
Figure 3.6: Average vehicle occupancy by vehicle type in each county

where the vehicle miles traveled was reduced over 85%.

The ADO for the 50 passenger case is 23.45, which is somewhat lower than expected. Through the way that the trips are assigned to vehicles, each 50 passenger vehicle trip must have at least 16 passengers. This means that a majority of the trips taken by 50 passenger vehicles depart with the vehicle less than half full. Compared to the 3 and 6 passenger vehicles, which depart with the vehicles nearly full for each trip, this is a very inefficient use of the larger vehicle.

Table 3.2: Average vehicle occupancy and average departure occupancy statewide

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVO</td>
<td>1.59</td>
<td>2.38</td>
<td>3.59</td>
<td>8.15</td>
<td>1.74</td>
</tr>
<tr>
<td>ADO</td>
<td>2.32</td>
<td>4.44</td>
<td>8.98</td>
<td>26.57</td>
<td>2.87</td>
</tr>
</tbody>
</table>
Figure 3.7: Average departure occupancy by vehicle type in each county

On a county level, we see that the AVO and ADO for 3, 6, and 15 passenger vehicles do not vary significantly across different counties, whereas for 50 passenger vehicles, there is a large variation in the AVO and some variation in the ADO in different counties (Figures 3.6 and 3.7). Interestingly, when we compare the AVO and ADO for 50 passenger vehicles, there does not seem to be a correlation between AVO and ADO. For example, Warren County, which has the highest AVO of 17.20, has a ADO of 24.11, which is only slightly above average.

3.2.2 Intercounty and Intracounty Trips

Another interesting aspect of the ridesharing trips is the percentage of trips that are taken intracounty, within a county, compared to the trips intercounty, or across different counties. Intracounty trips are typically shorter compared to intercounty trips. Looking
Figure 3.8: Percentage of Intracounty Person Trips originated by vehicle type in each county at the breakdown of intracounty vs. intercounty trips, we can attempt to understand why the average Vehicle Trip lengths across various vehicle types are much larger than the corresponding average Person Trip lengths. The percentages of intracounty Person Trips and intracounty Vehicle Trips are shown in Figures 3.8 and 3.9, respectively.

Particularly in the 15 and 50 passenger vehicle cases, the percentage of intracounty Person Trips is much higher than the percentage of intracounty Vehicle Trips. For example, in Atlantic County, 95% of 50 passenger Person Trips are intracounty, but only 60% of 50 passenger Vehicle Trips are intracounty. This indicates that there are many trips where most passengers are traveling short distances but the ridesharing algorithm included a small number of passengers who are traveling much further. This is a very inefficient use of larger vehicles as larger vehicles have higher operating costs and consume more fuel. It would be more cost efficient to serve the few passengers who are traveling farther with a smaller
Figure 3.9: Percentage of Intracounty Vehicle Trips originated by vehicle type in each county

vehicle, even if there was no ridesharing opportunity.

In a future iteration of the ridesharing assignments, the algorithm could consider these cases and use smaller vehicles to serve the trips that are farther away with few passengers. Although this is not a perfect or the most efficient ridesharing assignment, this thesis will continue to use this data set for further analyses, as this is a possible scenario for ridesharing.

3.3 Intermediate Stops

Though ridesharing brings many large scale benefits, it does add an additional inconvenience to the travelers. Travelers now have to share a vehicle with other people and would potentially have to take a longer journey. Looking at the number of intermediate stops made by each vehicle, shown in Figure 3.10, which is the number of stops between the origin and
the final destination of the vehicle, we can see how much people are being inconvenienced. Since the ridesharing was created with a maximum number of total destinations as 3, the maximum number of intermediate stops is 2.

We see that for 6, 15, and 50 passenger vehicles, the average number of intermediate stops is close to 2 in almost every county. The highest average intermediate stops occur for trips that originate in external locations, indicating that people going to New Jersey from these external locations almost always go to similar places in New Jersey at similar times. From what we know about the generated data, this makes sense, as the people coming to New Jersey from out of state are commuters who live in New Jersey and work outside of New Jersey or vice versa. Additionally, travelers from New York City and Philadelphia, in this scenario, always take the train into New Jersey, so there are much higher opportunities for ridesharing.
For 3 passenger vehicles, the average number of intermediate stops varies across counties. Again, we see that the average intermediate stops is highest for external locations. We also see that in the counties where the Person Trip volume was higher, the number of intermediate stops also tends to be greater, as more trips means more potential opportunities for ridesharing.

3.4 Summary

Through the various above analyses, we see that ridesharing has significantly decreased the amount of Vehicle Trips that are taken statewide. Vehicle miles traveled were reduced by over 40% and the number of Vehicle Trips taken was reduced by about a third. However, there are still inefficiencies in the way that ridesharing was implemented, which can be improved in a future re-implementation of ridesharing.
Chapter 4

Baseline Repositioning

Because an aTaxi system would operate with a fixed number of vehicles, it is at times necessary to move vehicles empty so that they can serve more trips, as they may not be needed at their destination. In some cases, vehicles can be used for a new trip departing from the pixel at which they arrive. However, this is not always the case, and the unused vehicles should be repositioned to pixels where they can serve another trip. In addition, moving vehicles empty allow us to serve more trips with fewer vehicles as opposed to purchasing a new vehicle each time there is a new trip, which is both impractical and infeasible. We refer to the relocation of vehicles without passengers as empty vehicle repositioning.

Empty vehicle repositioning incurs a cost, which we will measure by the number of miles that the vehicle travels empty. In order to compare the results of various empty vehicle repositioning strategies, we establish the upper and lower bounds for the fleet size given the trips that must be satisfied. The upper bound is established by a “naive” strategy, where vehicles are not moved empty during the day and all repositioned once a day. The lower bound is established by the assumption that vehicles can move infinitely fast. We assume that we have unlimited access to a “super source” of vehicles of all sizes.

4.1 Lower Bound - Infinitely Fast Repositioning

To establish the lower bound, we make the assumption that vehicles can be moved infinitely fast when they drop off their final passengers. As soon as a vehicle drops off
its final passenger, it is instantly moved to whatever pixel needs a vehicle for a departure and used there. Under this assumption, the fleet size needed is the maximum number of vehicles on the road at any given point during the day. Because we have discretized time by minutes, the minimum fleet size is the maximum vehicles on the road at any given minute during the day, which is shown in Figure 4.1. Note that for 15 and 50 passenger vehicles, the peak is much larger than the next largest number of vehicles on the road. This means that a nontrivial number of vehicles that must be purchased to satisfy this peak demand would be unused for the rest of the day.

This lower bound establishes the smallest fleet size necessary to serve all of the demand and provides a benchmark for comparison of the various empty vehicle repositioning strate-
gies that will be used. However, in this case, because we simply assume that when a trip is about to depart, a vehicle instantly arrives from some other place where there was an available vehicle, we do not know what the repositioning cost, or empty miles traveled, will be.

4.2 Upper Bound - Naive Repositioning

In the naive repositioning strategy, empty vehicles are moved only once per day. In this analysis, we start out with enough vehicles at each pixel such that all trips are able to be served. At midnight, empty vehicles are moved so that we are able to run the system again without adding more vehicles. Since we are assuming each day is the same in terms of trip demand, if we do not reposition, then we will need to add the same number of additional vehicles to each pixel each day. We refer to the repositioning of empty vehicles in the entire system as Early Morning Repositioning (EMR). While repositioning, we also assume that the vehicles are able to move infinitely fast to their destination pixels.

This strategy is implemented in the same way as Ku & Zhu’s repositioning strategy in the section “Simple aTaxi Management Strategies” of Kornhauser & ORF467F15 (2016). During the day, trips departing from a pixel are satisfied by either vehicle at the pixel or one that was brought to the pixel from the super source. Each arrival at a pixel increases the supply at the pixel and each departure decreases the supply. If there is a departure for which there is no available supply, a vehicle is brought from the super source. On day one, we assume that it can be brought to the pixel infinitely fast. For each subsequent day, we know how many vehicles are needed at each pixel at the beginning of the day and can position the fleet accordingly. This is because we assume the trip demand is the same each day, and the entire state is a closed system, so the number of vehicles that we need at each pixel at the beginning of the day will always be the same. The minimum fleet size needed to operate the system is the number of vehicles that needed to be brought from the super source during the day.

A summary of the upper and lower bounds of the fleet sizes needed to operate the system is shown in Table 4.1.
Table 4.1: Baseline Fleet Size Summary

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Trips Served</td>
<td>8,659,171</td>
<td>1,526,478</td>
<td>257,149</td>
<td>36,584</td>
</tr>
<tr>
<td>Active Pixels</td>
<td>21,641</td>
<td>19,231</td>
<td>15,097</td>
<td>8,344</td>
</tr>
<tr>
<td>Lower Bound - Infinite</td>
<td>632,947</td>
<td>133,275</td>
<td>43,957</td>
<td>11,467</td>
</tr>
<tr>
<td>Upper Bound - Naive</td>
<td>2,425,673</td>
<td>621,132</td>
<td>154,996</td>
<td>30,295</td>
</tr>
</tbody>
</table>

4.3 Repositioning Cost

At the end of the day, not all vehicles will end up perfectly at pixels such that there is exactly enough to satisfy the next day’s demand. Therefore, vehicles need to be moved so that there are the correct number of vehicles at each pixel to satisfy all of the demand for the next day. We do not know the repositioning cost in the infinitely fast repositioning case, so we will only calculate results for the naive repositioning case.

In repositioning the system, we want to minimize the total number of empty vehicle miles. We use two methods to find the minimum cost of repositioning the vehicles: the gravity model for trip distribution and a network flow linear program. In both models, the cost is the distance that a vehicle must travel while empty. As previously, we model the driven distance by taking $1.2 \times D_{cart}$.

4.3.1 Linear Program

The repositioning problem can be solved using the classic transportation linear program. We can model the linear program as follows:

$$\min \sum_{i \in I} \sum_{j \in J} D_{ij} T_{ij}$$

subject to $T \geq 0$

$$T_{i,i} = 0$$

$$\sum_{i \in I} T_{ij} = A_j, \quad \forall j \in J$$

$$\sum_{j \in J} T_{ij} = P_i, \quad \forall i \in I$$
where:

- $D_{ij}$ is the distance between pixel $i$ and pixel $j$
- $T_{ij}$ is trip matrix, or the number of vehicles moved from pixel $i$ to pixel $j$, $T$ is zero on its diagonal
- $\mathcal{I}$ and $\mathcal{J}$ are the set of active pixels
- $P_i$ is the number of excess vehicles available at the pixel $i$
- $A_j$ is the number of vehicles needed at pixel $j$

In this LP, we are minimizing the total distance traveled with the constraint that the number of vehicles moved from a pixel is equal to the number of vehicles at a pixel, and the number of vehicles moved to a pixel is equal to the number of vehicles needed at a pixel. Since we are only looking at excess and deficit vehicles at any pixel, all excess vehicles should be moved. Therefore, $\forall i, T_{ii} = 0$.

Even though we have already discretized New Jersey into pixels, optimizing over all pixels still results in an extremely large optimization problem. For the smallest case, 50 passenger vehicles, there are about $n = 8,000$ active pixels, which means there are roughly 6.5 million variables to optimize over, as $T$ is an $n \times n$ matrix. Because of this, we condense the problem by creating larger pixel blocks, termed “super pixels.” Super pixels are 3 x 3 blocks of the standard 0.5 x 0.5 mile pixels, shown in Figure 4.2. Starting at pixel (0,0), pixels are grouped into blocks of 9 pixels. The super pixel containing pixel (0,0) has its center at pixel (1,1). The super pixel center for any given pixel, $(x, y)$ is calculated as:

$$(\lfloor x/3 \rfloor + 1, \lfloor y/3 \rfloor + 1)$$

The supply or demand at each super pixel is simply the sum of the supply or demand
of the smaller pixels within the super pixel. Distances between super pixels are calculated from the center of each super pixel. This is able to greatly reduce the dimensionality of the linear program in the largest case from $n \approx 20,000$ to $n \approx 3,000$, which can be solved in reasonable time using an LP solver.

Though we are not considering the distances that vehicles have to travel to and from the center of the super pixel, we would have cases where vehicles travel both more than in a pixel to pixel repositioning and less than in a pixel to pixel repositioning. For example, in Figure 4.3a, we have a case where the super pixel to super pixel repositioning would be more than the pixel to pixel repositioning, and vice versa in Figure 4.3b. These cases would average out, making super pixel to super pixel repositioning is a good approximation for pixel to pixel repositioning.

The linear program was solved using the Gurobi solver for R, formulated in standard form as follows:

\[
\min_{x \in \mathbb{R}^{n^2}} \ c^T x \\
subject \ to \quad Ax = b \\
x \geq 0
\]

where:

- $c$ is the distance matrix, $D$ from LP 4.1, vectorized by row, i.e. the first $n$ entries of $c$ is the first row of $D$, the next $n$ entries are the second row of $D$, etc.
- $x$ is the trip matrix, $T$ from LP 4.1, vectorized in the same way as $c$
- $A \in \mathbb{R}^{2n \times n^2}$ is a sparse matrix that represents the constraint that the vehicles moved to and from a pixel must be equal to the supply and demand, respectively. In the first $n$
rows of $A$, each row $i$ has nonzero entries between $n_i$ and $n(i + 1) - 1$ where row $i$ of matrix $T$ was nonzero. In the second $n$ rows of $A$, each row $i$ has nonzero entries between $n_i$ and $n(i + 1) - 1$ where column $i$ of matrix $T$ was nonzero. For $n = 3$, the $A$ matrix is:

$$A = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

### 4.3.2 Gravity Model for Trip Distribution

The gravity model for trip distribution was developed by analogy to Newton’s law of gravitational attraction between two objects separated by a distance. It is used in the classic transportation planning model to determine trip distributions. The model creates a trip distribution matrix from a vector of trips produced in a set of activity zones and a vector of trips that are attracted to a set of activity zones, where activity zones can be residential, recreation, industrial, commercial, etc. A trip distribution matrix tells us how many trips there are between each pair of zones. The number of trips between zones based on the distance that the zones are from each other. The total number of trips produced and the total number of trips attracted are equal. More details about the gravity model can be found in Wilson (1967) and Calibrating and Testing a Gravity Model for Any Size Urban Area (1983).

Each index of the trip matrix, $T$, is given by:

$$T_{ij} = \frac{A_j F_{ij} K_{ij}}{\sum_{x \in \text{zones}} A_x F_{ix} K_{ix} P_i}$$

where:

$P$ is the production vector
$A$ is the attraction vector

$F$ is a matrix that represents the disutility of traveling between zones $i$ and $j$. The disutility is taken to be the inverse of the squared distance between $i$ and $j$ ($F_{ij} = \frac{1}{D_{ij}}$)

$K$ is a socioeconomic adjustment factor, which we did not include in this analysis.

$T$ can be found iteratively, as described by Wilson (1967). Our repositioning problem can be easily fit to this model. The production vector is the excess supply at each pixel, the attraction vector is the demand, or addition vehicles needed at each pixel, and our disutility matrix can be calculated from the distances between two pixels.

We attempt to use the gravity model for EMR repositioning as we can run the gravity model for larger sets of pixels than the LP and would be able to perform pixel to pixel repositioning. However, we found that there is a significant discrepancy between the gravity repositioning results and LP repositioning results.

### 4.3.3 Results

The number of empty repositioning miles for each vehicle type found using both the LP and gravity model is shown in Table 4.2.

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Program</td>
<td>16,039,770</td>
<td>5,515,587</td>
<td>854,912</td>
<td>204,821</td>
</tr>
<tr>
<td>Gravity Model</td>
<td>31,596,455</td>
<td>8,890,276</td>
<td>1,514,557</td>
<td>335,491</td>
</tr>
</tbody>
</table>

We see a significant difference in the number of repositioning miles obtained through each method, even though the number of vehicles repositioned in each case is equal. Looking at the distribution of travel distance for both methods (Figures 4.4 and 4.5), we see that while the maximum repositioning distance for the LP is at most 70 miles, the maximum repositioning distance for gravity is over 200 miles. Because of this, the total number of empty repositioning miles for the gravity model will be much higher.

Intuitively, the LP has a clear objective and is trying to minimize total miles traveled, so the solution will avoid trips with very long distances when possible. However, in the gravity model, there is no explicit objective to minimize the distance. The trips with longer
(a) 3 passenger vehicles
(b) 6 passenger vehicles
(c) 15 passenger vehicles
(d) 50 passenger vehicles

Figure 4.4: Naive strategy repositioning distances traveled, super pixel to super pixel, linear programming approach

distances, while unfavored, are not strictly excluded. The gravity model will assign more trips where the distance is smaller, but it also allows for longer trips. Relating this back to the transportation planning model, this assumes that while traveling long distances does give significant disutility, there are still people who will make these trips.

As we want to find minimum repositioning costs, the LP is a more accurate representation of the empty repositioning miles that we are interested in analyzing. We will be using the LP for super pixel to super pixel to solve the repositioning problem for the remainder of the analyses.
Figure 4.5: Naive strategy repositioning distances traveled, super pixel to super pixel, gravity model approach
Chapter 5

Local Repositioning

In the naive strategy, where cars are only moved empty at one point during the day, it is possible that there are vehicles at a pixel which are not being used for long periods during the day. To try and use vehicles more efficiently, a local repositioning strategy is implemented, where vehicles are repositioned short distances during the day in order to increase fleet usage. We look at two local repositioning strategies: a simple strategy looking only at pixels within 5 minutes of the departure pixel and an extended search strategy, looking at pixels that are may be farther away.

5.1 Simple Local Repositioning

In the local repositioning strategy, we attempt to increase vehicle usage and decrease the total number of vehicles needed by looking at nearby pixels for available vehicles instead of only at the pixel of departure. With a Departure Delay of 5 minutes, a passenger who arrives at an aTaxi stand will wait for up to 5 minutes for the vehicle to depart. This means that for any particular departure, we can look at the arrivals at nearby pixels up to 5 minutes prior to the departure time and see if there is an available vehicle. Again, assuming travel speeds of 30 miles per hour, we can find the pixels where a vehicle could travel from that pixel to the departure pixel before the trip departure.

Figure 5.1 shows the pixels that are within 5 minutes of driving time from the departure pixel. The departure pixel is in the center and the number in each pixel is the amount of
Figure 5.1: Pixels that can be reached from the departure pixel within 5 minutes
time that it takes to drive from that pixel to the departure pixel.

5.1.1 Strategy

The major assumption made in this scenario is that when the first passenger arrives,
the operator knows what type of vehicle to look for. In reality, this is difficult to predict,
as the operator would not know that a larger vehicle is needed until the requisite number
of passengers going in a similar direction arrive, which is not always at the same time as
the first passenger.

In this simulation, we look at how this system behaves with varying fleet sizes in between
the minimum fleet size and the naive strategy fleet size. The cases studied here are a fleet
size of 10%, 20%, 30%, 40%, and 50% in between the minimum naive size and the naive
strategy fleet size. We choose an initial distribution of vehicles to be proportional to the
Naive strategy initial distribution. The initial number of vehicles at each pixel for each
vehicle type is calculated as:

\[
\text{ceiling} \left( V_i \left[ \frac{FS_{\text{min}} + x\% (FS_{\text{naive}} - FS_{\text{min}})}{FS_{\text{naive}}} \right] \right)
\]

where

\( V_i \) is the naive strategy initial distribution at pixel \( i \),

\( FS_{\text{min}} \) is the minimum fleet size assuming infinite repositioning,
\( FS_{\text{naive}} \) is the naive strategy fleet size,

\[ x \in \{10, 20, 30, 40, 50\}. \]

We take the ceiling to ensure that all initial distributions are integer values, which means that the fleet size is not necessarily exactly 10%, 20%, 30%, 40%, and 50% in between the minimum and naive strategy fleet sizes. The fleet sizes of each scenario is shown in Table 5.1.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower Bound - Infinite</td>
<td>632,947</td>
<td>133,275</td>
<td>43,957</td>
<td>11,467</td>
</tr>
<tr>
<td>Local, 10% between</td>
<td>805,189</td>
<td>189,366</td>
<td>60,202</td>
<td>15,025</td>
</tr>
<tr>
<td>Local, 20% between</td>
<td>965,935</td>
<td>237,988</td>
<td>70,659</td>
<td>17,696</td>
</tr>
<tr>
<td>Local, 30% between</td>
<td>1,125,877</td>
<td>286,516</td>
<td>80,126</td>
<td>18,986</td>
</tr>
<tr>
<td>Local, 40% between</td>
<td>1,288,632</td>
<td>336,156</td>
<td>92,948</td>
<td>20,748</td>
</tr>
<tr>
<td>Local, 50% between</td>
<td>1,449,759</td>
<td>384,461</td>
<td>103,913</td>
<td>22,686</td>
</tr>
<tr>
<td>Upper Bound - Naive</td>
<td>2,425,673</td>
<td>621,132</td>
<td>154,996</td>
<td>30,295</td>
</tr>
</tbody>
</table>

5.1.2 Algorithm

First, a supply array is initialized with an initial distribution of vehicles as described above. Then, each trip, ordered by departure time, is analyzed. If there is an available vehicle at the departure pixel, it is used and the supply is decreased at the departure pixel and increased at the arrival pixel of that trip. If there is not an available vehicle at the departure pixel, then vehicle supplies at nearby pixels are checked, in order of time that it takes to reach the departure pixel. If an available vehicle is found at any of these pixels, then the supply is decreased at the pixel where the vehicle was found and increased at the destination pixel. If a vehicle was not found after looking at the pixels that are within 5 minutes travel time of the departure pixel, then that trip is not able to be served.

In the naive strategy, we were able to analyze each pixel individually for the entire day. The vehicle supply and trips taken at any pixel had no effect on other pixels. However, this is not the case here. Instead of looking at individual pixels, we are analyzing voxels of pixels, with the top voxel being the departure pixel at the time of departure and lower voxels corresponding to pixels at various times before departure. The voxelization, which is taking Figure 5.1 and adding a time dimension, is visualized in Figure 5.2.
5.1.3 Results

The fleet usage — the number of trips served with vehicles from each particular layer of pixels — for the 50% case is shown in Figure 5.3. For the other fleet sizes, the fleet usage is similar. A large majority of trips are served with vehicles that are at the pixel, either positioned there in the morning or vehicles that ended their trips at the pixel. The second largest group of trips is from the first layer of pixels — those that can be reached in one minute.

In both baseline strategies, naive and infinitely fast repositioning, all trips are served. With the local repositioning strategy, not all trips are able to be served. Unlike the naive strategy, where the available number of vehicles at a particular pixel only changes based on arrivals and departures at that pixel, in the local repositioning strategy, the available number of vehicles at a pixel can fluctuate based on the supply and demand at nearby pixels. Therefore, using a fleet size smaller than the naive strategy minimum fleet size results in trips for which there are no vehicles nearby to satisfy the trip. As the fleet size decreases, the amount of trips that are unserved increases, as seen in Figure 5.4.

Figure 5.2: Voxelization of pixels in local repositioning
5.1.4 Costs

This strategy also requires that the fleet be repositioned at the end of the day, similar to the naive strategy. The cost of the local repositioning strategy then, in terms of empty miles, is the empty miles traveled in the local repositioning and the empty miles traveled in the naive repositioning. Figure 5.5 compares the empty miles traveled in the various local repositioning cases to the naive case, where there is only EMR repositioning.

In all cases, the total number of empty repositioning miles is less than that of the naive strategy. As the fleet size increases, the EMR empty miles increase as well. Since there are now more vehicles in the system, there are more vehicles that need to be repositioned. Not surprisingly, as the number of vehicles decreases, the amount of local empty miles increases.
As there are fewer vehicles available, operators need to look farther from their pixel to find an available vehicle.

### 5.1.5 Initial Distribution of Vehicles

In this strategy, the initial distribution was selected based on a proportion of the naive repositioning strategy distribution. Because the supply of vehicles at a pixel fluctuates depending on trips departing near it, as well as trips that arrive near it, it is possible that regardless of the initial distribution, we end up with a similar usage during the day. If this was the case, since initial distribution does not matter, we would not need to do EMR, which would significantly reduce the empty miles traveled.
Figure 5.5: Empty miles comparison for various fleet sizes, simple local repositioning

We rerun the simple local repositioning strategy again for the 10% and 50% cases with the initial distribution being the end of day distribution from the first run. Intuitively, this case is when the operator does not do EMR, but instead just uses the vehicles that are remaining at each pixel for the next day. In Figure 5.6, we compare the percentage of unserved trips in each case. For all vehicles sizes, using a different distribution significantly increases the percentage of unserved trips. From this, we can conclude that the initial distribution does matter, and while the proportional naive repositioning distribution may not be the optimal distribution, it is a good distribution that allows us to have fairly low percentages of unserved trips.
5.2 Extended Vehicle Search

Previously, we have provided a level of service where passengers wait no longer than 5 minutes after arrival at the aTaxi stand before departure. If we could not provide this level of service, then these passengers were just simply not served. However, this kind of service is unrealistic and renders us unable to fulfill all of New Jersey’s mobility needs. Instead, we would like to see what happens if we ask passengers to wait longer than 5 minutes. In this simulation, we attempt to use local repositioning to serve all passengers with the given fleet size. We do not reject passengers for whom we cannot find a vehicle for in the first five minutes, but instead look farther and farther away until we do find an available vehicle.
5.2.1 Strategy

In the extended search strategy, we follow the simple local reposition strategy almost exactly. When a passenger arrives, we first attempt to serve the trip using vehicles available at the departure pixel. If there are no available vehicles at the departure pixel, we look first at pixels 1 minute away at 1 minute before departure. Then we look at pixels that can be reached within 2 minutes at 2 minutes before departure, continuing to five minutes before departure.

However, unlike the simple local repositioning strategy, if a vehicle is not found at pixel that can be reached within 5 minutes before departure, we look at pixels that are farther away at 5 minutes before departure, when the first passenger arrives at the aTaxi stand. This means that the passenger will have to wait longer than 5 minutes before they depart. The arrival time at the destination pixel is adjusted accordingly. We do not take into account that there may be another vehicle arrival at or closer to the departure pixel while the vehicle found is traveling to the departure pixel. The passengers simply wait until the vehicle found for them arrive. Additionally, it is possible that the passengers will find ridesharing opportunities on other trips as they are waiting. This, then, represents the maximum wait times possible, as a different implementation where priority of vehicles goes to passengers who wait the longest would result in decreased wait times.

As we did for the simple local repositioning strategy, we analyze this scenario for fleet sizes of 10%, 20%, 30%, 40%, and 50% between the minimum fleet size and the naive repositioning fleet size.

5.2.2 Results

In this strategy, we have that all trips are served, though some passengers may have to wait more than the Departure Delay of 5 minutes. The wait time is considered to be the amount of time passengers must wait for a vehicle to arrive after the 5 minute Departure Delay has passed. The average and maximum wait times for each case is shown in Tables 5.2 and 5.3. The total wait time for all passengers is shown in Table 5.4. In general, as the fleet size increases, the wait times decrease, though for 50 passenger vehicles, it seems that there
is no significant difference among the average wait times for different size fleets. The best case was with a fleet size of 50% between the minimum and naive fleet sizes. This resulted in the lowest maximum wait times for all vehicle types and the lowest average wait times for all vehicle types except for 50 passenger vehicles. For 50 passenger vehicles, the lowest wait time was achieved for a fleet size of 20% between the minimum and naive sizes.

Table 5.2: Average wait time (in minutes) for passengers waiting longer than 5 minutes

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local, 10%</td>
<td>18.22</td>
<td>10.24</td>
<td>8.87</td>
<td>7.28</td>
</tr>
<tr>
<td>Local, 20%</td>
<td>10.64</td>
<td>8.55</td>
<td>6.83</td>
<td>6.09</td>
</tr>
<tr>
<td>Local, 30%</td>
<td>8.52</td>
<td>6.99</td>
<td>5.97</td>
<td>6.12</td>
</tr>
<tr>
<td>Local, 40%</td>
<td>6.88</td>
<td>5.69</td>
<td>5.41</td>
<td>6.28</td>
</tr>
<tr>
<td>Local, 50%</td>
<td>5.56</td>
<td>4.69</td>
<td>5.18</td>
<td>6.66</td>
</tr>
</tbody>
</table>

Table 5.3: Maximum wait time (in minutes) for passengers waiting longer than 5 minutes

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local, 10%</td>
<td>195</td>
<td>56</td>
<td>98</td>
<td>76</td>
</tr>
<tr>
<td>Local, 20%</td>
<td>57</td>
<td>42</td>
<td>39</td>
<td>29</td>
</tr>
<tr>
<td>Local, 30%</td>
<td>48</td>
<td>38</td>
<td>36</td>
<td>27</td>
</tr>
<tr>
<td>Local, 40%</td>
<td>42</td>
<td>35</td>
<td>31</td>
<td>25</td>
</tr>
<tr>
<td>Local, 50%</td>
<td>28</td>
<td>32</td>
<td>30</td>
<td>23</td>
</tr>
</tbody>
</table>

Interestingly, as the vehicle sizes increase, the maximum and average wait times tend to decrease. This may be due to where there is more opportunity for ridesharing. As discussed previously, areas with large trip volume tend to have more ridesharing opportunities and these areas tend to be near each other. Because of this, it is likely that larger vehicles are traveling to and from popular origins and destinations, which means that the chance that there is an available vehicle relatively close to another popular origin is higher. However, trips without ridesharing are placed in 3 passenger vehicles and if there are very sporadic trips from an area where few people go, then it would be more difficult to find an available vehicle.

As expected, we see a decrease in total wait time as fleet size increases. In all vehicle types, and particularly in 3 passenger vehicles, we see that, proportionally, the total wait time decreases the most when increasing the fleet size from 10% to 20%. Overall, we see
Table 5.4: Total wait time (in minutes) for passengers waiting longer than 5 minutes

<table>
<thead>
<tr>
<th></th>
<th>3 Passenger</th>
<th>6 Passenger</th>
<th>15 Passenger</th>
<th>50 Passenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local, 10% between</td>
<td>19,051,530</td>
<td>2,921,052</td>
<td>486,150</td>
<td>60,608</td>
</tr>
<tr>
<td>Local, 20% between</td>
<td>8,792,609</td>
<td>1,831,477</td>
<td>265,021</td>
<td>31,594</td>
</tr>
<tr>
<td>Local, 30% between</td>
<td>5,533,311</td>
<td>1,125,007</td>
<td>162,659</td>
<td>24,543</td>
</tr>
<tr>
<td>Local, 40% between</td>
<td>3,313,656</td>
<td>670,548</td>
<td>97,425</td>
<td>17,690</td>
</tr>
<tr>
<td>Local, 50% between</td>
<td>1,852,625</td>
<td>385,552</td>
<td>63,515</td>
<td>12,388</td>
</tr>
</tbody>
</table>

decreasing marginal decreases in wait time as the fleet size increases, indicating that at some point, it will not be very beneficial in terms of decreasing wait times, to further increase the fleet size.

Finally, this strategy also requires EMR as well as local repositioning. The repositioning costs, relative to the naive case, is shown in Figure 5.7. In general, the total repositioning miles increased slightly from the simple local repositioning case, but are still very similar. Most of the increase in repositioning comes from EMR empty miles, rather than local empty miles, telling us that the vehicles end up farther from their beginning of day initial distribution. Even with the extended search, the total empty repositioning miles for each case is less than or at most, equal to the naive repositioning cost. This indicates then, that a smaller fleet size with repositioning during the day is more efficient than the naive strategy. It also reduces both the empty mileage costs from repositioning the fleet and capital costs from purchasing additional vehicles.
Figure 5.7: Empty miles comparison for various fleet sizes, local repositioning with extended search
Chapter 6

Summary and Next Steps

6.1 Summary

This thesis serves as a starting point for an actual implementation of an aTaxi system, detailing strategies that a statewide fleet operator might consider in their implementation. We first analyze the effects of ridesharing on the person and vehicle miles traveled, then investigate several repositioning strategies: a naive strategy to establish the upper bound, an impossible infinitely fast repositioning strategy to establish a lower bound, and local repositioning strategies to decrease the fleet size from the initial upper bound established. Finally, we look at the costs associated with each repositioning strategy in terms of empty vehicle miles traveled.

According to U.S. Department of Transportation, Office of Highway Policy Information (2015), there were 6,874,100 registered motor vehicles in New Jersey in 2014, including publicly owned buses. In this thesis, we have found that we are able to serve all of New Jersey’s travel demand using a smaller fleet. Additionally, we have increased mobility for groups of people for whom driving is not a safe option.

Even the upper bound of the fleet size, 3,232,096 total vehicles, is more than a 50% reduction from the number of vehicles currently in New Jersey. An optimally sized fleet would likely be much smaller than the upper bound. This fleet reduction would have significant benefits. Fewer vehicles on the road means less congestion and time wasted in traffic, as well as less pollution.
6.2 Limitations

Though care was taken to ensure the accuracy of the prior analyses conducted on the data set used for this thesis, at the end of the day, this data set is still a synthetically generated data set. A large amount of time and effort was spent creating this data set such that it reflects the average travel demand in New Jersey, but as we do not know what the travel demand actually is, we have no basis for comparison. We believe that this data set is representative of travel demand, but we have no basis for determining the accuracy of the data set.

Additionally, with 10.5 million data points to work with, one of the biggest challenges faced in this thesis was computational, both run-time and memory, limitations. All computations were run locally on a MacBook Pro. Even though we were able to reduce the computational intensity by discretizing New Jersey into pixels and analyzing each vehicle type separately, some analyses, such as calculating pixel to pixel repositioning costs, still proved to be too memory intensive.

6.3 Next Steps

There are many potential further research opportunities based on the groundwork laid out in this thesis.

6.3.1 Optimal Fleet Sizing

Now that we have an understanding of the costs and benefits associated with different fleet sizes and different operating techniques, the next step is to use this information to determine an optimal fleet size to serve New Jersey’s travel demand. Here, we lay out the some of the costs that should be taken into account in determining the optimal fleet size.

The goal is to maximize profit, the difference between revenue and cost, while still being able to meet the full demand, so we will consider the costs and revenues local repositioning strategy with extended vehicle search in the optimization.
Costs

We consider the following costs in the operation of the aTaxi system: initial capital costs of purchasing vehicles, maintenance costs, fuel costs, overhead costs, and waiting costs, i.e. the cost associated with the passengers who must wait longer than the advertised level of service (5 minutes). Furthermore, these costs are those that are dependent on fleet size or vehicle miles traveled, which is in turn dependent on fleet size. If we consider the cost in terms of per vehicle mile, we also need to determine the relationship between fleet size and vehicle miles, as well as the relationship between the fleet size and aggregated wait times.

The total vehicle miles traveled include the “loaded” vehicle miles, or vehicle miles with passengers, as well as the empty vehicle miles (local and EMR repositioning). Based on Figure 5.7, we see that fleet size and empty miles have a linear relationship and may be well approximated using linear regression. Additionally, we can also approximate the relationship between fleet size and aggregated wait times with linear or polynomial regression.

<table>
<thead>
<tr>
<th>Table 6.1: Fleet operation costs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>3 Passenger</td>
</tr>
<tr>
<td>Vehicle Purchase Cost</td>
</tr>
<tr>
<td>Vehicle Cost Per Day</td>
</tr>
<tr>
<td>Per Day Operating Cost</td>
</tr>
<tr>
<td>Per Mile Fuel Cost</td>
</tr>
<tr>
<td>6 Passenger</td>
</tr>
<tr>
<td>Vehicle Purchase Cost</td>
</tr>
<tr>
<td>Vehicle Cost Per Day</td>
</tr>
<tr>
<td>Per Day Operating Cost</td>
</tr>
<tr>
<td>Per Mile Fuel Cost</td>
</tr>
<tr>
<td>15 Passenger</td>
</tr>
<tr>
<td>Vehicle Purchase Cost</td>
</tr>
<tr>
<td>Vehicle Cost Per Day</td>
</tr>
<tr>
<td>Per Day Operating Cost</td>
</tr>
<tr>
<td>Per Mile Fuel Cost</td>
</tr>
<tr>
<td>50 Passenger</td>
</tr>
<tr>
<td>Vehicle Purchase Cost</td>
</tr>
<tr>
<td>Vehicle Cost Per Day</td>
</tr>
<tr>
<td>Per Day Operating Cost</td>
</tr>
<tr>
<td>Per Mile Fuel Cost</td>
</tr>
</tbody>
</table>

We provide estimates the capital, maintenance, fuel, and overhead costs in Table 6.1. Maintenance and overhead costs per day are combined into the “Per Day Operating Cost” category. When considering vehicle purchase costs, we must take into account that fleet quality vehicles, that is vehicles designed for longer lifetimes and more miles traveled, will be more expensive than the average comparable non-fleet vehicle. Our estimates for vehicle purchase costs are based relative to the cost of New Jersey Transit buses. The vehicle purchase costs must then be amortized over the expected lifetime of the vehicle and converted to cost per vehicle mile. We assume lifetimes of vehicles at 4, 5, 7, and 10 years for 3, 6, 15, and 50 passenger vehicles, respectively and 300 “typical” days per year. We further

---

1. 50 passenger vehicle based on per unit purchase cost of NJ Transit buses in 2015 NJ Transit Open Session Minutes, July 15, 2015 (2015). Others are estimated relative to 50 passenger vehicle cost.
estimate maintenance at 50% of capital costs and overhead at 25% of capital costs.

Next, we estimate that the fuel cost is $2.50 per gallon, as increases in cost of gasoline could be offset by increased electric usage and the mileage per gallon of 3, 6, 15, and 50 passenger vehicles at 20, 18, 14, and 3 miles per gallon, respectively. Fuel cost per mile is simply price of gasoline divided by miles per gallon.

Of course, these are very rough estimates and should be fine tuned when running an optimization model.

**Revenue**

Our revenue comes from the trips served and the fare per person is determined by the distance traveled. Since the Person Trip miles are constant in this scenario, the fare can determined such that the operator can make at least some percentage profit. However, there must be an upper bound on the fare that can be charged, otherwise, the system is not cost-efficient for passengers.

**Basic Optimization Model**

The problem of finding the optimal fleet size based on maximizing profit can be formulated as follows:

$$
\max_v \quad m_p^i f - (m_{\text{total}}^i)(c^{i}_v + c^{i}_g + c^{i}_o) - c_w t_w^i
$$

subject to

$$
m_{\text{total}}^i = m_{\text{loaded}}^i + m_{\text{local}}^i + m_{\text{EMR}}^i
$$

$$
m_{\text{EMR}}^i = \alpha_{1\text{EMR}}^i v^i + \alpha_{0\text{EMR}}^i
$$

$$
m_{\text{local}}^i = \alpha_{1\text{local}}^i v + \alpha_{0\text{local}}^i
$$

$$
t_w^i = \alpha_{1w}^i v + \alpha_{0w}^i
$$

$$
v_i \geq F S^i_{\text{min}}
$$

(6.1)

where:

- $i \in \{3, 6, 15, 50\}$ is the vehicle type
- $m_p^i$ is the total number of passenger miles traveled for vehicle type $i$
\( f \) is the fare per passenger mile
\( v \) is the number of vehicles
\( c_i \) is the flat cost of each vehicle type \( i \)
\( m_{\text{loaded}}^i, m_{\text{local}}^i \text{ and } m_{\text{EMR}}^i \) are the loaded vehicle miles, empty local repositioning miles, and EMR miles, respectively for vehicle type \( i \)
\( t_{\text{wait}}^i \) is the total number minutes that passengers have to wait beyond the advertised level of service for vehicle type \( i \)
\( c_f^i \) is per mile fuel cost for vehicle type \( i \)
\( c_c^i \) is per mile capitalization cost for vehicle type \( i \)
\( c_o^i \) is per mile operating cost for vehicle type \( i \)
\( c_w \) is per minute wait cost
\( \alpha \) correspond to linear regression coefficients
\( FS_{\text{min}}^i \) is the minimum fleet size for vehicle type \( i \)

In this model, we consider each vehicle type separately and find the optimal fleet size for each vehicle type. It would also be worth looking at a model where all vehicle types are jointly considered. We expect that such a model will discourage the purchase of 50 passenger vehicles, as these vehicles are significantly more expensive, compared to smaller vehicles. Another direction for future research could be to compare the costs of using 50 passenger vehicles to the costs of a fleet without 50 passenger vehicles and the 15 passenger vehicles are the largest vehicles in the fleet. This means that all 50 passenger vehicle trips would be taken in multiple 15 passenger vehicles.

### 6.3.2 Improvements on Ridesharing Methodology

The ridesharing methodology used to construct the data used for this thesis was somewhat simplified compared to real life scenarios due to the size of the data being considered. As noted in the analyses in Chapter 3, there are many cases where the ridesharing was not constructed very efficiently. We have large vehicles traveling long distances with very few passengers. Future research could focus on creating more optimal ridesharing implementations to minimize circuity or maximize Average Vehicle Occupancy. Specifically, a heuristic to find more optimal ridesharing trips is needed as the brute force method of looking for the
best possible combination of all passengers who arrive at the aTaxi stand within 5 minutes of the first passenger is too computationally intensive and time intensive.

Additionally, the ridesharing simulation does not incorporate dynamic ridesharing - that is, pickup up additional passengers after departing from the original departure location. A more robust ridesharing methodology incorporating these features would be more realistic and be able to further reduce the vehicle miles traveled.

6.3.3 Fleet Mix

Even with the current ridesharing implementation, there are opportunities for a smarter fleet usage. In this thesis, we have only considered a myopic strategy of assigning trips to vehicles: the number of passengers strictly determines the type of vehicle that is used and we consider each set of vehicular trips separately. However, in practice, this would not be the case. It may be more efficient at times to use smaller or larger vehicles, depending on what may be available. Future work would look at optimal use of the entire fleet.

In particular, from Figure 4.1, we saw that there are peak vehicle usages for 15 and 50 passenger vehicles around 7 a.m., while usage for these vehicle types during the rest of day was much lower. We required that our minimum fleet size be this peak for 15 and 50 passenger vehicles; however, this is very inefficient because many of these vehicles would not be used during the rest of the day. At the corresponding time periods for 3 and 6 passenger vehicles, there is less the maximum usage, which suggests that it may be more cost-efficient to use the smaller vehicles at those times than to purchase more larger vehicles.

One such implementation of a mixed-fleet usage (for infinitely fast repositioning, lower bound) could be as follows: fix the number of 15 and 50 passenger vehicles, perhaps to the second-largest peak during the day. Looking at 15 passenger trips only, keep track of the number of vehicles in use at any given time. Allow for regular departures of trips in those sized vehicles until the number of vehicles on the road reaches the number of vehicles available. This point represents the full usage of our fleet. If a 15 passenger vehicle becomes empty, i.e. it has dropped off its last passenger, it can be used for another trip. If there are no available vehicles for a departure, this departure is reassigned to a mixture of 6 and 3 passenger vehicles. At the time of departure, the number of passengers and the destinations
is known, and the passengers can be reassigned to a mix of 6 and 3 passenger vehicles such that Average Vehicle Occupancy for the trip is minimized. The same method can be used for 50 passenger vehicles.

The fleet sizes required to such a scenario can be compared to the fleet sizes required for the base scenario described in this thesis. We expect that the fleet size for 3 and 6 passenger vehicles may increase, but the fleet size for 15 and 50 passenger vehicles should decrease. Again, like the above scenario with using 15 passenger vehicles as the maximum vehicle size, this should be further analyzed to see whether the decrease in larger vehicles justifies the increase in smaller vehicles.

6.3.4 County-Level Management

In this thesis, we have focused on a single statewide aTaxi operator. However, in reality, an aTaxi system may not operated on a state-level, but on some kind of district level, as today’s taxis are. Continued research would explore the fleet management strategies and implications for county level operators, where an aTaxi that leaves its home county may not be able to pick up passengers at their destination counties and must return empty to its original county.

aTaxis that have to make an empty trip back to its original county will inevitability increase the empty vehicle miles traveled during the day, but because the early morning repositioning would only occur within a county, this number of empty miles should be less than the statewide EMR.

6.3.5 Taxi Source Placements

The aTaxi system in this thesis assumes that there is a parking structure at each pixel at each aTaxi stand with enough capacity to hold all of the vehicles that may be parked and not in use at the pixel during the day. With over 21,000 active pixels, even building the parking structures becomes a very large capital expense. To reduce this, further research can explore placing parking structures, or “super sources” strategically at different pixels. Ideally these super sources would be placed such that every active pixel is no farther than 5 minutes from a super source.
Figure 6.1: (A) shows a network graph. (B) shows a dominating set for the graph. If we think of the active pixels as a graph where each pixel center is a node and each pixel has a link to every other pixel that it can reach within 5 minutes, we can formulate this as a Dominating Set problem. An example of a dominating set is shown in Figure 6.1.

This problem can be easily formulated as an Integer Programming problem, as seen in Problem 6.2. However, finding the fewest number of nodes needed to form a dominating set, referred to as the Minimum Dominating Set problem, is known to be NP-complete, meaning there is no efficient algorithm that solves this problem. Fortunately, there are efficient algorithms to approximate the dominating set (Kuhn & Wattenhofer (2003)).

\[
\begin{align*}
\min & \sum_{i \in I} x_i \\
\text{subject to} & \sum_{j \in J} a_{ij}x_j \geq 1 & \forall i \in I \\
& x_i \in \{0, 1\}
\end{align*}
\]

(6.2)

where

- \( x_i \) are the nodes in the graph. \( x_i = 1 \) if node \( i \) is selected to be in the minimum dominating set and 0 otherwise
- \( a \) is the adjacency graph. \( a_{ij} = 1 \) if nodes \( i, j \) are within 5 minutes of each other and 0 otherwise. \( a \) is 1 on its diagonal, as each node is linked to itself
- \( I \) is the set of active pixels

\(^2\)Chaudhuri (2007)
Several different sets of super sources can be compared to determine the costs and effects. One set is to use whatever minimum set that found through an approximation of the integer program. Because this method would increase the amount of empty vehicle miles traveled, another possible set of super sources would consider reducing the empty vehicle miles incurred by this strategy. We can try to minimize the additional empty vehicle miles by strategically selecting an initial set of super sources. This initial set of super sources should be selected such that they are the pixels with the most trips originated during the day. Then, given this set of super sources, we want to find the minimum dominating set.
Appendix A

Selected R Code

This section contains a small selection of the code used for the analyses in this thesis.

A.1 Naive Repositioning Code

This is an example for 3 passenger vehicles in naive repositioning.

```r
library(plyr)
setwd("~/Desktop/Thesis/14Redo/Statewide")

# pull in data
arriveloc = "~/Desktop/Thesis/14Arrivals/distcalc/";
departloc = "~/Desktop/Thesis/14Departures/distcalc/";
arrivelist = list.files(arriveloc)
departlist = list.files(departloc)
nfiles = length(arrivelist)

minFleetArray = matrix(Inf, nrow = 80199, ncol = 3)
eodFleetArray = matrix(Inf, nrow = 80199, ncol = 3)
xVect = rep(1:201, each = 399)
yVect = rep(seq(1:399), 201)
minFleetArray[,1] = xVect
minFleetArray[,2] = yVect
eodFleetArray[,1] = xVect
eodFleetArray[,2] = yVect

# start of day
minfleets = 0;
eodSupply = 0;
totalArrivals = 0;
totalDeparts = 0;
totalCarTypeTrips = 0;

for (f in 1:nfiles) {
  S = read.csv(paste(arriveloc, arrivelist[f], sep=""), header=F, stringsAsFactors=F)
  S = S[,c(25:27,19)]
}
S = cbind(S, 0)
names(S) = c("X", "Y", "time", "occupancy", "wrap")
D = read.csv(paste(departloc, departlist[f], sep=""), header=F, stringsAsFactors=F)
D = D[, c(3:5, 19)]
names(D) = c("X", "Y", "time", "occupancy")

PassDepart = D[(D$occupancy<=3),]
PassDepart = PassDepart[order(PassDepart$time),]
PassArrivals = S[(S$occupancy<=3),]
PassArrivals = PassArrivals[order(PassArrivals$time),]
totalCarTypeTrips = totalCarTypeTrips + nrow(PassDepart)

for (s in 1:nrow(PassArrivals)){
  if (PassArrivals$time[s] > 86400)
    PassArrivals$time[s] = PassArrivals$time[s]−86400
    PassArrivals$wrap[s] = 1
}

for (s in 1:nrow(PassDepart)){
  if (PassDepart$time[s] > 86400)
    PassDepart$time[s] = PassDepart$time[s]−86400
}

PassArrivals$time = floor(PassArrivals$time/60)
PassDepart$time = floor(PassDepart$time/60)

pixelListA = ddply(PassArrivals, .(X,Y, time), summarize, sum(X))
pixelListD = ddply(PassDepart, .(X,Y, time), summarize, sum(X))
pixelList = bind_rows(pixelListA, pixelListD)
(pixelList = unique(pixelList[,1:2])
numPix = nrow(pixelList)

#run through all active pixels
for (k in 1:numPix){
  pixelX = pixelList$X[k]
  pixelY = pixelList$Y[k]
  index = 399*(pixelX−1) + pixelY

  #arrivals from given pixel
  pixelArrivals = PassArrivals[(PassArrivals$X == pixelX & PassArrivals$Y == pixelY),]
  wrapped = sum(pixelArrivals$wrap)
  #corner case when there are only departures from a pixel but no arrivals
  if (dim(pixelArrivals)[1] > 0) {
    pixelArrivals$type <= 'a'
  }

  #departures from given pixel
  pixelDepartures = PassDepart[(PassDepart$X == pixelX & PassDepart$Y == pixelY),]
  #corner case when there are only arrivals to a pixel but no departures
  if (dim(pixelDepartures)[1] > 0) {
    pixelDepartures$type <= 'd'
  }

  #combine the pixelArrivals and pixelDepartures tables & order by time
  pixelActivity <= bind_rows(pixelArrivals, pixelDepartures)
  pixelActivity <= pixelActivity[order(pixelActivity$time),]
  pixelAct <= nrow(pixelActivity)
#compare times and update a supply and min fleet number
pixS = 0;  # supply at that pixel at a given time
fleet = 0;  # cars added from super source

i = 1
while (i <= pixelAct) {
    # departure with no current supply
    if (pixelActivity$type[i] == 'd' & pixS == 0) {
        fleet = fleet + 1;
    }
    # departure with current supply
    else if (pixelActivity$type[i] == 'd' & pixS > 0) {
        pixS = pixS - 1
    }
    # arrival, add to supply
    else if (pixelActivity$type[i] == 'a') {
        pixS = pixS + 1;
    }
    i = i + 1
}

# document bod and eod fleet
minFleetArray[index,3] = fleet + wrapped
eodFleetArray[index,3] = pixS
minFleet = minFleet + fleet
eodSupply = eodSupply + pixS
}

eodFleetVector = as.data.frame(eodFleetArray)
minFleetVector = as.data.frame(minFleetArray)
names(eodFleetVector) = c("x", "y", "num")
names(minFleetVector) = c("x", "y", "num")
eodFleetVector1 = eodFleetVector[is.finite(eodFleetVector$num),]
minFleetVector1 = minFleetVector[is.finite(minFleetVector$num),]
imbalance = eodFleetVector1
imbalance$num = eodFleetVector1$num - minFleetVector1$num
save(imbalance, file = "imbalance.rdata")
save(eodFleetVector1, file = "EOD1.rdata")
save(minFleetVector1, file = "BOD1.rdata")

A.2 Gravity Repositioning Code

Repositioning using gravity model, super pixel to super pixel

setwd("~/Desktop/Thesis/14Redo/EMR/Prev")
library(plyr)
for (cars in c(50,15,6,3)) {
    BOD = load(paste0(cars, "BOD1.rdata"))
    fleetSize = sum(get(BOD)[,3])
    EOD = load(paste0(cars, "EOD1.rdata"))
    imbalance = get(BOD)
    imbalance$num = get(BOD)$num - get(BOD)$num
}
neg = imbalance$num < 0
dem = imbalance
dem$num = (-1) * dem$num * neg
supply = imbalance
supply$num = supply$num * (1 - neg)
zeros = (dem[,3] == 0 & supply[,3] == 0)
dem = dem[!zeros,]
supply = supply[!zeros,]
n = nrow(supply)
tempSupply = supply
tempDemand = dem
tempSupply[,1] = 3 * floor(supply[,1]/3) + 1
tempSupply[,2] = 3 * floor(supply[,2]/3) + 1
tempDemand[,1] = 3 * floor(dem[,1]/3) + 1
tempDemand[,2] = 3 * floor(dem[,2]/3) + 1
tempSupply = as.data.frame(tempSupply)
names(tempSupply) = c("x", "y", "num")
tempDemand = as.data.frame(tempDemand)
names(tempDemand) = c("x", "y", "num")

SPsupply = ddply(tempSupply, .(x, y), summarize, n = sum(num))
SPdemand = ddply(tempDemand, .(x, y), summarize, n = sum(num))
n = nrow(SPsupply)
for (i in 1:n){
    if (SPsupply[i,3] > 0 & SPdemand[i,3] > 0){
        minCars = min(SPsupply[i,3], SPdemand[i,3])
        SPsupply[i,3] = SPsupply[i,3] - minCars
        SPdemand[i,3] = SPdemand[i,3] - minCars
    }
}
zeros = SPsupply[,3] == 0 & SPdemand[,3] == 0
SPsupply = SPsupply[!zeros,]
SPdemand = SPdemand[!zeros,]
n = nrow(SPsupply)
D = matrix(NA, nrow = n, ncol = n)
for (i in 1:n){
    x = ((SPdemand[i,1] - SPsupply[,1])^2 + .5)^2
    y = ((SPdemand[i,2] - SPsupply[,2])^2 + .5)^2
    h = sqrt(x+y)
    D[i,] = h * 1.2
}
F = 1/D^2
sum = 0
for (i in 1:n){
    F[i,i] = 0
}
P = dem[,3]
A = supply[,3]
P = SPdemand[,3]
A = SPsupply[,3]

# note zero supply/demand
Arem = (A==0)
Prem = (P==0)

sumMatrix = F %% A
B = P/sumMatrix

trip = B %% t(A) + F
len = length(A)

Aout = colSums(trip)
Anew = A
Cprev = t(Aout)
count = 0
err = rep(0.1, len)

while(1){
diff = abs(Cprev - A)
if (all(diff < err)){
    break;
}
Ainput = Anew
A1 = A * Ainput

# remove zeros to guarantee no NaNs during division
A2 = A1[A1!=0]
C1 = Cprev[Cprev!=0]
AN1 = A2/C1
temp = 1

for (i in 1:len){
    if(Arem[i]){  
        Anew[i] = 0
    } else{
        Anew[i] = AN1[temp]
        temp = temp+1
    }
}

B = (P/ (F %% Anew))

trip = tcrossprod(B, Anew) + F
Aout = colSums(trip)
Cprev = t(Aout)
count = count +1
}

cost = D * trip
emptyMiles = sum(cost)
A.3 LP Repositioning Code

LP Repositioning, super pixel to super pixel

```r
library(Matrix)
library(gurobi)
library(plyr)

for (per in c(10, 20, 30, 40, 50)) {
  setwd(paste0("~/Desktop/Thesis/14Redo/Local/Unserved/", per, " Percent"))
  repoCost = c()
  for (cars in c(50, 15, 6, 3)) {
    BOD = get(load(paste0("startFleet", cars, ".rdata")))
    EOD = get(load(paste0("endFleet", cars, ".rdata")))
    active = (BOD[,3]>0 | EOD[,3]>0)
    BOD = BOD[,active,]
    EOD = EOD[,active,]
    imbalance = BOD
    fleetSize = sum(BOD[,3])
    imbalance[,3] = EOD[,3] - BOD[,3]
    neg = imbalance[,3] < 0
    dem = imbalance
    dem[,3] = (-1)*dem[,3]*neg
    supply = imbalance
    supply[,3] = supply[,3]*(1-neg)
    zeros = (dem[,3]==0 & supply[,3]==0)
    dem = dem[Zeros ,]
    supply = supply[Zeros ,]
    tempSupply = supply
    tempDemand = dem
    tempSupply[,1] = 3*floor(supply[,1]/3)+1
    tempSupply[,2] = 3*floor(supply[,2]/3)+1
    tempDemand[,1] = 3*floor(dem[,1]/3)+1
    tempDemand[,2] = 3*floor(dem[,2]/3)+1
    tempSupply = as.data.frame(tempSupply)
    names(tempSupply) = c("x", "y", "num")
    tempDemand = as.data.frame(tempDemand)
    names(tempDemand) = c("x", "y", "num")
    SPsupply = ddply(tempSupply, .(x, y), summarize, n = sum(num))
    SPdemand = ddply(tempDemand, .(x, y), summarize, n = sum(num))
    n = nrow(SPsupply)
    for (i in 1:n) {
      if (SPsupply[i,3]>0 & SPdemand[i,3]>0) {
        minCars = min(SPsupply[i,3], SPdemand[i,3])
        SPsupply[i,3] = SPsupply[i,3] - minCars
        SPdemand[i,3] = SPdemand[i,3] - minCars
      }
    }
    zeros = SPsupply[,3] == 0 & SPdemand[,3] == 0
    SPsupply = SPsupply[Zeros ,]
    SPdemand = SPdemand[Zeros ,]
  }
}
```
A.4 Local Repositioning Code

Functions used for local repositioning, extended vehicle search.

```r
# Local Repositioning
# expand search until a vehicle is found
# lastInd = 1442
```
xVect = rep(1:206, each = 399)
yVect = rep(seq(1:399), 206)

getTripInfo = function(carType){
  if (carType == 3){
    loc =˜/Desktop/Thesis/14Redo/Departures/3PaxDepart/
    files = list.files(loc)
  }
  else{
    loc =˜/Desktop/Thesis/14Redo/Departures/
    files = paste0(carType, "paxDepartures.csv")
  }
  return(list(files, loc))
}

#Step 1: Build global supply matrix - X, Y, supply by min
buildSupply = function(minFleetArray){
  supply = matrix(0, nrow = 82194, ncol = 1442)
  supply[,1] = xVect
  supply[,2] = yVect
  supply[,3:lastInd] = minFleetArray[,3]
  return(supply)
}

#convert EMR Fleet to matrix containing all pixels,
#where EMR fleet variable only has the active pixels
converEMRtoMatrix = function(EMR){
  fleet = matrix(0, nrow = 82194, ncol = 3)
  fleet[,1] = xVect
  fleet[,2] = yVect
  nPix = nrow(EMR)
  for (n in 1:nPix){
    index = calcIndex(EMR[n,1],EMR[n,2])
    fleet[index,3] = EMR[n,3]
  }
  return(fleet)
}

# If no vehicles are available at a pixel, search nearby for one
repositionLocal = function(supply,carType){
  unserved = matrix(0, nrow = 82194, ncol = 3)
  unserved[,1] = xVect
  unserved[,2] = yVect
  tripInfo = getTripInfo(carType)
  files = tripInfo[[1]]
  loc = tripInfo[[2]]
  nfiles = length(files)
  vehStats = matrix(0, ncol = 9, nrow = 2)
  emptyTrips = matrix(NA, ncol = 6, nrow = 500000)
  extendedTrips = rep(0,500000)
  ETIndex = 1
  extLen = length(extendedTrips)
  ETR = nrow(emptyTrips)
eIndex = 1
  colnames(emptyTrips) = c("oX", "oY", "dX", "dY", "dist", "layer")
  for (f in 1:nfiles){

PassDepart <- read.csv(paste(loc, files[f], sep=""), header=F, stringsAsFactors=F)
PassDepart <- PassDepart[,c(3:5,19,25:27)]
names(PassDepart) = c("X", "Y", "time", "occupancy","dX", "dY", "dTime")

nDepart = nrow(PassDepart)
vehStats[1,8] = vehStats[1,8]+nDepart

#atMidnight = PassDepart$time<86400 & PassDepart$dTime >=86400
PassDepart = PassDepart[order(PassDepart$time),]
wrap = PassDepart$time>=86400
PassDepart$time[wrap] = PassDepart$time[wrap] - 86400
PassDepart$dTime = floor(PassDepart$dTime/60)
wrapD = PassDepart$dTime>=86400
PassDepart$dTime = floor(PassDepart$dTime/60)

#look at each departure
#ptm <- proc.time()
for (k in 1:nDepart)
{
pixelX = PassDepart$X[k]
pixelY = PassDepart$Y[k]
time = PassDepart$time[k]
if (time < 0)
  time = time+1440
timeIndex = time + 3
index = calcIndex(pixelX, pixelY)
carsNeeded = ceiling(PassDepart$occupancy[k]/50)
occupancy = c(rep(50, floor(PassDepart$occupancy[k]/50)), PassDepart$occupancy[k]%/%50)
for (cn in 1:carsNeeded)
{
  assigned = FALSE
  #look at the departure pixel for a vehicle
  if (supply[index, timeIndex]>0){
    startIdx = supplyIndexStart(supply[,index], timeIndex)
    supply[index, startIdx:lastInd] = supply[index, startIdx:lastInd] - 1
    assigned = TRUE
    vehStats[1,1] = vehStats[1,1] + 1
    destIndex = calcIndex(PassDepart$dX[k], PassDepart$dY[k])
    destTime = PassDepart$dTime[k]+3
    supply[destIndex, destTime:1442] = supply[destIndex, destTime:1442] + 1
  }
  #if there are none, look first locally for a vehicle
  if (!assigned){
    for (t in 1:5){
      if (assigned)
        break
      indexList = c(calcIndex(pixelX+t, pixelY),
        calcIndex(pixelX-t, pixelY),
        calcIndex(pixelX, pixelY+t),
        calcIndex(pixelX, pixelY-t))
      for (a in 1:(t-1)){
        indexList = c(indexList, calcIndex(pixelX+a, pixelY),
          calcIndex(pixelX-a, pixelY),
          calcIndex(pixelX, pixelY+a),
          calcIndex(pixelX, pixelY-a))
      }
      for (b in 1:(t-1)){
        indexList = c(indexList, calcIndex(pixelX+a, pixelY+b),calcIndex(pixelX+a, pixelY-b))
      }
    }
  }
}
calcIndex(pixelX-a, pixelY+b), calcIndex(pixelX-a, pixelY-b))

for (i in 1:length(indexList)){
    currentIndex = indexList[i]
    if (currentIndex <= 0)
        next
    if (timeIndex - t < 3)
        timeIndex = lastInd+t
    thisInd = timeIndex-t
    if (supply[currentIndex, thisInd] > 0){
        assigned = TRUE
        startIdx = supplyIndexStart(supply[currentIndex,], thisInd)
        destX = supply[currentIndex,1]
        destY = supply[currentIndex,2]
        emptyDist = calcDist(pixelX, pixelY, destX, destY)
        supply[currentIndex, startIdx:lastInd] = supply[currentIndex, startIdx:lastInd] - 1
        emptyTrips[eIndex,] = c(pixelX, pixelY, destX, destY, emptyDist, t)
        eIndex = eIndex + 1
        if (eIndex > ETR){
            addl = matrix(NA, ncol = 6, nrow = 50000)
            emptyTrips = rbind(emptyTrips, addl)
            ETR = ETR + 50000
        }
        vehStats[1,t+1] = vehStats[1,t+1] + 1
        vehStats[2,t+1] = vehStats[2,t+1] + emptyDist
        destIndex = calcIndex(PassDepart$dX[k], PassDepart$dY[k])
        destTime = PassDepart$dTime[k]+3
        supply[destIndex, destTime:1442] = supply[destIndex, destTime:1442] + 1
        break
    }
}
}

if still not found, extend search until a vehicle is found

while(!assigned){
    indexList = c(calcIndex(pixelX+t, pixelY),
                  calcIndex(pixelX-t, pixelY),
                  calcIndex(pixelX, pixelY+t),
                  calcIndex(pixelX, pixelY-t))
    for (a in 1:(t-1)){
        indexList = c(indexList, calcIndex(pixelX+a, pixelY+(t-1)), calcIndex(pixelX+a,
                                                                                 pixelY-(t-1)),
                   calcIndex(pixelX-a, pixelY+(t-1)), calcIndex(pixelX-a, pixelY-(t-1)))
    }
    for (b in 1:(t-2)){
        indexList = c(indexList, calcIndex(pixelX+(t-1), pixelY+b), calcIndex(pixelX+(t-1),
                                                                                 pixelY-b),
                   calcIndex(pixelX-(t-1), pixelY+b), calcIndex(pixelX-(t-1), pixelY-b))
    }
    for (i in 1:length(indexList)){
        currentIndex = indexList[i]
        if (currentIndex <= 0 | currentIndex > 82194)
            next
        if (timeIndex - 5 < 3)
            timeIndex = lastInd+5
}
thisInd = timeIndex - 5
if (supply[currentIndex, thisInd] > 0) {
  assigned = TRUE
  startIdx = supplyIndexStart(supply[currentIndex,], thisInd)
  destX = supply[currentIndex, 1]
  destY = supply[currentIndex, 2]
  emptyDist = calcDist(pixelX, pixelY, destX, destY)
  supply[currentIndex, startIdx:lastInd] = supply[currentIndex, startIdx:lastInd] - 1
  emptyTrips[eIndex,] = c(pixelX, pixelY, destX, destY, emptyDist, t)
  eIndex = eIndex + 1
  if (eIndex > ETR){
    add1 = matrix(NA, ncol = 6, nrow = 500000)
    emptyTrips = rbind(emptyTrips, add1)
    ETR = ETR + 500000
  }
  extendedTrips[ETIndex:ETIndex+cn-1] = t
  ETIndex = ETIndex+cn
  if (ETIndex > extLen){
    extendedTrips = c(extendedTrips, rep(0,500000))
    extLen = extLen + 500000
  }
  ind = 8
  destIndex = calcIndex(PassDepart$dX[k], PassDepart$dY[k])
  destTime = PassDepart$dTime[k] + 3 + t - 5
  if (destTime > 1442)
    destTime = destTime - 1442
  supply[destIndex, destTime:1442] = supply[destIndex, destTime:1442] + 1
  vehStats[1, ind+1] = vehStats[1, ind+1] + 1
  vehStats[2, ind+1] = vehStats[2, ind+1] + emptyDist
  break
}
t = t+1
if (t > 1440)
  break
}
#none available, unserved demand
if (!assigned){
  unserved[index, 3] = unserved[index, 3] + 1
  vehStats[1, 7] = vehStats[1, 7] + 1
  destIndex = calcIndex(PassDepart$dX[k], PassDepart$dY[k])
  destTime = PassDepart$dTime[k]+3
  supply[destIndex, destTime:1442] = supply[destIndex, destTime:1442] + 1
}
}
#proc.time() - ptm
}
emptyTrips = emptyTrips[!is.na(emptyTrips[,1]),]
return (list("EmptyTrips" = emptyTrips, "supplyMatrix" = supply,
  "stats" = vehStats, "unserved" = unserved,"extended" = extendedTrips))
}
#finds the first index to remove supply from
supplyIndexStart = function(sRow, idx){
  value = sRow[idx]
  firstIdx = idx
for ( i in idx:3)
{
    if(aRow[i] == value)
        firstIdx = i
    else
        break

    return (firstIdx)
}

# calculated distance between two pixels
calcDist <- function(oX, oY, dX, dY)
{
x = ((dX - oX)^.5)^2;
y = ((dY - oY)^.5)^2;
h = sqrt(x+y);
d = 1.2*h;
return (d)
}

# calculated index of array for a pixel
calIndex = function(x,y)
{
    return (399*(x-1) + y)
}
References


73


