School of Engineering and Applied Science

Columbia University

CEORE4011 – Infrastructure Systems Optimization

Optimizing Shared Taxi Rides in New York City

Team members:
Chi Hin Tam, ct3183
Zayne Wu, yw4176
Zongyang Zhao, zz3157
Wei Mian, mw3725
Yueqi Shen, ys3613
Davish Vivish, dv2507

Dec/31/2023

1. Introduction

In New York City, green taxis are an integral part of the streetscape, providing convenience to urban transportation. With the acceleration of urbanization, the efficiency and quality of taxi services directly impact the city's rhythm of life and the daily commute of its residents. This study aims to gain a deeper understanding of the usage patterns of New York City's taxis, especially in terms of daily temporal and geographical distribution. Through the analysis of taxi trip data on January 19, 2016, we will reveal key information about peak travel times, hotspot areas, and passenger habits. Furthermore, we will explore how to utilize integer programming models to address the issue of taxi trip sharing, with the goal of enhancing operational efficiency, reducing traffic congestion, and promoting environmental sustainability. These analyses and models not only help to improve the quality of taxi services but also provide data support and decision-making references for intelligent urban transportation planning.

2. Data Visualization and Description

2.1 Visualization

2.1.1 Hourly Distribution of Taxi Trips

Figure 1 illustrates the distribution of taxi trips ranging from 0 to 23, representing each hour in the 24-hour day on January 19, 2016. Notably, this bar chart exhibits a bimodal distribution, highlighting two peak periods which appear to occur around 8-9 AM and 18-19 PM showing that approximately 2700 and 3600 taxi trips were initiated during these times respectively. The troughs of activity, representing the lowest number of trips, occur in the very early morning hours, between 12 AM and 6 AM, with the number of trips during these times falling below 1000.

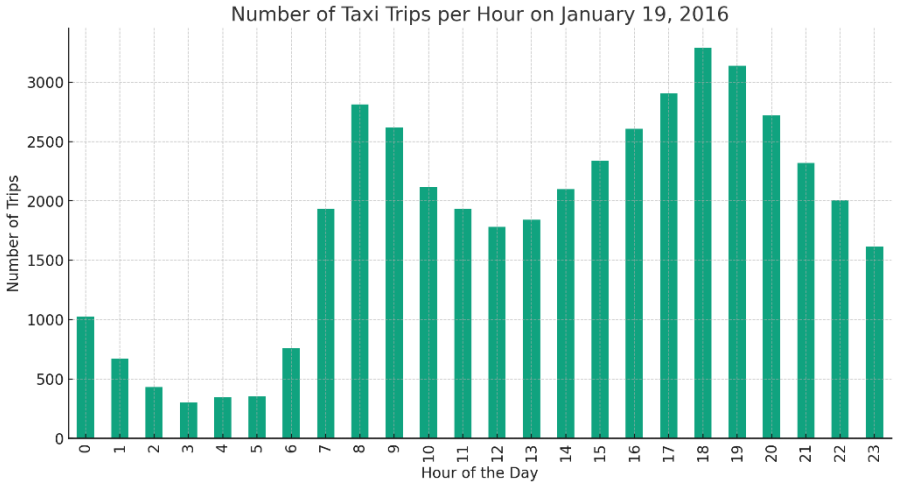

Figure 1. Hourly Distribution of Taxi Trips on January 19, 2016

2.1.2 Dropoff and pickup Locations

Figure 2 is a scatter plot that geographically and statistically indicates that the dispersion of taxi pickup points in New York City on that day. Cross symbols on the plot indicate the coordinates of each pickup point, with latitude on the y-axis ranging approximately from 40.60 to 40.90, and longitude on the x-axis from about -74.15 to -73.75. Notably the pickup points mainly clustered around the longitude of -73.95 to

-73.90 and latitude of 40.75 to 40.80.

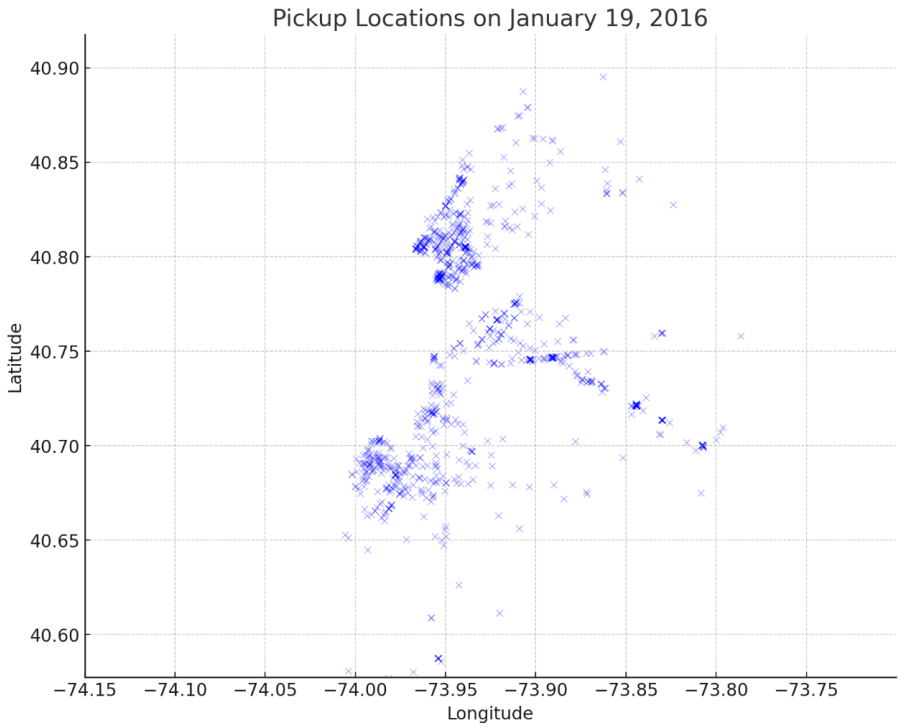

Figure 2. Pickup Locations on January 19, 2016

The scatter plot shown in Figure 3 presents the geographically statistical distribution of taxi trip destinations in New York City. Each cross-symbol marks the latitude and longitude of a dropoff location, clustering densely around longitudes -74.00 to -73.90 and latitudes 40.75 to 40.80.

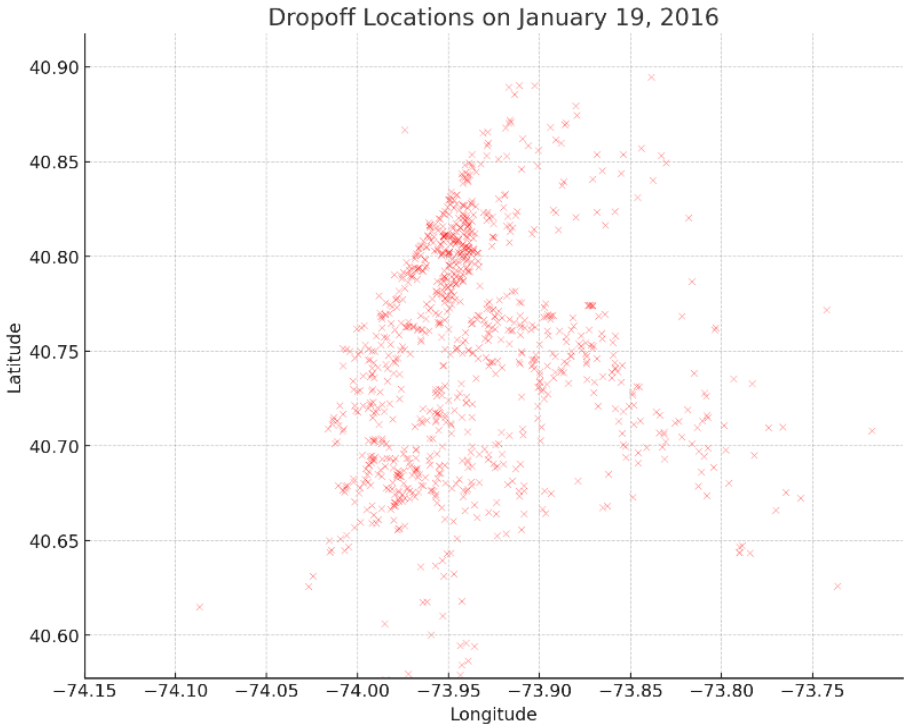

Figure 3. Dropoff Locations on January 19, 2016

2.1.3 Hourly Trip Count and Total Amount

The line graph shown in Figure 4 depicts the variation in the total amount of taxi fares accumulated over each hour of the day. The graph exhibits a cyclical pattern with the total earnings peaking roughly around $80,000 at 8 AM and reaching the highest peak of approximately $90,000 at 6 PM. The lowest point of the graph occurs at 3 AM, with the total amount close to $10,000.

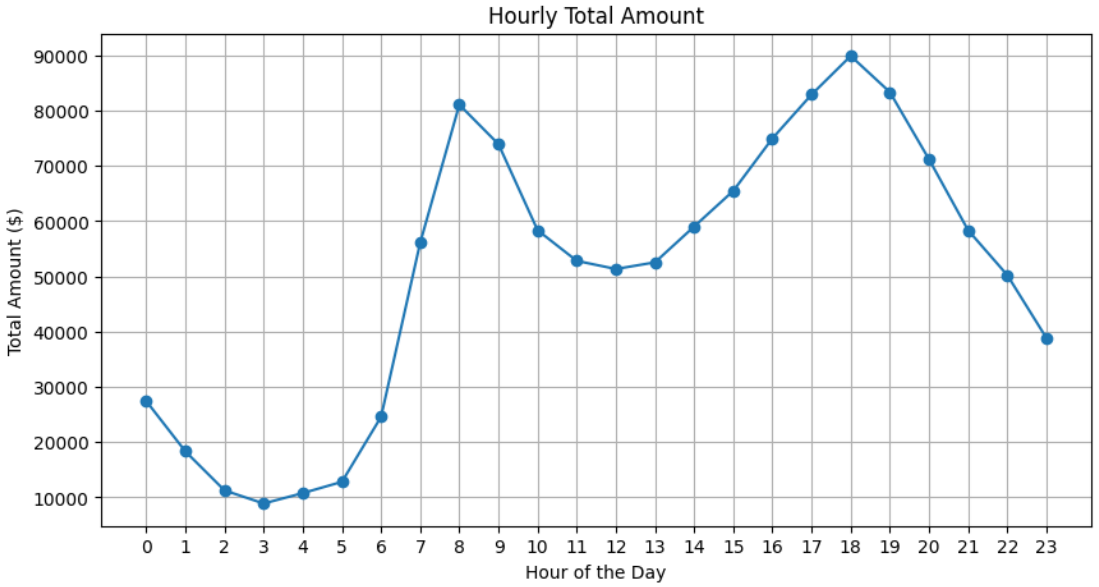

Figure 4. Hourly Total Amount

2.1.4 Distribution of Passenger counts in Taxi Trips

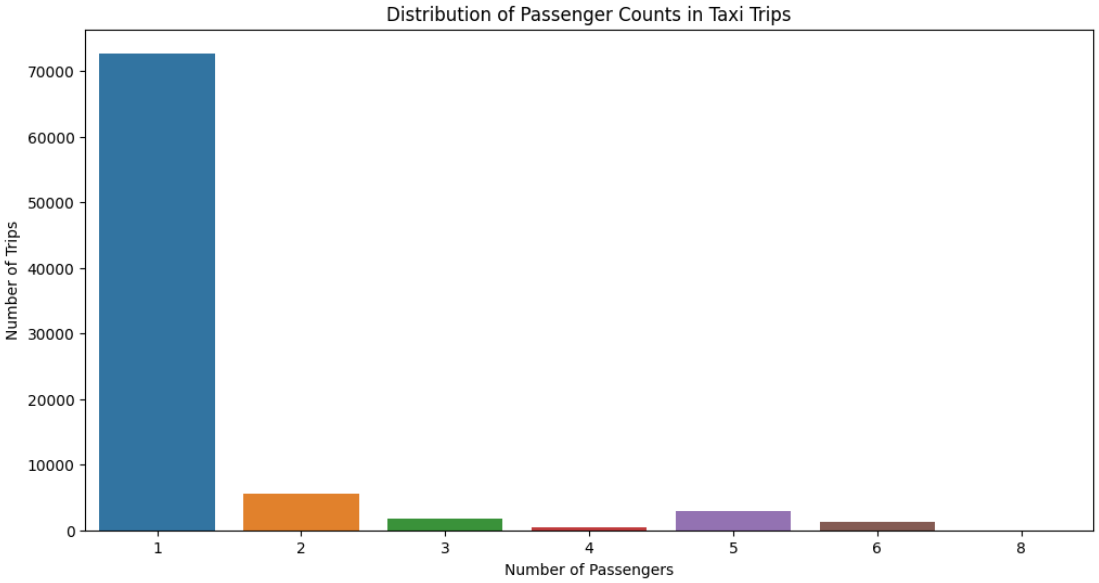

Figure 5. Distribution of Passenger Counts in Taxi Trips

In the analyzed taxi trip data, we found that the vast majority of trips consisted of single passengers, totaling 37,700 occurrences. In comparison, the number of trips with two passengers was 2,866, and those with three passengers amounted to 887. Moreover, trips with five and six passengers were relatively infrequent, with 1,608

and 673 occurrences, respectively. Instances of more than six passengers were extremely rare in the data.

2.2 Data Description

The graph above indicates that there is significant value in promoting and developing shared taxi services.

Firstly, regarding temporal distribution, the frequency of taxi service usage throughout the day exhibits noticeable peaks and troughs. The data shows that during the night (from midnight to 5 AM), both the number of taxi trips and total revenue are relatively low, while there is a marked increase during morning and afternoon hours, especially during the rush hours. This fluctuation suggests a high demand for taxis during peak periods and a decrease in resource utilization during off-peak times. Shared taxi services could alleviate the shortage of taxis during peak times and enhance vehicle utilization rates during off-peak periods.

Secondly, the analysis of geographical distribution indicates that high-demand areas for taxi services are primarily concentrated in the city center and commercial districts. These regions are typically densely populated and bustling with commercial activities, hence the concentrated demand for taxis. Shared taxi services could effectively improve transportation efficiency in these high-demand areas and reduce the time wasted waiting for taxis.

Lastly, the distribution of passenger numbers reveals the tremendous potential for trip sharing. In the analyzed data, trips with single passengers are significantly higher in number than those with two or more passengers, indicating that most taxi trips have room to improve in terms of passenger load efficiency. Trip sharing could not only enhance the occupancy rate per vehicle and reduce travel costs per passenger but also effectively decrease the total mileage driven by vehicles, thereby reducing traffic congestion and exhaust emissions.

In conclusion, shared taxi services can enhance urban transportation efficiency and reduce travel costs through trip sharing, while also having a positive impact on environmental protection. Reducing mileage translates into lower fuel consumption and exhaust emissions, which helps to alleviate urban air pollution and greenhouse gas emissions, crucial for promoting sustainable urban development. Thus, shared taxi services not only meet the growing travel needs of people but also contribute to the optimal allocation of resources.

3. Model Formulation and Problem Solving

3.1 Model Formulation Methodology

We primarily looked at two constraints for our same day data, mainly the maximum distance of drop off and pickup location, as well as the maximum time interval for said pickup and drop off times. We also want to look at how different time periods as well as distance of drop off and pickup affect the pairing of the trips, in which one of our two primary objectives in modeling is to observe how different time in the day would affect the pairings of trips. Our second primary objective is to observe how many different trips can be paired by varying the maximum time waiting interval and distance of pickup and drop off location. Therefore, we would analyze our datasets by

making all else constant and varying either time waiting interval and distance of pickup and drop off location independently. It is important to also note that there's a hidden constraint in the taxi's capacity, in which we made it to 4 as most taxis in New York only have a maximum capacity of 4.

The way we formulated our code was to first slice our data sets to a smaller period of time, as a large data set cannot be analyzed due to its sheer size. This slicing could also aid our goal of looking at different periods of time, in which we focused on a time interval of 15 minutes. As seen previously, there are fluctuations in the number of passengers depending on the time of the day, so we chose 4 different time slots with drastically different numbers of passengers to see if there's any trend. Due to the scope of the problem, we mainly focused on pairing two trips together and not more.

3.2 Code Formulation and Problem Solving

There are a few main parts when we construct the code, each we must consider and add the constraints we mentioned above. First, we load and display the sliced data through utilizing pandas and displaying the csv file. We read the number of passengers, pickup time, pickup location and drop off location in terms of latitude and longitude. Since the pickup and drop off locations are in longitude and latitude, we need to convert them into distances for optimization. Thus, we used haversine distance function to calculate the distance between two points of a single trip from their pickup and drop off longitude and latitude, so that we can use said data to pair trips. Then we used for loops and if functions to find potential pairings with our constrained maximum time interval and pickup/ drop off distance. We then try to maximize the number of paired trips through using an integer programming problem solver named linprog coo_matrix from SciPy. Lastly, we created a constraint in which only unique pairs of shared trips and each trip can be only part of one shared trip. It is important to note again that this code only formulates pairs of trips to be shared, in which it does not add three or four trips together with 1 passenger, but that is not the goal of this project thus it can be ignored.

(The code can be seen in our appendix.)

4. Results and Discussion

4.1 Varying Maximum Pickup/ Drop Off Distance and Pickup Time Interval

Our first step was to examine what the effects of maximum pickup and drop off distance would be to our paired trips, with all else constant. We chose a time where there would be the greatest number of passengers for the most optimized and idealistic data sets for other projections. It can be seen below that as the distance increases, more pairs of trips can be made. At first it had a trend of a sharp increase, from around 370 to 410, but it started to hit a max and dampened after 5km, and after 10km, the number of paired trips stayed constant.

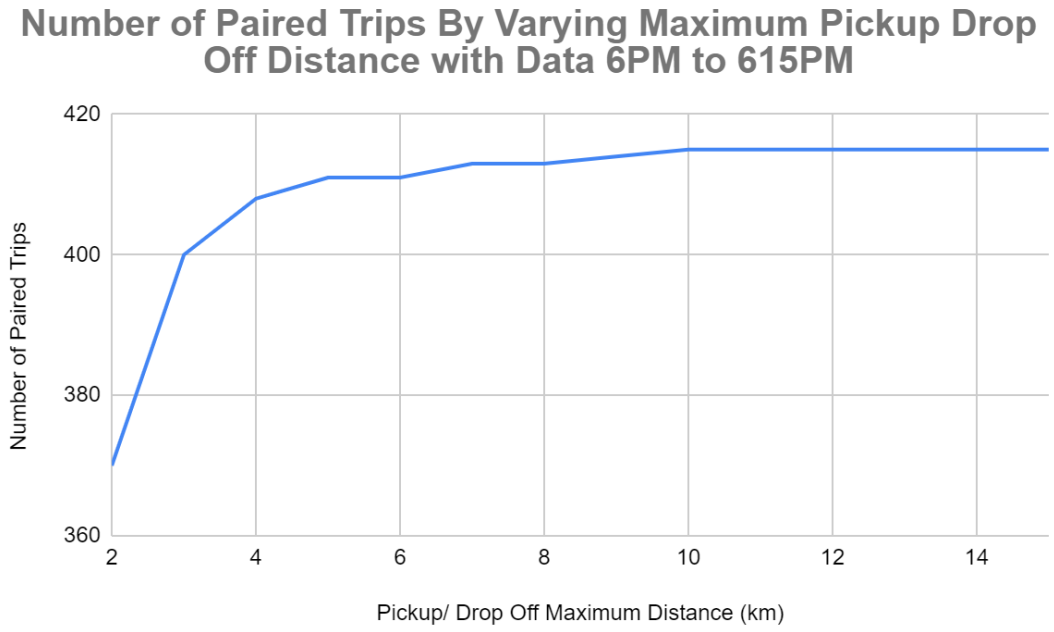

Figure 6. Number of Paired Trips by Varying Maximum Pickup Drop Off Distance with Data 6PM to 615PM

We also looked at how the number of paired trips would vary depending on the interval of pickup time. In which we also observed a similar trend compared to when varying maximum pickup and drop off distance. Where it first rises sharply from 0 minutes to 3-4 minutes, and then it starts to flatten out and we could see it actually stayed constant starting from a maximum of 13 minutes at the end.

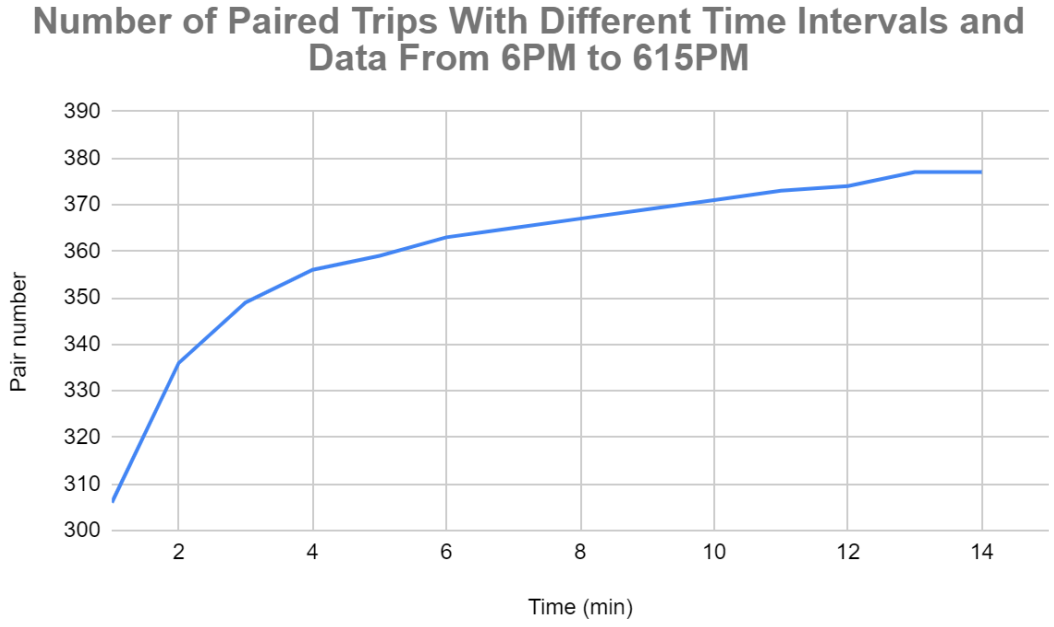

Figure 7. Number of Paired Trips with Different Time Intervals and Data From 6PM to 615PM

This beckons the question whether other data sets at different times of the day would affect the trend of paired trips, and whether they would yield the same trend or not. Thus, we conducted the same optimization for four separate times of the day, namely 3-315am, 7-715am, 1130-1145am, and lastly 6-615pm. Each represents high, medium, and low numbers of passengers as seen from section 2 of this report. and it turns out they're all very similar as seen below. It turns out they're all very similar as seen below.

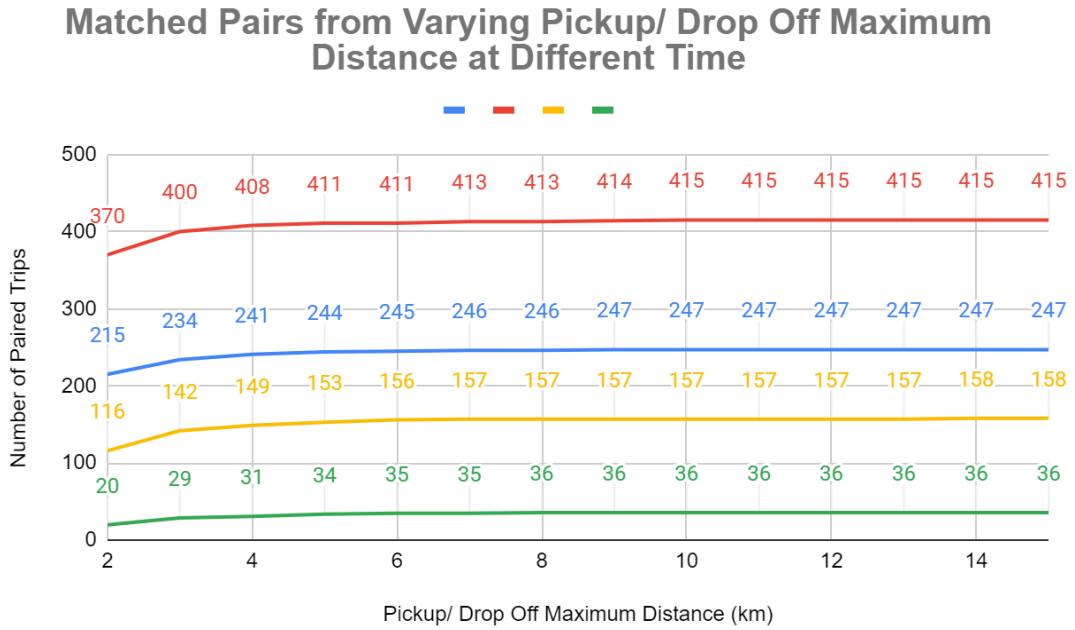

Figure 8. Matched Pairs from Varying Pickup/ Dropoff Maximum Distance at Different Time (Red: 6-615pm, Blue: 1130-1145am, Yellow: 7-715am, Green: 3-315am)

4.2 Most Shareable Trip

The most 'sharable' taxi route varies at different times of the day.. Here are the details:

For the peak hour in the morning (7 AM-7:15 AM)

- Pickup Point: (40.74784851, -73.95697784)
- Dropoff Point: (40.77447128, -73.87229919)
- From Long Island City to LGA
- Number of Potential Shareable Trips: 44

For 11PM to 12PM

- Pickup Point: (40.78821182, -73.94102478)
- Dropoff Point: (40.83632278, -73.93468475)
- From Upper East to Harlem
- Number of Potential Shareable Trips: 262

For the peak hour in the evening (6 PM-6:15 PM)

- Pickup Point: (40.74664688, -73.89151764)
- Dropoff Point: to (40.72927856, -73.88905334)
- Near Long Island City
- Number of Potential Shareable Trips: 28

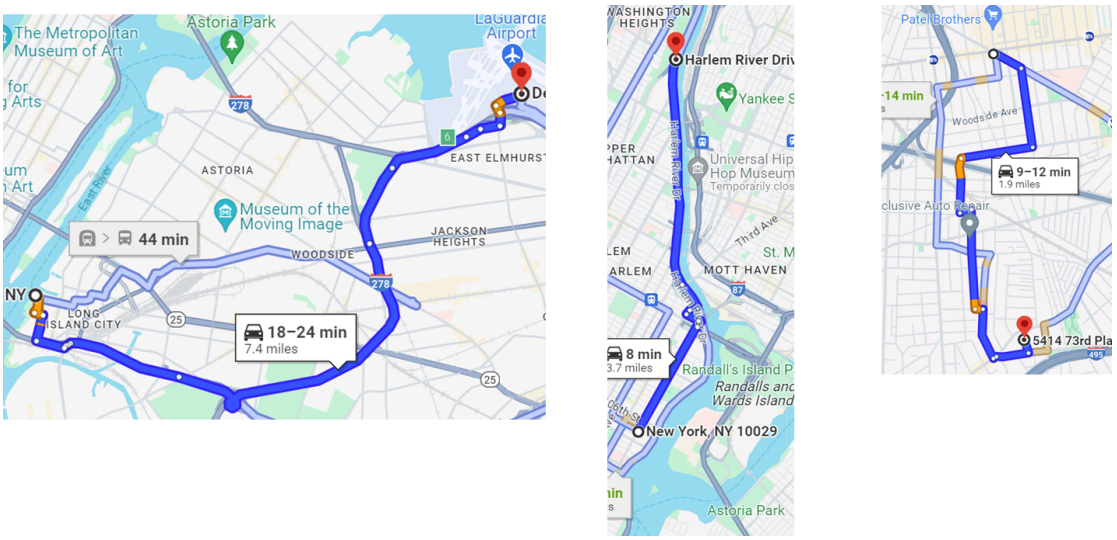

Figure 9. Most Shareable Trip Route (From left to right, 7-715am, 11-12am, 6-615pm)

Demand for shared trips varies significantly throughout the day, with midday having the highest potential for sharing. Certain areas like Long Island City and Upper Manhattan have higher ride-sharing potential, which could be targeted strategically. Different strategies might be needed for different times of day, such as airport runs in the morning and localized shared trips in the evening.

5. Conclusion

Our research has provided valuable insights into optimizing shared taxi rides in New York City. Analyzing taxi trip data from January 19, 2016, we uncovered significant patterns in taxi usage, both temporally and geographically. The study highlighted the potential for efficiency gains in the taxi industry through strategic trip sharing, particularly during peak hours and in high-demand areas. Our findings emphasize the benefits of shared mobility in improving urban transportation efficiency, reducing traffic congestion, and contributing to environmental sustainability. This research not only sheds light on the operational dynamics of taxi services in a bustling metropolis but also offers practical solutions for intelligent urban transportation planning and sustainable city development.

Appendix

```
import pandas as pd

# Load the dataset
file_path = '2016_Green_Taxi_Trip_Data_6PM_to_615PM.csv'
taxi_data = pd.read_csv(file_path)

# Display the first few rows of the dataset
taxi_data.head()
from itertools import combinations
import numpy as np

def haversine_distance(lat1, lon1, lat2, lon2):
    """
    Calculate the Haversine distance between two points on the earth.
    """
    # convert decimal degrees to radians
    lon1, lat1, lon2, lat2 = map(np.radians, [lon1, lat1, lon2, lat2])

    # Haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = np.sin(dlat/2)**2 + np.cos(lat1) * np.cos(lat2) * np.sin(dlon/2)**2
    c = 2 * np.arcsin(np.sqrt(a))
    r = 6371  # Radius of earth in kilometers
    return c * r

# Add pickup datetime as a datetime object
taxi_data['pickup_datetime'] =
pd.to_datetime(taxi_data['lpep_pickup_datetime'])

# Initialize a list to store potential pairs
potential_pairs = []

# Iterate over all combinations of trips
for (idx1, trip1), (idx2, trip2) in combinations(taxi_data.iterrows(), 2):
    # Check the time difference
    time_difference = abs((trip1['pickup_datetime'] -
trip2['pickup_datetime']).total_seconds() / 60)
    if time_difference <= 15:  # within 10 minutes ##### CHANGE HERE FOR
MAXIMIM PICKUP TIME
        # Check pickup and dropoff distance
        pickup_distance = haversine_distance(trip1['Pickup_latitude'],
trip1['Pickup_longitude'],
                                             trip2['Pickup_latitude'],
trip2['Pickup_longitude'])
        dropoff_distance = haversine_distance(trip1['Dropoff_latitude'],
trip1['Dropoff_longitude'],
                                              trip2['Dropoff_latitude'],
trip2['Dropoff_longitude'])

        if pickup_distance <= 2 and dropoff_distance <= 2:  # within 2 km
##### CHANGE HERE FOR MAXIMUM PICKUP/ DROPOFF DISTANCE IN KM
            potential_pairs.append((idx1, idx2))

# Number of potential pairs found
len(potential_pairs), potential_pairs[:10]  # Display first 10 pairs for
inspection
from scipy.optimize import linprog
from scipy.sparse import coo_matrix

# Number of trips in the dataset
num_trips = len(taxi_data)

# Creating the optimization problem
# Each potential pair is a binary variable in the model (1 if the pair is
chosen, 0 otherwise)
```

```
# Coefficients for the objective function (maximize the number of pairs)
c = -1 * np.ones(len(potential_pairs))  # Negative because linprog does
minimization

# Constraints: Each trip can be part of at most one shared ride

# Creating sparse matrix for constraints
row_indices = []
col_indices = []
for i, (idx1, idx2) in enumerate(potential_pairs):
    row_indices += [idx1, idx2]
    col_indices += [i, i]

data = np.ones(len(row_indices))
A = coo_matrix((data, (row_indices, col_indices)), shape=(num_trips,
len(potential_pairs)))
b = np.ones(num_trips)  # Right-hand side of the inequalities

# Bounds for each variable (0 or 1)
x_bounds = [(0, 1) for _ in potential_pairs]

# Solving the integer linear programming problem
res = linprog(c, A_ub=A, b_ub=b, bounds=x_bounds, method='highs-ipm',
options={"disp": True})

# Number of matched pairs
matched_pairs = np.sum(res.x)
matched_pairs, res.success, res.message
```

Code for most ‘shareable’ trip

```
import pandas as pd
import pulp
from geopy.distance import great_circle
from datetime import datetime

# Load the dataset
file_path = '2016_Green_Taxi_Trip_Data_6PM_to_615PM.csv'
taxi_data = pd.read_csv(file_path)

# Preprocessing: Convert datetime strings to datetime objects
taxi_data['lpep_pickup_datetime']                                    =
pd.to_datetime(taxi_data['lpep_pickup_datetime'])
taxi_data['Lpep_dropoff_datetime']                                   =
pd.to_datetime(taxi_data['Lpep_dropoff_datetime'])

# Define parameters
MAX_PICKUP_TIME_DIFF  =  pd.Timedelta(minutes=10)     #  Consider
relaxing this if needed
MAX_DISTANCE = 1.0  # Consider increasing this if needed
MAX_CAPACITY = 4

# Function to calculate distance
def calculate_distance(row1, row2):
```

```
                return    great_circle((row1['Pickup_latitude'],
row1['Pickup_longitude']),
                                       (row2['Pickup_latitude'],
row2['Pickup_longitude'])).kilometers

# Create the ILP problem
problem = pulp.LpProblem("MaximizeSharedRoutes", pulp.LpMaximize)

# Decision variables: Each trip as a potential shared route
potential_shared_route = pulp.LpVariable.dicts("Route", (i for i
in range(len(taxi_data))), 0, 1, pulp.LpBinary)

# Objective Function and Constraints
for i in range(len(taxi_data)):
    # Calculate the number of shareable trips for each route
    shareable_trips = sum(1 for j in range(len(taxi_data))
                                       if (abs(taxi_data.loc[i,
'lpep_pickup_datetime']            -            taxi_data.loc[j,
'lpep_pickup_datetime']) <= MAX_PICKUP_TIME_DIFF and

calculate_distance(taxi_data.loc[i],     taxi_data.loc[j])     <=
MAX_DISTANCE and
                                taxi_data.loc[i, 'Passenger_count']
+ taxi_data.loc[j, 'Passenger_count'] <= MAX_CAPACITY))
    problem += potential_shared_route[i] * shareable_trips

# Solve the problem
problem.solve()

# Extract the results
chosen_route_index  =  [i  for  i  in  range(len(taxi_data))  if
pulp.value(potential_shared_route[i]) == 1]

# Output the chosen route's details and shareable trips count
if chosen_route_index:
    chosen_route = chosen_route_index[0]
                                    chosen_route_details         =
taxi_data.iloc[chosen_route][['Pickup_longitude',
'Pickup_latitude', 'Dropoff_longitude', 'Dropoff_latitude']]
                    chosen_route_details['Pickup_location']     =
f"({chosen_route_details['Pickup_latitude']},
{chosen_route_details['Pickup_longitude']})"
                   chosen_route_details['Dropoff_location']     =
```

```
f"({chosen_route_details['Dropoff_latitude']},
{chosen_route_details['Dropoff_longitude']})"
    shareable_count = sum(1 for j in range(len(taxi_data))
                                if (abs(taxi_data.loc[chosen_route,
'lpep_pickup_datetime']             -             taxi_data.loc[j,
'lpep_pickup_datetime']) <= MAX_PICKUP_TIME_DIFF and

calculate_distance(taxi_data.loc[chosen_route],  taxi_data.loc[j])
<= MAX_DISTANCE and
                                        taxi_data.loc[chosen_route,
'Passenger_count']   +   taxi_data.loc[j,   'Passenger_count']   <=
MAX_CAPACITY))
    result = {
                                    "Chosen    Route    Details":
chosen_route_details[['Pickup_location',
'Dropoff_location']].to_dict(),
        "Shareable Trips Count": shareable_count
    }
else:
    result = "No optimal shared route found."

print(result)
```