EINDHOVEN UNIVERSITY OF TECHNOLOGY

Artificial intelligence for logistics and its interfaces

1CM240

Case Study 1

Group 6

Q1 - 2021/2022

Name	Student ID
M.R. Klaassen	0968875
N.C.C. Haenen	1240829
G. Wijsman	0873477

Assignment coordinator: Dr. Albert Schrotenboer

Eindhoven, October 3, 2021

Contents

1 Business Problem

1.1 Context

The Bike sharing system has been around for a long time. The City of Amsterdam first implemented this system in 1965, which was known as "Witte Fietsen." This program collected old bicycles, painted them white, and placed them on the streets for public use without any form of regulation or payment. However, without locks and regulation many of the bicycles were damaged or stolen which led to the project being terminated. Although the program was initially a failure, it paved the way for similar systems to emerge, although it would take more than 40 years until other programs initially became popular.

In the late 2000s, and particularly during the last decade, self-service bike sharing would gain popularity globally. Citi Bike is one of these many bike sharing enterprises around the world. The public bicycle sharing system is active in the New York City boroughs of the Bronx, Brooklyn, Manhattan, and Queens is shown on the map in Figure 1. City bike is considered the largest bike sharing program in the United States. It started in 2013 with 332 stations and over 6000 bikes, but it has grown since to more than 700 stations and 12000 bikes with more expansion plans on the way.

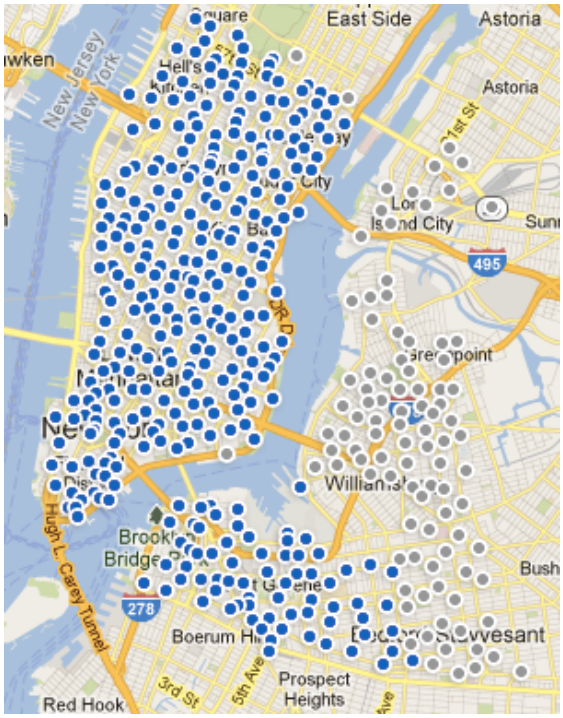

Figure 1: CitiBike stations in New York city.

These bike programs were created to reduce emissions from traffic, decrease road wear, and to reduce traffic and road congestion. They serve as an alternative or an addition to public transportation in the sense that they offer a suitable response for part of inner-city transportation demand and counter the "last-kilometer problem". The definition of the last-kilometer problem is that when travelling in urban areas, the public transportation routes and lack of parking stations hinder travelers from reaching their exact destination, which means they would have to walk the last kilometer. Besides these advantages they are foremostly a green alternative to transportation and they stimulate passengers to get exercise and increase public health.

1.2 Re-balancing Problem

Travelers can go to a bike station nearby which has bikes available, check in to get a bike and they can drop it off at a station which has free bike slots (docks) available as soon as they are done using it. The problem with this system is that some stations have high demand for bikes, while some station have low demand for bikes but high demand for docking stations. When left unchecked, this would soon lead to a disparity among all bike stations and that is why the stations have to be re-balanced. This re-balancing is done by using trucks, or capacitated vehicles that pick up bicycles from stations where the level of occupation is too high and deliver them to stations where the level is too low. This results in optimizing a vehicle routing problem (VRP) with capacity constraints. These problems are known as Bike sharing re-balancing Problem (BRP). It revolves around which decision to on how to route the vehicles, to result in minimal costs and maximum customer satisfaction. This problem can either be modeled as a static or dynamic problem. In the static optimization problem, the stochastic demand is estimated beforehand and the bikes are redistributed according to the expected demand distribution. The dynamic version however, uses real-time information about the system which is then continuously updated to determine the redistribution plan over time (Dell'Amico et al., 2018)

This report proposes a dynamic solution method for the re-balancing problem for Citi Bike in New York City. By using rides information from June 2014, a demand distribution is created through which the outgoing and incoming bike demand for each Citi Bike station in New York City can be predicted. A Random Forest Regression model is used as a machine learning application on the dataset to predict bike and dock demand for all station when provided a date and time in the future. A policy is then created to to determine the optimal route for the vehicle to/from which to move bikes to ensure maximum customer service.

2 SSDP

2.1 State variable

We define $S_t, t \in T$ as the state observed at period t. A state comprises the hour of the day t^h, the simulation time in seconds t^s, the arrival time of the vehicle t^a, the vector containing the current capacities of all stations C_t^s, the capacity available in the vehicle C_t^v at time t, and the location of the vehicle l_t at time t. Therefore, we write the state variable using the following notation:

$$S_t = (t^h, t^s, t^a, C_t^s, C_t^v, l_t)$$

The time of the state is important as some variables, such as customer arrival rates and vehicle driving speed, are dependent on the hour of the day. Capacities of other stations will play a large role in the decision-making process, as explained in Section 3.2. The vehicle capacity is crucial to register at each state as capacity constraint can never be violated. Finally, the location of the vehicle l_t will be an also important attribute of the state. This attribute was added to the model enabling for location-based decision making.

2.2 Decision variable

The moments in time on which a decision is made is denoted with $k \in K$. Therefore, the set of parameters S_k denotes the state on which a decision $x \in X_{S_k}$ is made. Every decision $x_k(S_k)$ is described by a vector of three variables that indicate where the vehicle will pick up bikes (denoted with x_k^{p}), where the vehicle will drop the bikes (denoted with x_k^{d}), and how much bikes will be transported (denoted with x_k^n). Therefore, for the decision variable we write:

$$x_k = (x_k^{\mathrm{p}}, x_k^{\mathrm{d}}, x_k^n)$$

The first two decision variables are integers ranging from 0 to 327, referring to the index of the station list. The last decision variable is an integer, ranging from 1 to the maximum vehicle capacity. Details will be given in Section 3.2.

2.3 Exogenous information

After each decision point k we observe bike supply and demand, which results in new capacities of stations. The exogenous information denoted by $W_{k+1}(S_k)$ does not depend on the decision taken in period k. The arrival process of bikes at each station is stochastic, such that the capacities at each station are also stochastic.

2.4 Transition function

The transitions function describes how we transition towards the state in period $k+1$. The transition function depends on the state S_k, the decision x_k, and the demand information W_{k+1} that is revealed after x_k is determined. For the transition function we use the following notation:

$$S_{k+1} = S^M(S_k, x_k, W_{k+1})$$

The capacities at moment k are known. The decision x_k made at this moment tells us how many bikes are transported to which location. Based on the location of the vehicle and the capacities of the other stations it will be decided from which station to which station the bikes are transported. The transition function is used to update the location of the vehicle and the capacities of the other stations.

2.5 Objective function

We deal with a state-dependent problem with a cost function that depends again on the state S_k, the decision x_k, and the demand information W_{k+1} that is revealed after x_k is determined. For each time a customers arrives at a station with no bikes available, we incur a cost of €1. A cost of €10 is incurred if a vehicle drops more bikes at a station than possible. The transition function will be used to determine the capacities at each station after taking a certain decision x_k. Then, the first part of the cost function is used to incur costs each time the capacity of a station becomes smaller than zero or when the vehicle drops bikes that will bring the capacity at a station above ten. The weight of this cost is denoted with $\alpha \in [0, 1]$. Furthermore, we also incur a cost based on the vehicle travel time. The more time it takes for the vehicle to travel from one station to another, the less stations the vehicle can visit. Therefore, it is decided to incur a cost for traveling. The weight of this cost is denoted using $\beta \in [0, 1]$. The values of α and β will be optimized. The objective is to minimize the summation of these expected total costs:

$$\min \Big[\Big(\mathbb{E} \sum_{k \in K} C(S_k, x_k(S_k)|S_0) \Big) \Big]$$

3 Solution Approach

In this section we will discuss the solution approach. This entails demand predictions and decision-making.

3.1 Demand predictions

In this section we will discuss how demand is predicted. Two random forest models are trained on datasets based on the "TestData.csv" dataset.

First of all, we will discuss which data preparation steps are conducted. The "TestData.csv" dataset contains 936.880 rows with information about individual trips, such as the trip duration, start- and endstation, and longitude and latitude of these stations. Furthermore, there is some individual trip information, such as the bike ID, gender and user type. These data are transformed to obtain two datasets. One for incoming bikes per station per hour, and one for outgoing bikes per station per hour. These datasets have the following features: station ID, date, day of the week, hourly interval and bike count. Hourly interval means during which two hours the bike count is registered, and bike count is the total number of incoming/outgoing bikes in

its respective interval. Moreover, also the possibility of reducing the time interval to half hours has been investigated. The details will be discussed later on in this chapter.

Second of all, these two datasets are used to train two random forest models. One model predicts the outgoing bikes per station, and the other predicts the incoming bikes per station. Features that are used to predict the number of bikes are: station ID, day and hour. Since 'day' is a categorical variable, one hot encoding transformation is used. This makes sure that the machine learning algorithm does not assume that there is an order or ranking present. The dataset for outgoing bikes has 161.012 rows and for incoming bikes there are 160.879 rows. Both datasets are split into training and test sets, where 20% of data is reserved for testing. Figures 2 and 3 show each feature's importance per model. For both models, it can be seen that station ID is the most imported feature, followed by the hour interval. Lastly, weekend days have higher feature importance than weekdays.

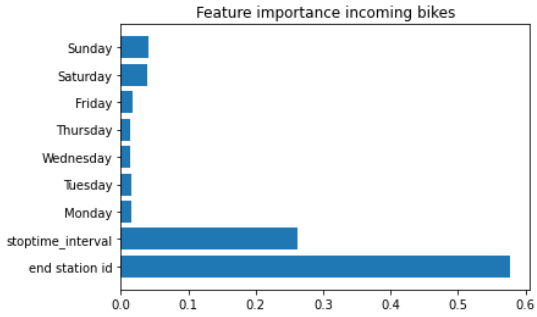

Figure 2: Feature importance incoming bikes

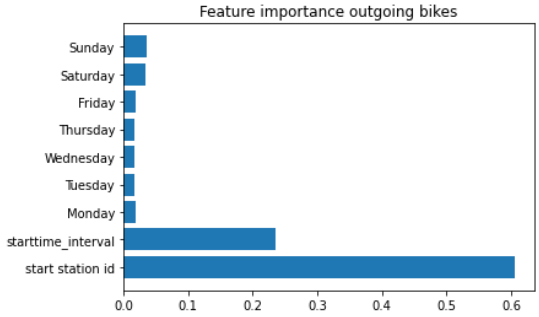

Figure 3: Feature importance outgoing bikes

Table 1: Random forest performance

Time interval	Action type	R^2	MAE
1 hour	Outgoing Bikes	0.61	2.48
	Incoming Bikes	0.40	2.38
0.5 hour	Outgoing Bikes	0.42	1.70
	Incoming Bikes	0.23	1.62

The results for half hour and full hour time intervals can be found in table 1. This table shows the average random forest performance of predicting outgoing and incoming bikes per station per time interval through 10-fold cross validation. It can be seen that the 1 hour models explain more variance than the half hour models. However, their Mean Absolute Error (MAE) is higher. MAE describes the absolute prediction error. This means, that per time interval the average error in predicted vs actual bikes is displayed. The 1 hour time interval is twice as long as the half hour interval. However, the MAE scores are less than twice as big. Therefore, based on higher variance explained and relatively lower MAE, the 1 hour time interval is chosen to predict future demand.

3.2 Decision-making

This part will describe how the implemented policy works and it could result in a more cost-effective decision-making process. The process of decision-making consists of two parts. First, it will be discussed how the optimal values for parameters α and β are found such that the system performs at its best. Secondly, the system is further improved using some reinforced learning algorithm that allows us to find the optimal number of bikes being transported at each vehicle trip. The used method will be extensively explained. We will end this section with some words about how the used policy could be positioned along the lines of the four fundamental policies for SSDP's.

3.2.1 Parameter optimization

As described in chapter 2, we include two forms of costs in order to optimize the decision making at each state. First, we include costs for each time a customer demand arrives at a station with no capacity. Furthermore, we also assume there are costs included for the transport of bikes. Both these costs will be explained in more detail.

As described before, the customer demand over time at each station is forecasted using random forest models. This implies that the expected demand (i.e. expected change in capacity) over a certain time period can be determined for each station. Therefore, if the station capacities are known at a certain moment in time, the expected capacities per station after some time step can be calculated. Costs are included for each time that the expected capacity change is large enough such that we a capacity smaller than zero at a station. A cost of €1 is included for each time a customer cannot be served. This cost is directly related with the costs for negative expected capacity. For instance, if we observe an expected demand of 0.5 during a certain time step and the current capacity of that station is zero, we include an expected cost of €0.5.

The other type of cost that is assumed to be relevant in the system is the transportation cost of the bikes. To come up with the most cost-effective decision policy, it is not sufficient to only include the cost of lost sales. Transport should be included as a cost in the decision-making for the following two reasons. First, a vehicle needs fuel to be able to drive and a driver that must be paid. Secondly, each trip from station to station takes time, during which no transport between any other pair of station is possible (as we only deal with one vehicle). In other words, a very long trip means a long time that the vehicles is occupied such that is not able to serve other stations. The transportation costs are roughly estimated. It is assumed that the vehicle drives, on average, 20 kilometers per hour. If the vehicle drives ten kilometers on each liter gazoline (assuming a price per liter of €2), and the driver is paid €30 per hour, the transportation costs equal €34 per hour. If all other costs are included, such as road tax and insurances, it is assumed that the total transport costs per hour are €36. This implies that a cost of €0.01 is included for each the time the vehicle is driving.

Given the costs of expected negative capacity, the transportation costs and current location of the vehicle, each decision can be ranked. This will be further explained using the example of Figure 4. It depicts a similar bike-sharing system with six station, each having the capacity as indicated in the figure. The current location of the vehicle is at station one. At each decision-making moment, there are three types of decisions that must be made. First of all, it has to be decided at which station the bikes will be picked up. This will always be the station with maximum capacity. As shown in the figure, this will be station 5. Then, it must be decided to which station the vehicle will go to drop the bikes. This will be the station that performs the best in terms of expected cost of lost sales and transportation costs. It can be observed that going from station 5 to any of the others stations (i.e. station 2, 3, 4, or 6) results in different transportation lengths. For instance, going from station 5 to station 4 results in a much lower transportation time than going from station 5 to station 2. To determine to which station the vehicle has to travel to drop the bikes, all possible scenarios will be compared. For each station it will be determined how good the decision would be to choose for that station. This is done by considering the travel time it takes to go to that specific station and what the total expected negative capacity over all station will be if it is decided to go to that station. For example, imagine the decision going from station 5 to station 4, the travel time will be calculated, as well as how much the capacity at each station will change during that travel time. If there is negative expected capacity at any station as a results of choosing for station 4, that decision would result in lower performance. Iterating through each of the stations results in a list of performances indicating how good each decision is. An imaginary example of such a ranking is provided in Table 2. It shows for each station the costs for total expected lost sales and transportation costs. Summing these up results in to cost performance of that station. The lower the cost performance, the better it would be to choose for that station to drop the bikes. As shown, the station that performs the best in terms of cost performance is station 3. Choosing for this station results in a expected cost of lost sales equal to 1, incurred during a transportation time of is $t_1 + t_2$ seconds.

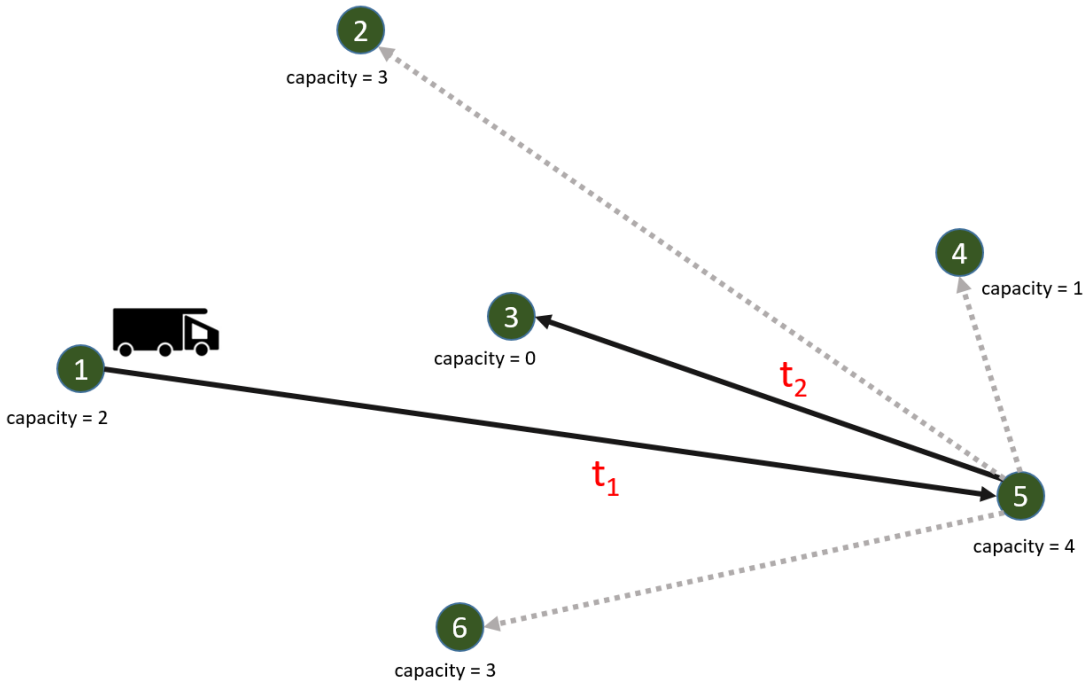

Figure 4: Illustrative example of decision-making policy

Table 2: Finding the station that performs the best in terms of cost performance

Station to drop bikes	**Cost of lost sales**	**Transportation cost**	**Cost performance**
2	4	9	13
3	1	6	6
4	2	5	7
6	2	7	9

In the example from above, the cost performance of each station is calculated assuming that the cost of lost sales and the transportation cost have equal weight. However, adjusting the weights could results in different decisions. For instance, suppose the transportation cost has a much higher weight than the cost of lost sales. Then, probably station 4 would be station with the best ranking at it is the closest to station 5. If we are less interested in minimizing our cost of lost sales, but our aim is to minimize transportation costs, the decision-making will be different. Therefore, the aim is to find parameters α and β that lead to the best performing system.
The performance-based decision-making process as described above has been implemented in the simulation model. Different combinations of values for α and β were tested, ranging from 0.05 to 0.95. To test the performance of a combination of parameter values, the model was executed using the parameter values of Table 3.

Table 3: Parameter values used to find the optimal values for *alpha* and *beta*

Parameter name	**Value**
OUTPUT_FLAG	FALSE
BUSY_PARAMETER	4
STATIONCAPACITY_PARAMETER	10
TOTALBIKES_PARAMETER	0.7
VEHICLECAPACITY_PARAMETER	20
SEED_PARAMETER	10102019
NITERATIONS_PARAMETER	5
EPISODELENGTH_PARAMETER	2

For each combination of parameter values, the average performance in terms of percentage customers served has been calculated over five observations. The results are summarized in Figure 5. It is shown that the policy performs at its best when α is equal to 0.6, meaning that β equals 0.4. From now on, these parameter values are used in the decision-making process to calculate the performance ranking of all different stations at a certain state.

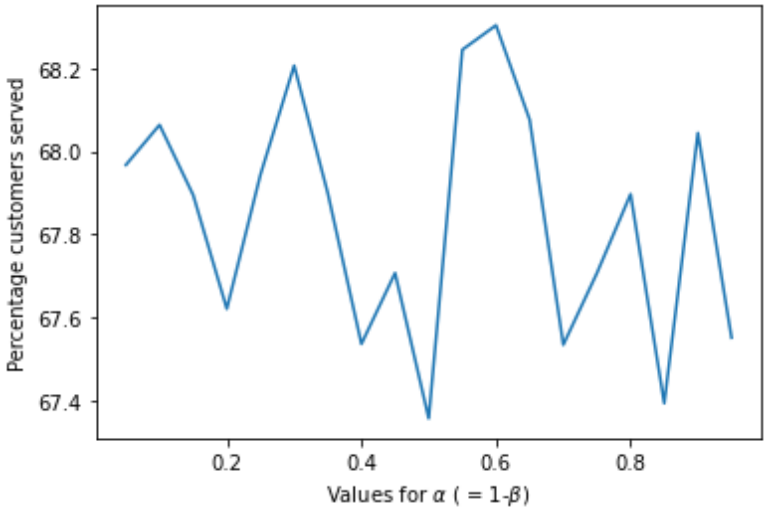

Figure 5: Finding the optimal parameter values for *alpha* and *beta*

Now decision-making policies for first two types of decision are explained, the only type of decision that must be elaborated on is the number of bikes that is transported from station to station. In the current situation, this number is equal to half of the difference between the two capacities of the stations (as long as constraints are not violated). However, in the improved decision-making policy, some kind of training algorithm has been implemented to optimize this decision. This is further explained in the next sub-section.

3.2.2 Reinforcement learning

After is has been decided at which station bikes will be picked up and at which station the bikes will be delivered, the only decision that is left is about the number of bikes that should be transported. A simple form of reinforcement learning has been used to optimize the number of bikes that should be transported. This part elaborates on the machine learning algorithm that being is being implemented.

The reinforcement technique that is used, seeks to find the optimal number of bikes to be transported, based on the current location of the vehicle, the station where it should pick up the bikes, and the station where it is delivering the bikes. Based on this information this information from the past and the corresponding decision and reward the system is able to select those decisions that will result in the biggest rewards.

The technique that is used is illustrated with an example in Table 4. The table shows the data that is required for the reinforcement technique for five different decision moments (note that the learning table in the simulation consists of many more rows containing information at decision moments from the past).The table indicates at each decision moment the current location of the vehicle, the first station it will visit to pick up the bikes, the second station it will visit to drop the bikes, the number of bikes that are transported, and the associated cost for taking that decision. The idea is that the system learns from decisions made in the past by taking into account the cost for taking a decision at a certain state. In the first row of Table 4 it is shown that a cost of €1 is incurred after it was decided to transport 7 bikes from station 14 to station 87 when the current location is 214. At the fourth decision moment the location, first station, and second station are the same as in the first decision moment. However, as the cost of choosing 7 bikes was penalized with a cost, the system tries a different number of bikes. After choosing

to transport 5 bikes, the cost incurred turned out to be equal to 0. At the next decision with the same vehicle location, first station, and second station the system will recognize this and will decide to transport again 5 bikes. This way, the system learn from decision made in the past and it becomes smarter at each next decision moment.

Table 4: Learning table for an improved decision-making

Decision moment	Current location	First station	Second station	Number of bikes	Cost
1	214	14	87	7	1
2	23	45	301	9	1
3	77	96	18	3	1
4	214	14	87	**5**	**0**
5	23	45	301	**9**	**0**
...	...	...	...	...	...

It should be noted that at each decision moment when the current location, first station and second station are known, the learning table is checked for a similar combination that resulted in a zero-cost decision. However, in case the learning table does not contain a row with such similarities, the system will choose a random integer (denoting the number of bikes to be transported) with as minimum one bike and as maximum the difference in capacity between the two stations. In case the difference exceeds the maximum vehicle capacity, a random integer will be chosen between one and the maximum capacity. After choosing a random integer as the number of bikes being transported, more data is added to the learning table which allows the system the become smarter and improve future decision-making.

3.2.3 Policy Positioning

In this section we will discuss which type of stochastic optimization learning model we have created. There are two fundamental strategies: policy search and lookahead approximation. Policy search searches over a class of functions for making decisions to optimize some metric. Lookahead approximations approximate the impact of a decision now on the future. Our proposed policy is a single policy, with reinforcement learning on the number of bikes. This makes use of a look-up table to determine how good a particular past action was in a particular state. Furthermore, random forest models are used to predict hourly demand per station. Since most of the decision making is based on demand prediction, we argue that our policy is a lookahead approximation. As said before, it approximates the impact of a decision now on the future based on this demand prediction.

The Direct lookahead policies have different sub-classes. The proposed policy can be seen as a form of the classic Rolling Horizon policy. The rolling horizon policy is often used for decision making in dynamic stochastic environments. First, a general forecast of relevant information for a number of future time periods is needed. The forecast can either be deterministic or stochastic. Then the timeframe is divided into a number of equally large decision periods. The decision in the first periode becomes the most important and you want to make the most optimal decisions for this period. After this period ends the second period becomes the most important one and again the decisions which lead to the optimal result for the second period are chosen. During each period, the information is continuously updated for the next periods. This procedure repeats every period justifying the term rolling horizon decision making for the practice. Here, the term "horizon" refers to the number of periods in the future for which the forecast is made. It is this horizon, that gets "rolled over" each period Sethi & Sorger (1991).

The proposed policy also creates a forecast for the demand of all stations at the beginning using a random forest generator. Then the timeframe of is divided into multiple equally large periods. After each period the demand is updated with real-time data from the stations and hence the forecasting results are updated so for the next period the optimal decision for the truck routing problem can again be found.

4 Computational Results

In this section we will first discuss some preliminary data analysis, followed by the simulation results.

4.1 Data Analysis

For this case study, the Citi Bike's monthly system data was used from June 2014. In this section a quick overview of the insights derived from the data exploration will be discussed. In this month, New York City had 328 active bike stations with 6.328 different bicycles. In this month, according to the data, there were a total of 936880 trips taken, with a mean of 31.229 trips per day. Friday, 06-06-2014 was the busiest day with 37.719 total trips taken on that day and Friday 13-06 being the least busiest day with 18.308 total trips.

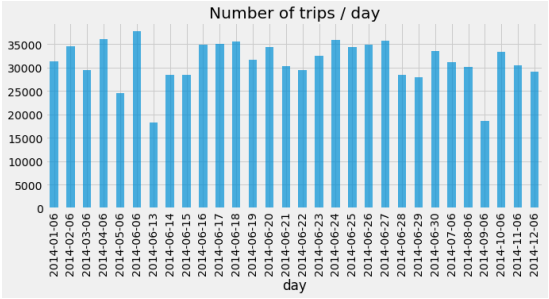

Figure 6: Total number of trips per day of the month

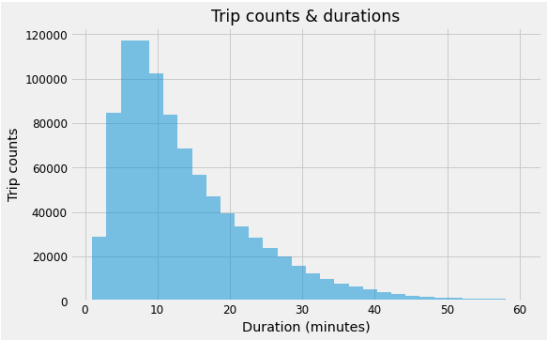

Figure 7: Trip Duration

From figure 7 it can be observed that most of the trips have a short duration, between 5 and 15 minutes. The average trip duration is 14.87 which is just under 15 minutes and the longest trip duration was 359,6 minutes.

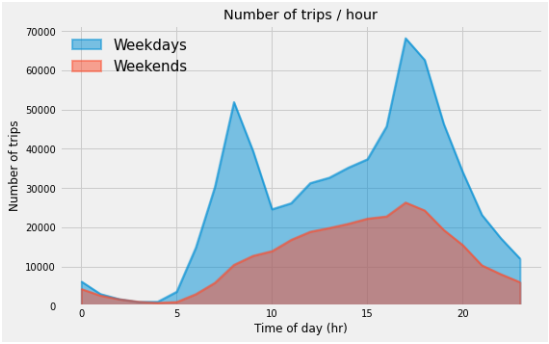

Figure 8: Trips per 24 hours for Weekdays and Weekends

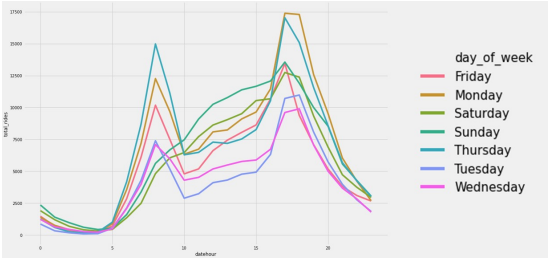

Figure 9: Trips per 24 hours per day

As can be observed from figure 8 and 9, the number of bike rides peaks on weekdays during the morning and afternoon rush hour periods (7am-9am, 4pm-7pm). This can be attributed to the daily commute of people working in the city center, whom travel to work each morning and back home in the afternoon on Monday through Friday. In the weekend days, Saturday and Sunday, the number of trips per hour of the day is much more evenly distributed throughout the day.

Figures 10 and 11 show the capacity change for some bike stations per hour over 48 hour timeframe. It was chosen not to do this for all stations since that would make the graphs unreadable, instead some stations which show interesting capacity change distribution have been chosen to illustrate this. Figure 10 shows that station 135, 168 and 228 have very high positive capacity fluctuations where the capacity spikes in the morning rush hours on both days. This would

imply that these stations have very high docking demand during these hours. Many customers want to place their bicycles at these stations. They also have high negative spikes during the evening rush hours. This implies that during these hours, demand is high for bikes and many customers want to use these stations to pick a bicycle. On the other hand, Figure 11 shows that some stations, for example station 144 and station 245, have a very steady capacity which stays almost constant for the entire timeframe. This would imply that both the demand for bikes and docks remains more or less equal throughout both days.

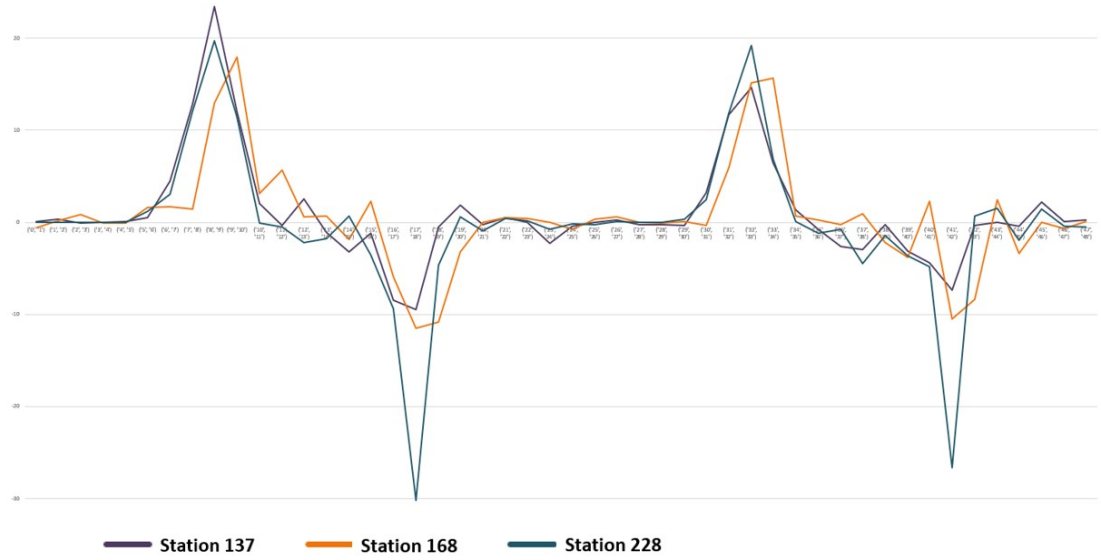

Figure 10: Example of stations with high capacity fluctuations

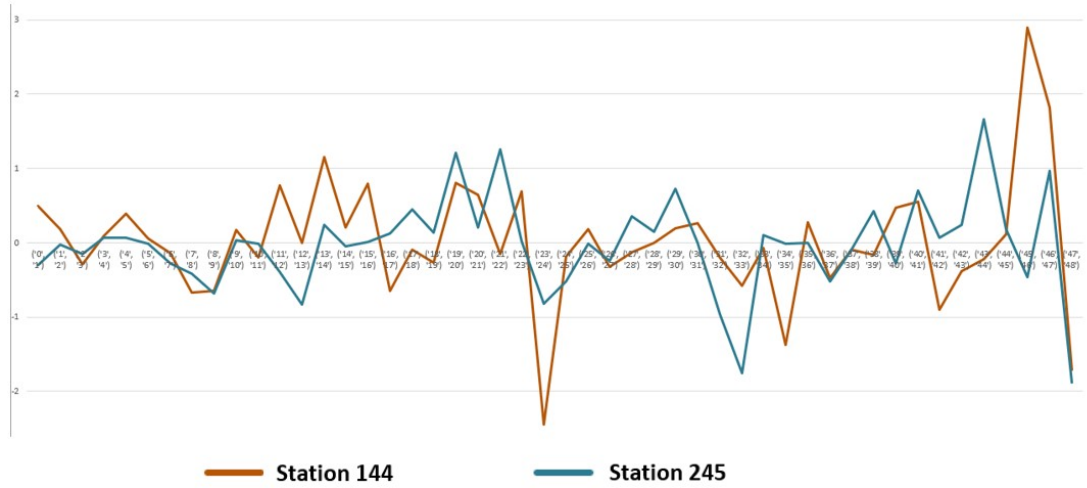

Figure 11: Example of stations with low capacity fluctuations

4.2 Simulation Results

In this section we will discuss results from the benchmark policy compared to our proposed policy.
The performance indicators used to compare both policies are: Performance , percentage of customers served , average bikes per vehicle trip, average vehicle distance traveled, average vehicle travel time. The results are generated through the average of 50 iterations of a 1 day simulated period for all combinations of BUSY_PARAMETER = {8,10,12,15} and TOTALBIKES_PARAMETER = {0.6, 0.7}.

Table 5: Comparison of simulation results between the benchmark model and the proposed model under different parameter values

TOTALBIKES_PARAMETER = 0.6

	Busyness = 8		**Busyness = 10**		**Busyness = 12**		**Busyness = 15**	
	Benchmark	Proposed	Benchmark	Proposed	Benchmark	Proposed	Benchmark	Proposed
Performance	2250.88	5862.84	1472.48	3868.2	1040.56	3191.8	675.12	4481.6
Percentage served	81.18	78.42	85	82.1	87.68	84.6	90.44	88.2
Bikes	7.33	1.99	6.82	3.71	6.3	3.43	5.78	2.16
Vehicle travel time	661.13	429.51	657.3	711.99	641.09	565.83	630.48	324.07
Vehicle distance	666.2	337.62	654.34	478.49	647.16	10.28	640.55	250.84

TOTALBIKES_PARAMETER = 0.7

	Busyness = 8		**Busyness = 10**		**Busyness = 12**		**Busyness = 15**	
	Benchmark	Proposed	Benchmark	Proposed	Benchmark	Proposed	Benchmark	Proposed
Performance	2077.2	5366.6	1375.96	4474.4	999.88	3829.4	659.72	4597.64
Percentage served	84.42	80.6	87.92	85.8	90.3	86.7	92.88	90.1
Bikes	7.82	2.98	7.3	2.71	6.76	3.37	6.2	1.72
Vehicle travel time	646	498.69	663.13	413.76	649.29	686	657.5	338.99
Vehicle distance	658.89	354.71	657.21	286.87	655.01	447.86	644.42	250.73

Table 5 shows the results for the benchmark and proposed policy. Unfortunately, it can be seen that on every test case the benchmark policy performs better than the proposed policy in terms of cost and percentage of customers served. The underperformance of the proposed policy can also be seen in figures **??** and 12. We will try to reflect on why this may have happened. From the table it can be seen that especially the costs in the proposed policy are between approximately 2 to 4 times larger than the benchmark policy, whereas the percentage of served customers is more or less equal. Since the percentage of customers served is considered the most important performance metric in this model the results for the benchmark and proposed policy have been included in Figure 12 using all different parameter settings. From this figure it can be observed that indeed the initial benchmark model scores slightly better than the proposed solution on the percentage level of customer service.

This might indicate that there are many instances where bikes are dropped to a station which causes its capacity to be exceeded. In turn, this leads to a €10 cost. This might be due to a number of reasons, for example:
First of all, the fact that a learning table is used to determine the number of bikes. The system needs to be trained and does so by making mistakes. In a time horizon of only 1 day it has little time to learn from its mistakes. Therefore, the system might behave better if the time horizon is increased. Due to time constraints, only a quick check was possible. In the case of Busyness = 12 and and total_bikes = 0.6 with a duration of 7 days, the Benchmark had a cost of 9379.14 and service rate of 84.4% , and the proposed policy had a cost of 134,249.4 and a service rate of and 62.7%. Unfortunately, increasing the time horizon does not improve performance of the proposed policy.

Second of all, the policy takes decisions based on a demand forecast 1 decision moment away. With an MAE's of 2.48 or 2.38, perhaps a random forest regression may not have been the best idea. It might have been better to treat this issue as a classification problem based on risk of stock out. Each station may be classified as no risk, low risk, medium risk or high risk of stock out and decisions can be based on these classification predictions.

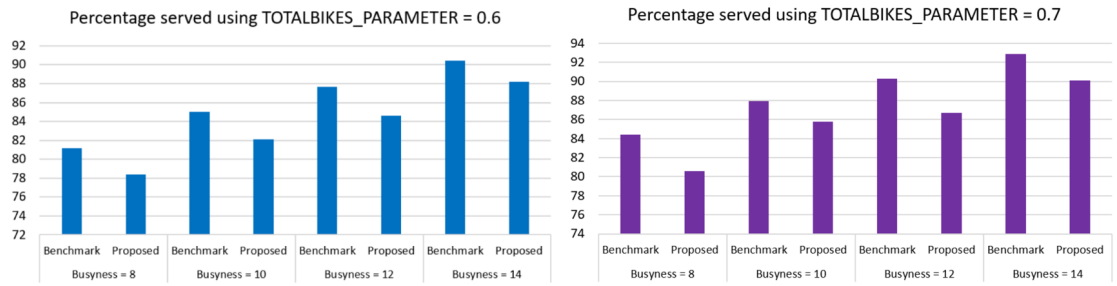

Figure 12: Comparison between the benchmark en proposed model in terms of percentage customers served for different values for busyness using the TOTALBIKES_PARAMETER equal to 0.6 and 0.7, respectively

5 Managerial Results

In this section managerial advice will be provided.

5.1 Managerial advice.

The previous section provided the computational results for the benchmark and implemented policy and reflection on why it performs worse than the benchmark. In this section, managerial advice will be provided. Unfortunately, the proposed solution did not improve the current solution and thus it would not be wise to implement this policy. However, a learning algorithm is often a good strategy for improving current solutions. We believe that with a slightly different approach and with more time to train the model, a better strategy could be found. As for practical implications, they sometimes depend on what management finds important. In this case the customer satisfaction was deemed the most important. If this would be the case, then situations with a vehicle with a larger capacity or maybe even extra vehicles could be deployed in order to increase customer satisfaction levels. Besides customer satisfaction there are other metrics that could also be very important. Firstly, cost reduction is considered to play a vital role in operations management. If the main driver would be cost reduction instead of customer satisfaction the outcome could be different since other KPI's will play a more important role. Customer satisfaction is of course also very important, but it could play less of a role here, since Citibike has a monopoly position on the bike system in New York. If customers are not satisfied with the service level there is not much they can do besides quitting the service. For cost reduction, the KPI's concerned with the vehicle like Average travel time, Average load and the percentage that the vehicle is empty could be more important and policies that explore techniques to optimize these parameters would be very interesting to consider in that situation. In conclusion, for this particular case it would not be wise to implement the proposed policy. However, we believe that it would be very important to explore other options and different models to improve the current situation.

5.2 Implications for further research.

This research has implication for future work where more options to solve this problem could be explored. Interesting ideas for exploration include increasing and decreasing the capacity at some stations. With machine learning techniques on the demand distribution, the demand per station can accurately be forecasted. This means that for busy stations the number of bikes and docks could be increased and for stations that are less busy they could be decreased. It would be interesting to see if the customer service level could be improved while the total number of bikes and docks stays the same. Another idea for future work might include, increasing the number of vehicles and constructing operating zones for each separate vehicle. Determining the zones could be done by creating clusters using a greedy algorithm. In each zone the actual traffic conditions experience could be determined per time of day. In fact, it is well known that a detailed representation of road network congestion is required to assure reliable logistic costs, while the static assumption may often lead to no optimal solutions (Figliozzi, 2010).

References

Dell'Amico, M., Iori, M., Novellani, S., & Subramanian, A. (2018, 12). The bike sharing rebalancing problem with stochastic demands. *Transportation Research Part B Methodological*, *118*, 362-380. doi: 10.1016/j.trb.2018.10.015

Figliozzi, M. (2010, 07). The impacts of congestion on commercial vehicle tour characteristics and costs. *Transportation Research Part E: Logistics and Transportation Review*, *46*, 496-506. doi: 10.1016/j.tre.2009.04.005

Sethi, S., & Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research*, *29*, 387-415. doi: 10.1007/BF022836075