# Optimizing the Allocation of Taxis in Manhattan

Louis Raison and Yanchunni Guo

June 28, 2023

## 1 Introduction

### 1.1 Problem Statement

Taxi systems are important components of the urban transportation system because they complement public transport for their flexibility in routes, destinations and supply. In dense cities like New York City, the problem of assigning taxis to different rides is a complex issue, because there is little prior knowledge regarding where and when the next rides will be. The demand at different locations depends on a number of factors such as the day of the week, time of the day, weather and so on. Without proper prior planning, it's likely that too few or too many taxis are deployed to a location; the former leads to unsatisfied demand while the latter results in excessive supply.

If we can predict the demand at different locations of the city and control the amount of taxis deployed to different zones ahead of time, we can reduce the inefficiencies in this system.

### 1.2 Objective

The objective of this project is to deploy the right amount of taxis to a specific location while ensuring all the demands are satisfied.

## 2 Methodology

The project is divided into three parts:

1. Predict the number of pickups and drop-offs in different zone at different time of the day (30min interval)
2. Generate "close scenarios" based on prediction
3. Optimize the allocation of taxis over the four scenarios picked from "close scenarios" and take into account consecutive rides to reduce the number of taxis deployed to each zone

Data used: yellow cab trips information within Manhattan, New York in June 2019 and hourly weather information

## 3 Exploratory Data Analysis

Manhattan is divided in to 67 zones; there are in total 5,807,894 number of trips in June 2019 within Manhattan.

Figure 1 shows that different zones behave differently in different day of the week. Some zones have more pickups and drop-offs during weekday while other zones are more busy during weekends.

Figure 2 gives the total number of rides at different zones of Manhattan between 9am to 10am vs 11am to 12am. Darker color means there are more pickups/drop-offs. We can tell zones behave differently in different time of a day. Zones such as Washington Heights North (243) has lesser rides during 11-12am comparing to 9-10am. These preliminary findings indicate that zones exhibit different patterns; hence using machine learning models to predict the number of pickups/drop-offs in each zone is promising.

Features we considered in the predictive models include: time in a day (hour, first 30minute or second 30 minute in an hour), day of week, wind speed, temperature, dew point, precipitation, rain or no rain

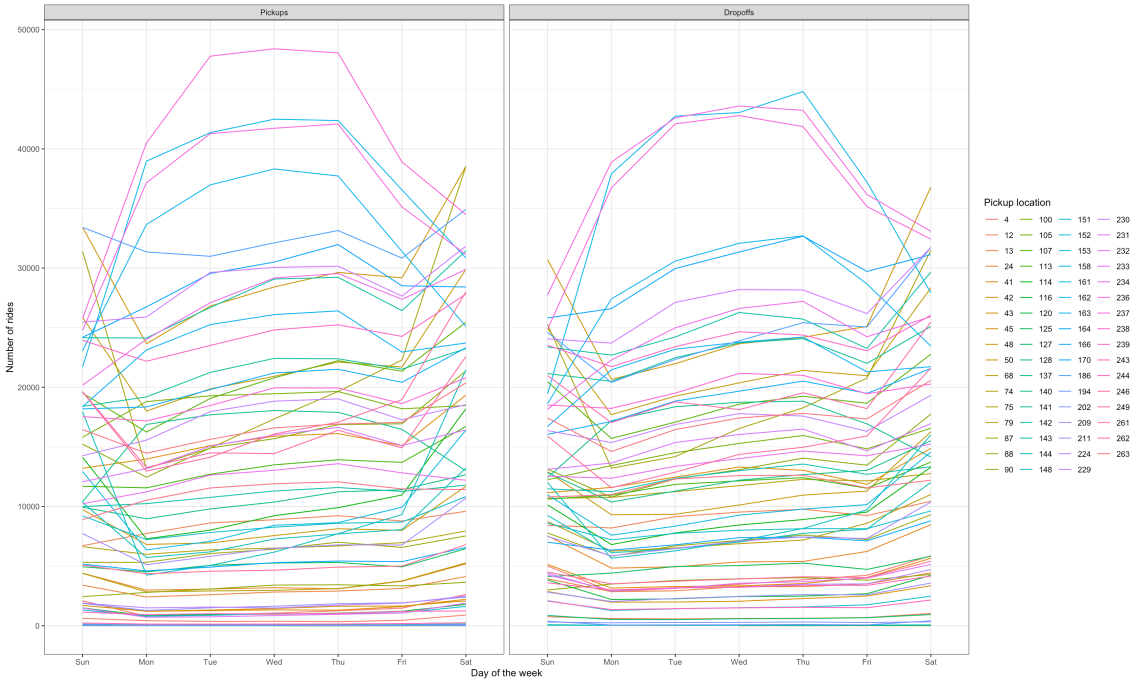



Figure 1: Number of Pickups/Drop-offs at Different Zones in each Day of the Week

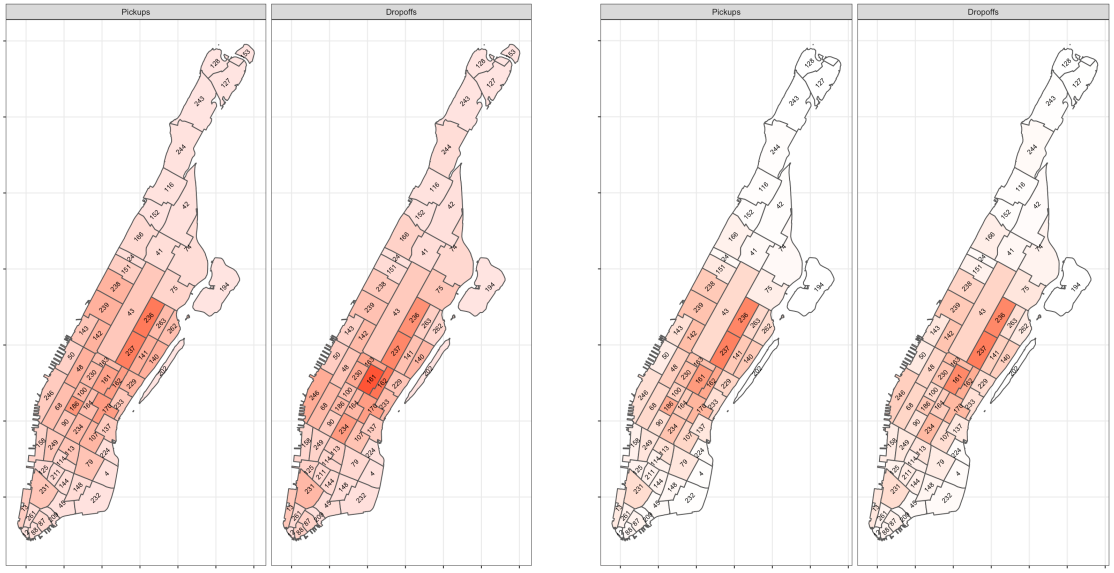



Figure 2: Total Number of Rides in Different Zones of Manhattan between 9-10am vs 11-12am (weekday)

# 4 Predictive Models

Hourly weather information includes wind speed, temperature, dew point, precipitation and rain. Missing values are filled by K-NN optimal impute. Grid search is performed to find the best K.

Prediction results is used to create realistic scenarios for the optimization model. We first create tree models (Random Forests/Optimal Regression Tree) to predict the number of pickups/drop-offs in each zone during a specific time period, and then look at the training data that falls into the same leaf node as the prediction. These training data are selected to be candidate scenarios for the optimization model. Lastly, we sample from these training data to create realistic scenarios for the optimization model.

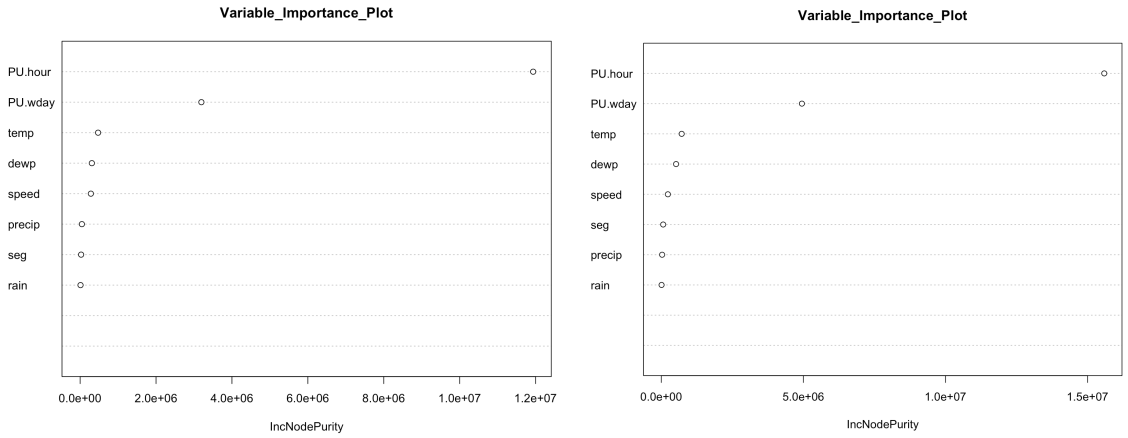
Training data is created using trips information from 06-01-2019 to 06-23-2019 and testing data is created using trips information from 06-24-2019 to 06-30-2019. Number of pickups and drop-offs

are predicted based on 30-minute interval. In order to be able to generate scenarios for each zone, two predictive models (one for predicting pickups, one for predicting drop-offs) are created for each zone individually.

## 4.1 Random Forests

Cross validation is done to tune the number of trees and the number of variables randomly sampled as candidate at each split. Take a specific zone (zone 161) as an example, the important variables for predicting number of pickups/drop-offs are given in Figure 3.

We observe that in both cases, the most important variable is hour of the day, followed by day of the week and temperature.




(a) Pickups

(b) Drop-offs

Figure 3: Variable Importance Plots

## 4.2 Optimal Regression Tree

Another model used is Optimal Regression Tree. Grid search is performed to select the maximum depth of tree and minimum number of observations in a leaf node. Take zone 161 as an example, Figure 4 shows the optimal regression tree used to predict the number of pickups while Figure 5 shows the optimal regression tree used to predict the number of drop-offs.

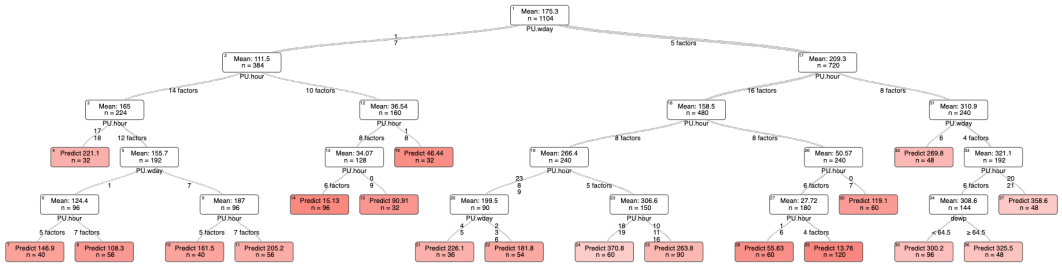


Figure 4: Optimal Regression Tree for Predicting Number of Pickups (Zone 161)

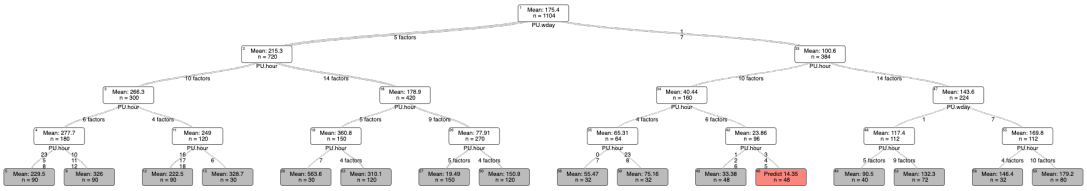



Figure 5: Optimal Regression Tree for Predicting Number of Drop-offs (Zone 161, first 5 layers)

When predicting the number of pickups in zone 161, the first split is day of the week, subsequent splits are hour of the day and dew point. According to the optimal regression tree, most number of pickups in zone 161 occurs during weekday, 6pm/7pm. This result fits with our intuition, because zone 161 is mid-town center and it is usually busy during weekday peak-hours.

# 5 Optimization Model

As we stated initially, our goal is to deploy an appropriate number of taxis to all the different locations of the city. In order to achieve that goal, we started from one of the main difficulties of optimization based on prediction: it is easy to take a decision on very simplified problems but not necessarily easy to merge the individual decisions together.

In particular, we could predict what the demand of taxis rides is in the next time interval in a given location, and then allocate the right number of taxis to the location to satisfy all the rides. The difficult thing here is to know *how to redeploy the taxis and with what frequency.* The main issue of this deployment is that it *doesn't take into account the taxis that can be used for more than one ride.*

The main idea of the model here is to take that into account in a 15 to 30 minutes interval to see how many taxis can be saved on typical scenarios. For this model, we supposed that all the demand should be satisfied, but we could of course do an optimization that maximizes profit instead.

## 5.1 Scenario Generation

We used the trees that we made earlier to generate plausible scenarios. We took a time of day on the prediction we wanted to make, and looked at the predicted leaf of the tree. Then we took some elements in these leaves (as many as the number of scenarios that we wanted to generate, typically 4 or 5), and looked at the rides that were associated with them. We changed their times to fit the time that we wanted to use for our optimization, and then had a few different distributions of plausible rides.

After checking, it happened that the times of day of these "generated rides" were already similar to the ones we were trying to estimate, which confirms our choice of "artificially" moving the rides to the right time.

## 5.2 Model Formulation

Our model formulation gives the number of taxis and deployment that minimizes the total distance made by the taxis overall. It takes into account the fact that taxis can make consecutive rides, if these rides are compatible.

- **Sets**: rides ($i = 1, \ldots, n$), zones ($j = 1, \ldots, m$), taxis ($k = 1, \ldots, K$), scenarios ($l = 1, \ldots, L$)
- **Parameters**: $p_{ij}^{(l)}$ (ride $i$ pickup in zone $j$), $d_{ij}^{(l)}$ (ride $i$ dropoff in zone $j$), $z'^{(l)}_{ij}$ (ride $i$ compatible with a pickup in position $j$ as a first ride), $z_{i_1 i_2}^{(l)}$ (rides $i_1$ and $i_2$ are compatible), $L_{j_1 j_2}$ (distance between $j_1$ and $j_2$)
- **Variables**:
  - $\beta_{ki}^{(l)}$ (taxi $k$ assigned to ride $i$ in scenario $l$),
  - $\gamma_{kj}$ (taxi $k$ deployed in zone $j$, independent of the scenario),
  - $\delta_{i_1 i_2}^{(l)}$ (rides $i_1$ and $i_2$ are consecutive in one taxi's rides),
  - $\zeta_{ki}^{(l)}$ (ride $i$ is the first ride of taxi $k$),
  - $\alpha_{kij}^{(l)}$ (ride $i$ is the first ride for taxi $k$ starting in zone $j$)
- **Objective**: Our choice was to minimize the distance made by taxis over all the scenarios at the same time:

$$\min \left( \frac{1}{L} \sum_{l, i_1 < i_2, j_1, j_2} \delta_{i_1 i_2}^{(l)} d_{i_1 j_1}^{(l)} p_{i_2 j_2}^{(l)} L_{j_1 j_2} \right) + \frac{1}{L} \sum_{l, i, k, j_1, j_2} \alpha_{kij}^{(l)} p_{ij_2}^{(l)} L_{j_1 j_2}$$

  The first term describes the distance added by the allocation of the consecutive rides. The second term accounts for the distance to the first ride of each taxi. The total distance for the rides is fixed because we have to satisfy the rides anyway.

- **Constraints**:
  - Each ride is assigned to exactly one taxi

$$\sum_k \beta_{ki}^{(l)} = 1, \forall k, l$$

  - Each taxi is assigned to exactly one location

$$\sum_{j:1,\ldots,m} \gamma_{kj} = 1, \forall k$$

  - Define which rides are compatible (two rides are compatible if a taxi has sufficient time to travel from the end position of the first ride to the starting position of the second ride before the second ride starts)

$$\beta_{ki_1}^{(l)} + \beta_{ki_2}^{(l)} \leq z_{i_1 i_2}^{(l)} + 1, \forall i_1 < i_2, l, k$$

  - Determine which rides are consecutive

$$\delta_{i_1 i_2}^{(l)} \geq \beta_{ki_1}^{(l)} + \beta_{ki_2}^{(l)} + \sum_{i: i_1+1,\ldots,i_2-1} \beta_{ki}^{(l)} - 1, \forall i_1 < i_2, l, k$$

  - Define starting ride

$$\zeta_{ki}^{(l)} \geq \beta_{ki}^{(l)} - \sum_{i_2:1,\ldots,i-1} \beta_{ki_2}^{(l)}, \forall l, k, i$$

$$\alpha_{ikj}^{(l)} \geq \zeta_{ki}^{(l)} + \gamma_{kj} - 1,\ \zeta_{ki}^{l} \leq 1 + z'^{(l)}_{ij} - \gamma_{kj}, \forall l, k, i, j$$

# 6 Results and Discussions

## 6.1 Prediction Results

Out-of-sample R-squared is used to evaluate the two models. Figure 5 and Figure 6 compare the OSR of the two models when predicting the number of pickups/drop-offs. When predicting the number of pickups, out of the 67 zones, Optimal Regression Tree performs better than Random Forests in 49 zones, Random Forests performs better than Optimal Regression Tree in 13 zones and 5 zones had same OSR. When predicting the number of drop-offs, Optimal Regression Tree performs better than Random Forests in 57 zones, Random Forests performs better than Optimal Regression Tree in 6 zones and 4 zones had same OSR.

We notice that OSR is low at zones that do not have a lot of rides throughout the day. We also notice for most zones, weekday and weekend behave differently. However, current training data only consists of three weeks' information. To further improve the performance of the predictive model, more data should be used to train the model.

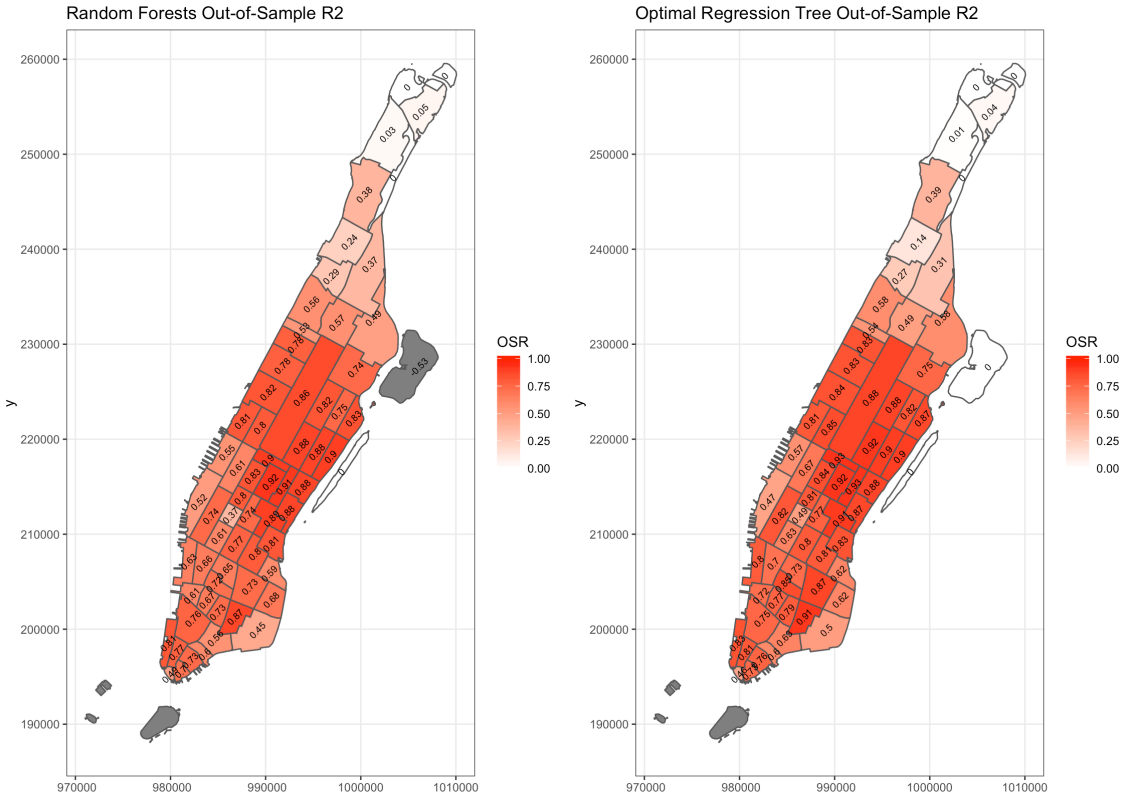



Figure 6: Out-of-Sample R-squared Comparison (Predicting Pickups)

## 6.2 Optimization Results

### 6.2.1 Performance

The results for the optimization are mitigated. On the one hand, We are able to reduce the number of taxis by 3% on average, compared to the model for which we deploy one taxi for each ride (in a 15 minutes interval). On the other hand, we did not have any data on what the real number of taxis was, to compare. However, we can assume that if the deployment of taxis is regular over time, the number of taxis is similar to the highest distribution of rides that we had for the scenarios.

The comparison between the real situation, the number of taxis deployed in the basic model and the number of taxis deployed in our improved model can be seen on figure 8. We can compare for each zone the basic assignment of taxis without consolidation of the consecutive rides, our optimal assignment, the real situation and the different computed scenarios. As we see, the improvement is not extremely significant, but might be more significant as we increase the time range (more taxis can do consecutive rides).

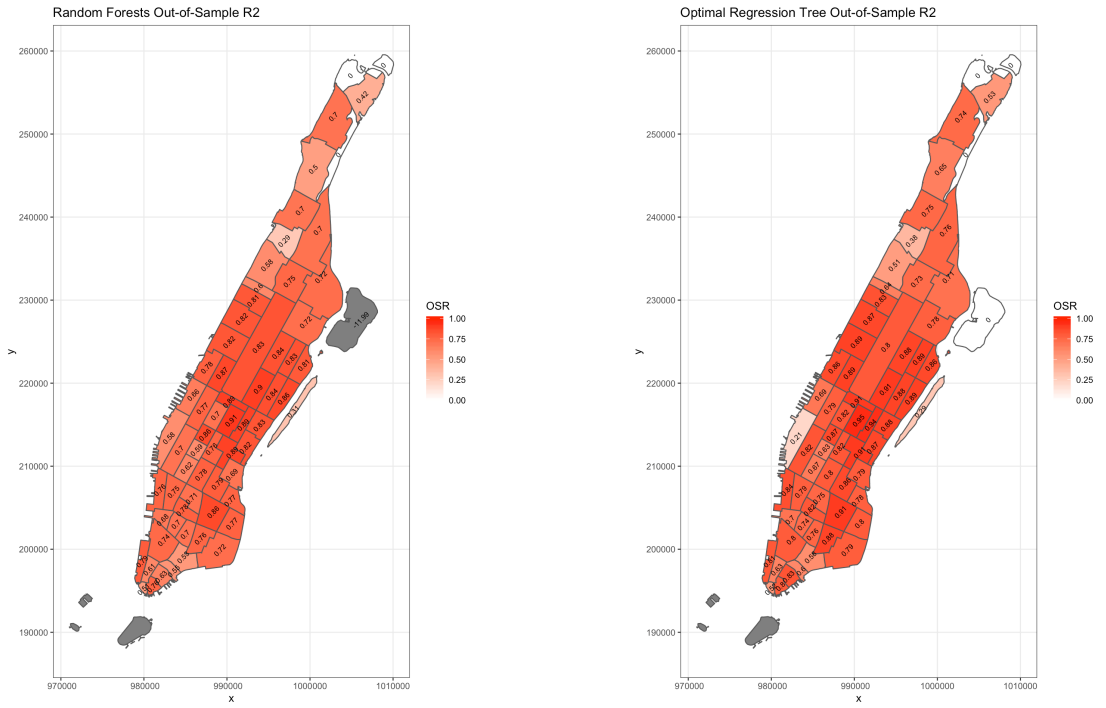



Figure 7: Out-of-Sample R-squared Comparison (Predicting Drop-offs)

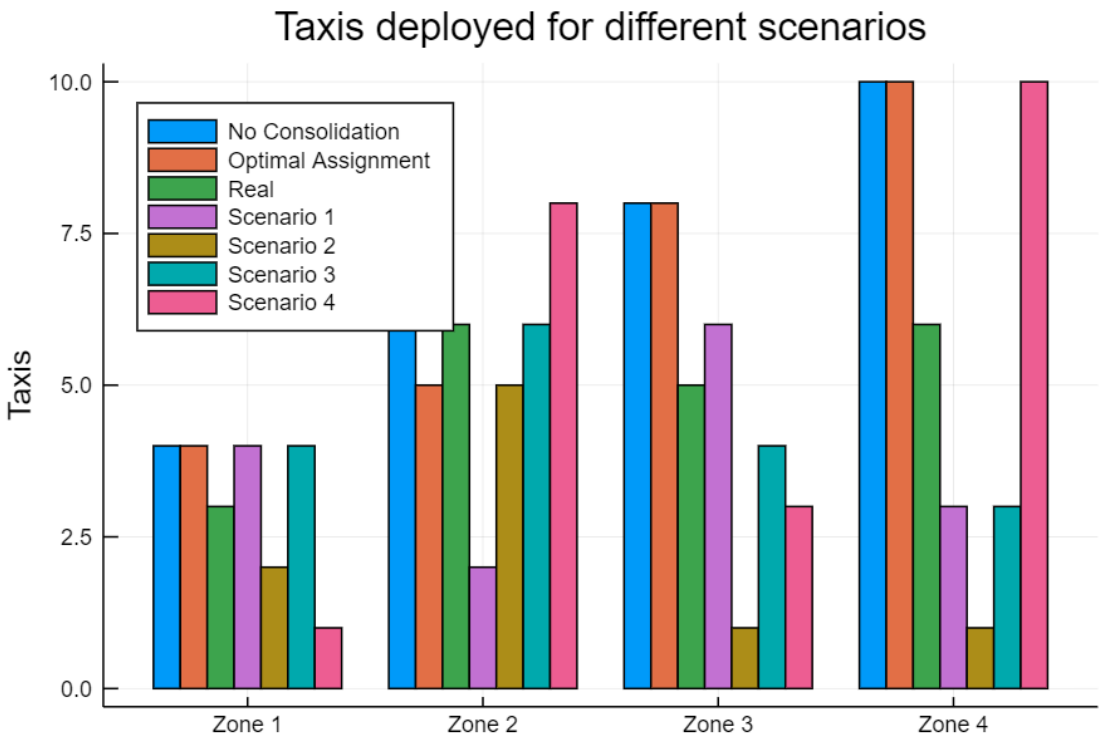



Figure 8: Optimization results on a given time

### 6.2.2 Scalability

The problem is formulated as a linear integer optimization problem, therefore we have to use a high number of variables, that scales linearly with the number of neighborhoods, the number of trips, the number of taxis and the number of scenarios. This makes the problem impossible to scale on all Manhattan, and even difficult to scale when the number of trips is too high.

When the problem is feasible for a given number of taxis, the algorithms generally finds the optimal solution in 1 to 10 seconds, for around 30 trips, 4 neighborhoods and 4 scenarios.

When the problem is not feasible however, the algorithm cannot prove unfeasibility in a reasonable time (at least one hour). We therefore try with less and less taxis until the algorithm cannot provide us with a feasible solution in time.

# 7 Conclusion

In this project, we built a predictive model to estimated the number of pickups and drop-offs in different zones of Manhattan on a given day. Multiple scenarios are then created based on the prediction. Lastly, an optimization model is built to determine the best taxi deployment strategy across all the scenarios.

The predictive model achieved reasonable out-of-sample performance at most of the zones. Predictive performance can be further improved if more data is used to train the model.

The optimization model takes into consideration not only one scenario, but multiple scenarios. Although this method is not as scalable as the other methods currently used, it is feasible on small clusters of neighborhoods and gives good and robust results. In addition, by knowing the profit made for all rides and the cost of taxis, we could easily change this model to a profit optimization without changing much of the model.

The question of the redeployment is still an issue, because we only optimize in a small time window at a few zones. Potential future works include combining these small models into a bigger one.