Integer Programming Models to Determine Optimal Rush Hour Bike Allocation: An Empirical Case Study on Boston Bluebikes

Lan Zhang

Abstract

This project aims to formulate integer-programming models that determine the optimal allocation of bikes across stations in a bike-sharing system during rush hour intervals by minimizing total instances of unmet customer demand for available bikes and empty drop-off slots. The results of these integer programs when applied to Bluebikes ridership data inform operators of each station's optimal bike-fill percentage at the beginning of morning and evening rush hours and are heavily correlated with the geographic distribution of Boston's residential and business districts. Additionally, associated sensitivity analysis on binding station capacity parameters can be used to determine optimal system expansion strategies.

1 Introduction

Bike-sharing systems have contributed to a significant transportation paradigm shift in urban areas during the last decade, as they provide cost-effective and sustainable means of traveling between two locations for commuting and tourism purposes. A bike-sharing system is composed of stations scattered across a city where users can take an available bike from any station and drop off the bike to a vacant dock at another station near their desired destination. Bluebikes, Boston's bike-sharing system, has grown from 65 stations and just over 14,000 trips in 2011 to over 365 stations and more than 2 million trips in 2020.[1]

However, with the rise in usage of bike-sharing systems, system imbalance becomes an increasingly prominent issue. Especially during rush hours when large numbers of individuals are traveling to and from work, bikes accumulate at certain stations in the city which then leaves other stations empty. Customer dissatisfaction often arises from two cases caused by asymmetric demand for bikes across the city.[2] In the first case, users who want to rent a bike may be met with fully empty stations and are then required to find other means of transportation. In the second case, customers may find that the station at their desired destination has no empty slots and are then required to end their trip at a different location.

Like all bike-sharing systems, Bluebikes uses 4-5 vans around the clock to rebalance bikes across stations in order to better meet customer demand.[3] Because trucking operations are expensive, about $1 per bike moved, numerous studies have been conducted that cluster stations based on whether they lose, gain, or self-balance bikes over the course of a given day and predict expected demand for bikes at each station using stochastic models such as Markov Chains.[4,5] Results from these studies have helped operators determine where to build new bikeshare stations and how to best allocate bikes in order to meet anticipated user demand.

The goal of this project is to use integer programs (IPs) to determine the optimal number of bikes to place at each bikeshare station at the start of morning and evening rush hours to meet the sharp increases in

[1] https://www.bluebikes.com/system-data
[2] https://d21xlh2maitm24.cloudfront.net/wdc/cabi-2014surveyreport.pdf?mtime=20161206135936
[3] https://www.bluebikes.com/about/media-kit
[4] https://ecommons.cornell.edu/handle/1813/40922
[5] https://downloads.hindawi.com/journals/mpe/2018/8028714.pdf

system demand. I then apply the formulated models to Bluebikes data to determine the optimal state of rush hour bike allocation. Finally, I conduct sensitivity analysis on the model to explore how varying parameters such as the net flow of bikes at a station (which is dependent on both weather and season), the total number of bikes in the system, and the number of docks available at a given station impact the results of the optimization problem.

This paper is organized as follows. In Section 2, I will propose an integer programming model that solves for optimal rush hour bike allocation based on the net flow of bikes over the course of the rush hour. I will also propose an extension to the initial IP that dynamically updates the number of bikes at a given station every n minutes and minimizes the sum of mismatched demand and supply for both bikes and empty docks across all n-minute length intervals in a given rush hour. Finally, in Section 2, I will describe the data collection and cleaning process for Bluebikes data in order to calculate IP parameters before solving. The results of the Bluebikes case study will be presented and examined in Section 3, and I will conduct a sensitivity analysis on my chosen parameters and discuss model shortcomings in Section 4. A conclusion will follow in Section 5.

2 Description of Models & Data

2.1 Integer Program Formulation for Optimal Rush-Hour Bike Allocation

2.1.1 Integer Program Formulation Based on Total Net Flow of Bikes

To determine the optimal allocation of bikes across stations at the start of the morning and evening rush hour periods such that the sum of users who want to rent a bike but are met with empty stations and those who seek to return a bike but are met with full stations is minimized, I formulated an integer program, as the objective function and constraints are linear and all decision variables are integral. In order to estimate the demand of bikes and empty slots at a given station, I calculate the net flow of bikes corresponding to each station from available bikeshare data over the course of both morning and evening rush hours.

Let X_s, the decision variables, equal the number of bikes placed at station s at the beginning of rush hour. The parameters are defined as follows: let B be the total number of bikes in the city-wide system, let C_s be the capacity (total number of docks) at station s, and let $flow_s$ be the net flow of bikes at station s over the course of the rush hour. More explicitly, letting in_s be the total number of bikes coming into station s over the course of a predefined rush-hour period and out_s be the total number of bikes leaving station s over this same rush hour interval, I define $flow_s = in_s - out_s$. Thus, if $flow_s < 0$, more bikes are rented from station s than dropped off, and vice versa for cases where $flow_s > 0$. Additionally, let $S = \{1, 2, ..., N\}$ be the set of stations in the bike-sharing system, with N being the total number of stations.

Thus, $X_s + flow_s$ is the total number of bikes available and $C_s - (X_s + flow_s)$ is the total number of slots available at station s at the end of the rush hour period. Ideally, we want $X_s + flow_s \geq 0$ such that there are enough bikes placed at station s at the beginning of the rush hour to meet the overall demand for bikes, taking into account the total inflow of bikes over the same time period. Additionally, we would also like $C_s - (X_s + flow_s) \geq 0$ such that there are enough slots available for customers who want to drop off bikes at station s, accounting for slots that are emptied due to bikes being rented during the same interval. It follows that the formulated integer program should assign a penalty to all instances where $X_s + flow$ is less than 0 or greater than C_s, as the former would result in $X_s + flow < 0$ and the latter would result in $C_s - (X_s + flow) < 0$, and minimize the sum of these penalty terms across all stations. Let us call the first integer program Model A for ease of future reference. Model A is formulated as follows, with auxiliary variables U_s and O_s capturing the total number of instances where station s has too few bikes or too few slots to meet customer demand, respectively.

$$\begin{aligned}
\text{minimize} \quad & \sum_{s \in S} O_s + U_s \\
\text{s.t.} \quad & \sum_{s \in S} X_s \leq B \\
& X_s \leq C_s, \ \forall s \in S \\
& U_s \geq -(X_s + flow_s), \ \forall s \in S \\
& O_s \geq X_s + flow_s - C_s, \ \forall s \in S \\
& O_s, U_s \geq 0, \ \forall s \in S \\
& X_s \in \mathbb{Z}^{0^+}, \ \forall s \in S
\end{aligned}$$

Note that the auxiliary variables O_s and U_s are lower bounded by 0 because we only want to penalize instances where there is a bike shortage (when $-(X_s + flow_s) > 0$, via the third constraint) or a dock shortage (when $-[C_s - (X_s + flow_s)] = X_s + flow_s - C_s > 0$, via the fourth constraint).

2.1.2 Integer Program Extension Based on Net Flow Calculated at n-Minute Sub-intervals

Because Model A determines the optimal number of bikes to place at each station s based on the total net flow of bikes calculated over the entire rush hour interval, it does not dynamically capture the time variant aspect of the problem. At each point in time during rush hours, the number of bikes at a given station and the net flow of bikes likely change, especially at busy stations. Thus, I chose to extend Model A to minimize the total number of instances in each n-minute subinterval (such that n is significantly smaller than the total length of the rush hour) where the demand for bikes/slots is greater than the number of bikes/slots available. The total number of timesteps M that the extended integer program minimizes over is $M = R/n$, where R is the total number of minutes in the rush hour interval. For example, if we define the morning rush hour to be between 7-10am and choose a subinterval of length $n = 10$ minutes, we would have $R = 180$ minutes and $M = R/n = 18$ time intervals.

The variable definitions for X_s, B, and C_s hold from the previous subsection for the below model. Additionally, let Y_t^s be the number of bikes at station s at timestep t, and let auxiliary variables U_t^s and O_t^s capture the unmet demand in bikes and slots, respectively, at station s at timestep t. Let the set $T = \{1, ..., M\}$ be the total number of time intervals in the rush hour, where $M = R/n$ as defined earlier. The extended integer program, which I will refer to as Model B, is formulated as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{s \in S} \sum_{t \in T} O_t^s + U_t^s \\
\text{s.t.} \quad & \sum_{s \in S} X_s \leq B \\
& X_s \leq C_s, \ \forall s \in S \\
& U_t^s \geq -(Y_t^s + flow_t^s), \ \forall s \in S, \ \forall t \in T \\
& O_t^s \geq Y_t^s + flow_t^s - C_s, \ \forall s \in S, \ \forall t \in T \\
& Y_t^s = X_s + \sum_{i=0}^{t-1} flow_i^s, \ \forall s \in S, \ \forall t \in T \\
& O_t^s, U_t^s \geq 0, \ \forall s \in S, \ \forall t \in T \\
& X_s \in \mathbb{Z}^{0^+}, \ \forall s \in S
\end{aligned}$$

Note that the fifth equality constraint, $Y_t^s = X_s + \sum_{i=0}^{t-1} flow_i^s$ calculates the total number of bikes at station s at timestep t by summing up the net flow at all previous timesteps and adding that sum to the initial allocation of bikes, X_s.

2.2 Boston Bluebikes Data Collection & Cleaning

In order to apply the models formulated in subsection 2.1 to empirical data, I obtained bikeshare data from the Bluebikes website which the company publishes in a CSV format on a monthly basis.[6] For each month, all trips taken throughout the month are recorded, along with the start stations, destination stations, start times, end times, and other user demographic features. I chose to download bikeshare data from 2019 because the pandemic in 2020 has drastically impacted system dynamics in terms of rush hour traffic flow. The distribution of the number of Bluebikes trips taken over the course of a day after March 2020 follows a smoother normal distribution rather than a bi-modal distribution corresponding to peaks in system usage during morning and evening rush hours.[7]

For the empirical case study portion of this project, I chose to use Bluebikes data from July 2019, August 2019, September 2019, and October 2019 to estimate the net flow parameters of the integer programs proposed in the previous section. After concatenating the four datasets, I converted the date and time of each recorded trip into Python `datetime` objects and removed all rides taken on a weekend day using the `datetime` module's associated `DayOfWeek` property.

I defined the hours corresponding to morning and evening rush hours by examining Bluebikes system usage data from 2019, locating the two time intervals where average usage peaked during the day; I defined the morning rush hour period to be between 7-10am and the evening rush hour period to be between 4-7pm.[8] For Model B, I chose to use a 10-minute subinterval for the Boston case study. Ten minutes is a long enough period to obtain estimates for net flow that are robust to large fluctuations, but small enough such that when I divide a pre-defined 180-minute rush hour period into 10 minutes intervals, I obtain 18 timesteps to optimize over. For each station s in the Bluebikes system, I calculated the average net flow of bikes during each 10-minute subinterval for both the morning and evening rush hours. I did so by subtracting the total number of trips starting at station s from the total number of trips ending at station s during each 10-minute subinterval, before dividing this difference by the total number of unique trip dates in the dataset. Then, to get the average net flow of bikes across an entire given rush hour period for Model A, I summed each of the 10-minute subintervals corresponding to a given rush hour period.

3 Results & Analysis

3.1 Optimal Bluebikes Allocation Based on Total Net Flow

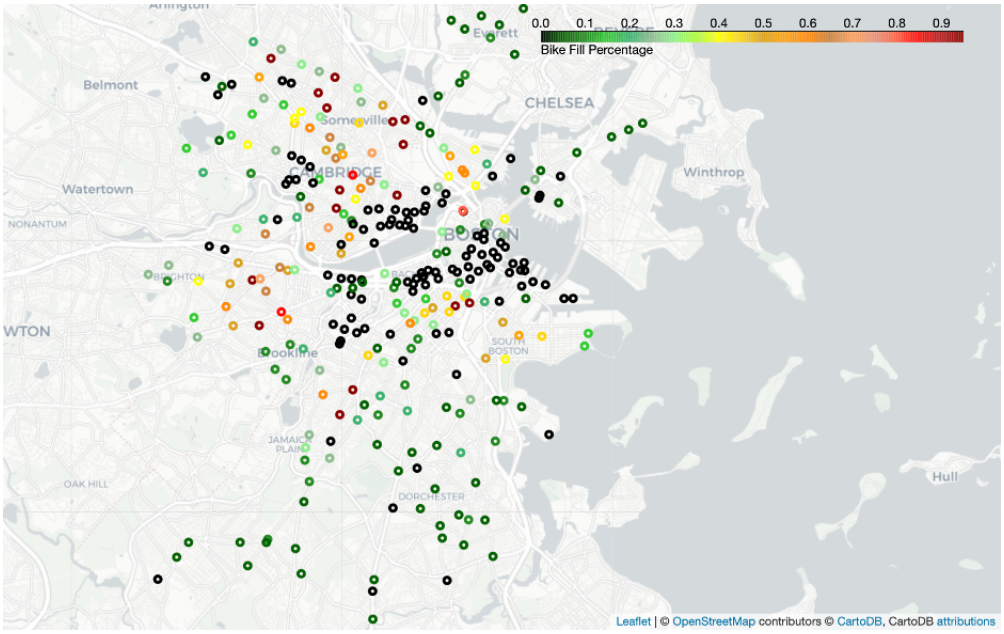
Each month, Bluebikes updates a document listing the number of docks for each station, which allowed me to then accurately set the model parameters C_s (the capacity of each station s).[9] I set B, the total number of bikes in the system, to 2500 bikes as this statistic is also published by Bluebikes at the end of each calendar year. The calculation of the total net flow parameter for each station during the morning and evening rush hours is detailed above in section 2.2.

I implemented and solved all integer programs in the Python extension module of Gurobi, a commercial

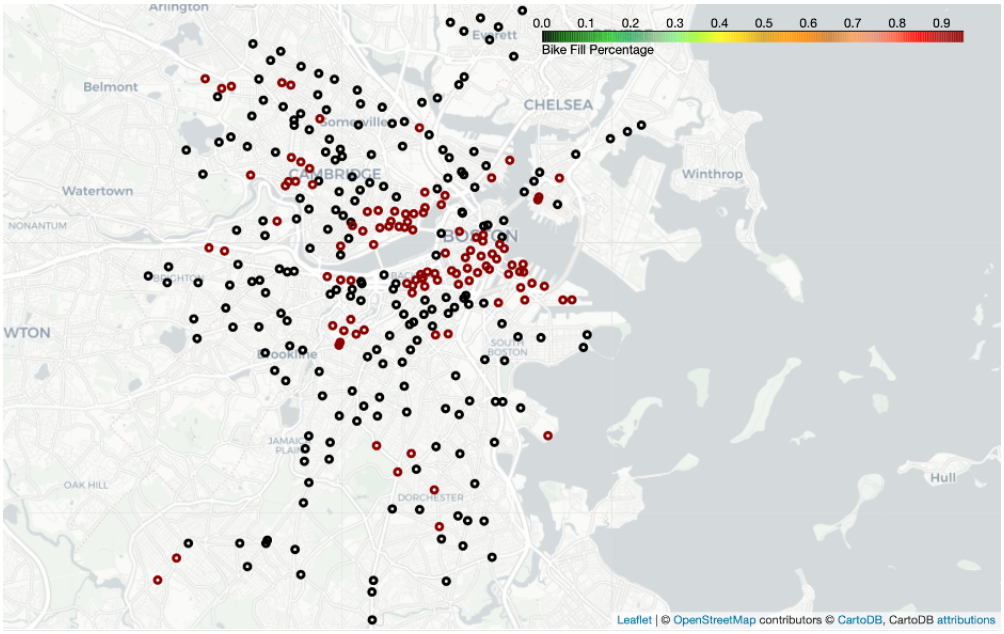
[6] https://s3.amazonaws.com/hubway-data/index.html
[7] https://www.visualization.bike/bluebikes/overview/2020/7/
[8] https://www.visualization.bike/bluebikes/overview/2019/
[9] https://www.bluebikes.com/system-data

optimization solver for linear and mixed integer programming.[10] Upon solving Model A with the respective parameters for both the morning and evening rush hour periods, I obtained the following results displayed in Figure 1. Each circle marker corresponds to a Bluebikes station and its associated color denotes the optimal bike fill percentage for the station at the beginning of the morning (subfigure (a)) and evening (subfigure (b)) rush hour period.

Figure 1: Optimal Bike Allocation at Each Bluebikes Station based on Total Net Flow for Morning vs. Evening Rush Hours



(a) Morning Rush Hour



(b) Evening Rush Hour

Interestingly, the total number of bikes required to achieve the optimal bike allocation for both the morning and evening rush hours is less than $B = 2500$, the total number of bikes in the Bluebikes system. For

[10] https://www.gurobi.com/documentation/9.1/quickstart_mac/cs_python.html

the morning rush hour, a total of 1074 bikes are required in the optimal solution, where the average total unmet demand for bikes is 55.315 bikes and and the average total unmet demand for slots is 194.888 slots (resulting in an objective value of 250.202). For the evening rush hour, a total of 2031 bikes are required in the optimal solution, where the average total unmet demand for bikes is 214.943 bikes and the average total unmet demand for slots is 72.180 slots (resulting in an objective value of 287.124).

Examining the results in Figure 1, we see that the morning rush hour optimal bike allocation matches the geographic distribution of residential versus business districts in the greater Boston area. Bluebikes stations located along the Charles River and coastline have very low optimal fill percentages as many businesses are located in this region; this matches our intuition because commuters are often looking to drop off bikes at these stations before work in the morning. On the other hand, Bluebikes stations in the surrounding neighborhoods require a higher fill percentage as commuters are looking to pick up bikes from these regions in the morning in order to get to their offices located in the business districts. From Figure 1, we are also able to identify stations that lose bikes (the markers colored red/maroon), gain bikes (the markers colored green/black), and self-balance bikes (the markers colored yellow/orange).

However, for the evening rush hour period, the optimal solution assigns to each station either zero bikes (if the associated net flow of bikes at that station, $flow_s$, is positive) or its full capacity worth of bikes (if $flow_s$ is negative). Further examining the numerical morning rush hour solution reveals a similar behavior with respect to stations with positive total net flow but not to stations with negative net flow. By construction of Model A, assigning X_s the value of zero when $flow_s$ is positive makes sense, as the auxiliary variable U_s in the third constraint will always be negative and thus never increase the objective value. Similarly, the auxiliary variable O_s in the fourth constraint is minimized with $X_s = 0$. As for the case of assigning X_s to equal the capacity of station s when $flow_s$ is negative, this optimal solution also makes sense by the construction of Model A as doing so would minimize both the values of auxiliary variables U_s and O_s. However, the total number of bikes assigned across all variables X_s is limited by B (the number of Bluebikes available in the system). Thus the solution of assigning all stations s where $flow_s$ is positive its full capacity worth of bikes may not always be feasible, as evidenced by the different optimal solution obtained for the morning rush hour period using Model A.

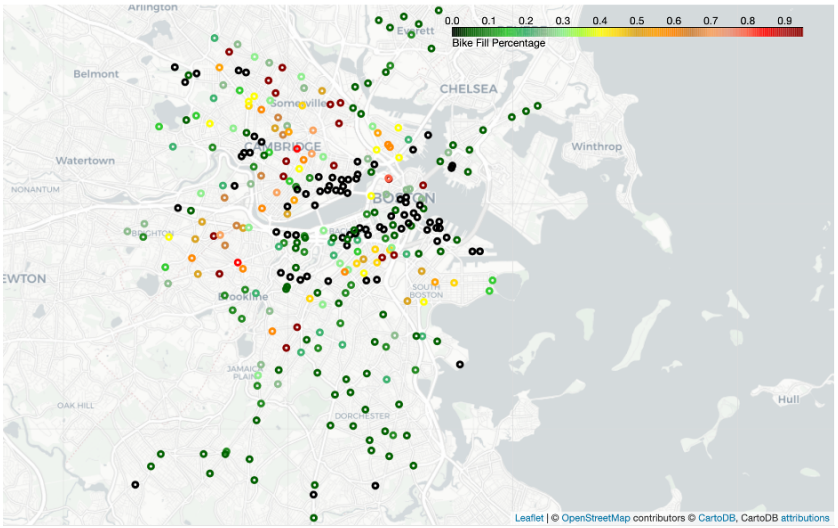
3.2 Optimal Bluebikes Allocation Based on Net Flow Calculated at 10-Minute Intervals

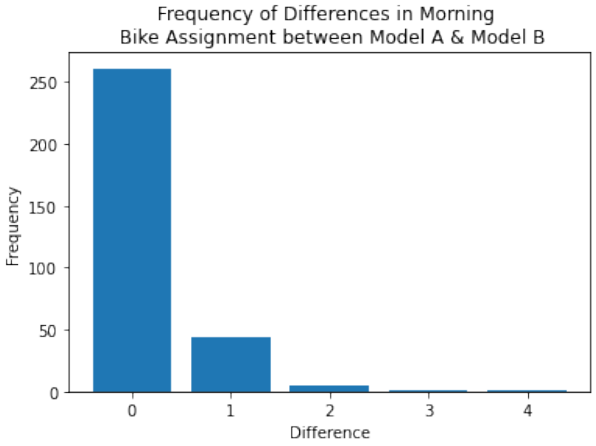
To solve the integer program extension (Model B) detailed in 2.1.2, I used the same parameter values for B (the total number of bikes in the system) and C_s (the capacity of each station s) from subsection 3.1 as published by Bluebikes. For the morning rush hour, the optimal solution returned by the Gurobi solver for the integer program extension had very few differences when compared to that returned by Model A. For each station s, I subtracted the optimal bike allocation for the station returned by Model A from the optimal bike allocation for the station returned by Model B. The optimal bike allocation for 261 stations remained the same, and there were differences of 1 bike for 44 stations, 2 bikes for 5 stations, 3 bikes for 1 station, and 4 bikes for 1 station. Each of these stations with an associated nonzero difference were assigned an allocation of 0 bikes in the optimal solution from Model A. By minimizing the unmet demand for bikes and slots at each 10-minute subinterval, Model B is able to better account for small fluctuations in flow during the rush hour interval than Model A. The frequency of differences between the optimal solutions of the two integer programs is plotted on the right hand side of Figure 2. On the left hand side, we can visually see small differences between Model A's (Figure 1a) and Model B's optimal morning rush hour solution for a few Bluebikes stations along the Charles River/coastline.

For the evening rush hour, the optimal solution for Model B differed drastically from that of Model A.

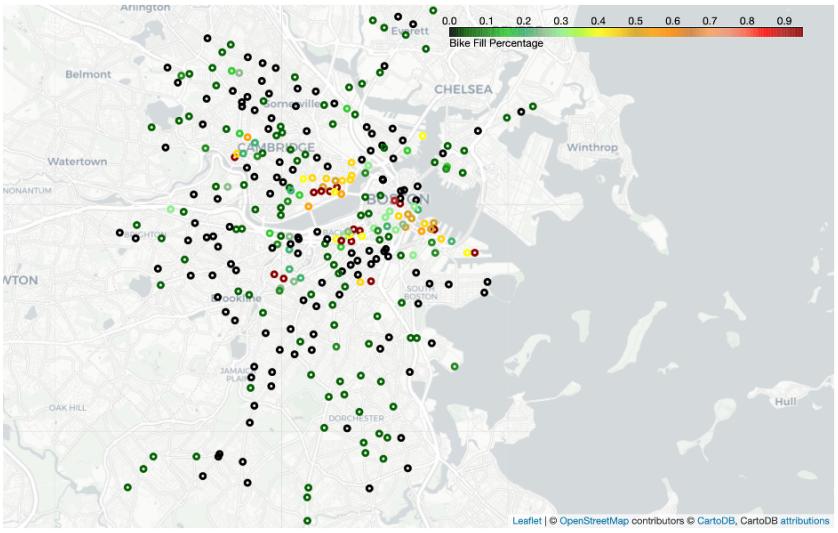
Instead of assigning X_s the value of zero if $flow_s > 0$ and assigning X_s the value of C_s if $flow_s < 0$, we see a much more nuanced optimal bike allocation with Model B as shown in subfigure (b) of Figure 2. Thus, when doing the same comparison between the optimal bike allocation for each station returned by Model A vs. that returned by Model B, we see large negative differences for stations s initially assigned its full capacity worth of bikes and small positive differences for stations s initially assigned 0 bikes. Comparing the morning and evening rush hour optimal solutions of Model B, we see that the geographic distribution of stations that lose and gain bikes are complementary to that of the morning rush hour solution. Bluebikes stations located in business districts (along the Charlies River and coastline) have high optimal fill percentages for the evening rush hour as commuters often take bikes from these locations at the end of the work day. On the other hand, Bluebikes stations in nearby residential districts have low optimal fill percentages as commuters often drop off bikes at these locations during the evening.

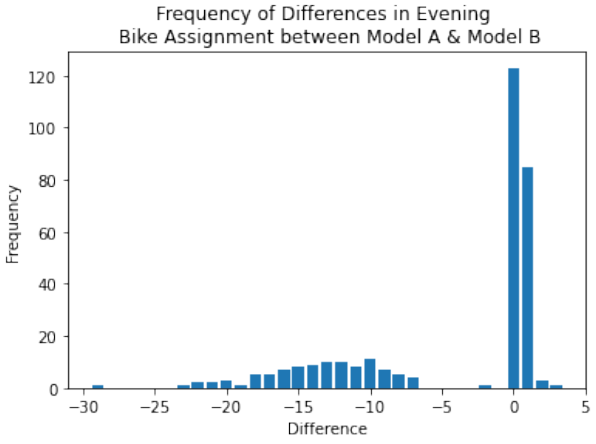
Figure 2: Optimal Bike Allocation at Each Bluebikes Station for Model B during Morning vs. Evening Rush Hours (Left) & The Frequency of Differences in Morning vs. Evening Bike Assignments between Model A and Model B (Right)





(a) Morning Rush Hour





(b) Evening Rush Hour

Interestingly for Model B, the total number of bikes required to achieve the optimal bike allocation for both the morning and evening rush hours is also less than $B = 2500$ (the overall number of Bluebikes in the system). For the morning rush hour, a total of 1135 bikes are required in the optimal solution, where the average total unmet demand for bikes across all timesteps is 464.719 bikes and the average total unmet demand for slots across all timesteps is 1116.629 slots. Thus, we have an optimal objective value of 1581.348. For the evening rush hour, a total of 794 bikes are required in the optimal solution, where the average total unmet demand for bikes across all timesteps is 1576.146 bikes and the average total unmet demand for slots across all timesteps is 731.090 slots. This gives us an optimal objective value of 2307.236. With Model B, we observe a much larger unmet demand for bikes and slots when compared to Model A as we are able to capture fluctuations and misalignments in supply and demand within each 10-minute subinterval rather than across the entire rush hour period.

4 Discussion

4.1 Uniqueness of the Optimal Solution

From the results of Model A discussed in Section 3.1, we see that for the evening rush hour period, the optimal solution consists of assigning a station either zero bikes if its associated net flow is positive or assigning a station its full dock capacity worth of bikes if its associated net flow is negative. Though this solution is optimal, it is likely not unique and points to a model limitation, as the returned optimal solution may be less informative of system dynamics than an alternative optimal solution. In this Bluebikes case study, at least one other optimal solution exists that returns the same objective value but uses fewer bikes by not overassigning bikes to stations with positive net flow. With experimentation, I found that an alternate optimal solution using only 794 bikes can be obtained by limiting the total number of bikes in the Bluebikes system to $B = 2000$, as the solution presented in 3.1 would violate this constraint as it uses a total of 2031 bikes. The plot of this alternate optimal solution on a map can be found in Figure 3 of the appendix, and it much more closely resembles the optimal evening rush hour solution returned by Model B than the previous alternative optimal solution from 3.1. Additionally, a unique optimal solution can be obtained by minimizing the total number of bikes used; this can be done by adding X_s and X_t^s to the objective functions of Model A and Model B respectively, both weighted by some small penalty term such that minimizing unmet demand for bikes and slots still takes precedent.

4.2 Sensitivity to Net Flow Parameters

I examined the sensitivity of the results returned by Model B for the Bluebikes case study to the net flow parameters associated with each station s. Bikeshare ridership is heavily impacted by seasonal weather changes: during warmer months more rides are taken while during colder months, system usage decreases. To calculate the net flow parameters associated with warmer months, I used Bluebikes ridership data from May 2019, June 2019, July 2019, and August 2019. To calculate the net flow parameters associated with colder months, I used Bluebikes ridership data from November 2019, December 2019, January 2020, and February 2020.

For the warmer months, a total of 1016 bikes are needed for the optimal morning rush hour allocation, resulting in an objective value of 955.966 units of unmet demand. A total of 747 bikes are needed for the optimal evening rush hour allocation, resulting in an objective value of 1820.614 units of unmet demand. On the other hand for the colder months, a total of 758 bikes are needed for the optimal morning rush hour allocation, resulting in an objective value of 435.128 units of unmet demand. A total of 595 bikes are needed for the optimal evening rush hour allocation, resulting in an objective value of 509.140 units of unmet demand. We observe that fewer bikes are required during the colder months than the warmer

months to achieve the optimal allocation for both morning and evening rush hours, and that the total unmet demand is also lower for colder months. Upon plotting the results obtained from the two different sets of parameters on a map, which can be found in Figures 4 and 5 of the Appendix, we can see visually that the general geographic variation of optimal rush-hour bike allocation remains similar for both the colder and warmer months, though the optimal fill percentages are lower during colder months.

Additionally in Section 3, I made the assumption when calculating net flow parameters for both Models A and B that the data Bluebikes publishes accurately represents the true demand for bikes and slots. This assumption may not always hold due to censoring: the available data cannot capture instances where a customer may have wanted to borrow a bike but found that no bikes were available at the origin station, as well as instances where a customer may have wanted to end their trip at a specific destination station but found that it was already at full capacity. Thus, the observed data may only represent a lower bound for the true demand of bikes and slots. However, even though the numerical results for Bluebikes rush hour allocation in Section 3 may be affected by censoring, the models proposed in this project can be adapted easily to different datasets and return results that can guide operators to meet user demand.

4.3 Sensitivity to Station Dock Capacity

The only binding parameter in both Models A and B that bikeshare operators have control over is C_s, the dock capacity of station s. I explored the effect of increasing the capacity of stations whose optimal bike assignment by Model B equalled its total capacity (or more formally, stations s where $X_s = C_s$). I chose to focus on Model B for this parameter sensitivity analysis because the impact on the optimal objective value of increasing binding parameters C_s by 1 unit is less straightforward for Model B than for Model A. In the optimal solution returned by Model B for the morning rush hour, three stations are assigned fill levels that equal their dock capacity, stations 3 (359 Broadway – Broadway at Fayette Street), 94 (Community Path at Cedar Street), and 208 (Nashua Street at Red Auerbach Way). For the evening rush hour period, seven stations are assigned optimal fill levels that equal their dock capacity, stations 17 (Ames St at Main St), 141 (HMS/HSPH – Avenue Louis Pasteur at Longwood Ave), 171 (Kendall Street), 172 (Kendall T), 181 (Longwood Ave at Binney St), 184 (MIT Stata Center at Vassar St / Main St), and 242 (Seaport Square – Seaport Blvd at Northern Ave). The resulting optimal objective values (units of unmet demand for bikes/slots) upon increasing each of the binding station capacity parameters by 1 (while holding all other parameters constant) for the morning and evening rush hour models, respectively, are listed in the table below:

Table 1: Effect of Increasing Binding Dock Capacity Parameters of Listed Stations by 1 on Optimal Objective Value (Model B)

Morning Rush Hour

Station(s)	Resulting Obj. Value
None (original)	1581.348
3	1581.348 (no change)
94	1573.573
208	1569.348

Evening Rush Hour

Station(s)	Resulting Obj. Value
None (original)	2307.236
17	2294.236
141	2307.236 (no change)
171	2299.236
172	2301.393
181	2296.236
184	2298.270
242	2294.404

Since building an extra dock is a relatively constant cost across all stations, the results of this sensitivity analysis can help operators prioritize the order of stations to expand. From the results for the "Morning Rush Hour" model, a new dock should be built at station 208 (Nashua Street at Red Auerbach Way) first as doing so decreases the total average unmet demand for bikes (and thus the resulting objective value) by the greatest amount. Similarly, from the results for the "Evening Rush Hour" model, a new dock should be built at station 17 (Ames St at Main St) first since this unit increase would also result in the largest decrease in the optimal objective value. A similar sensitivity analysis can also be completed for all stations assigned an optimal bike fill of zero by Model B during the morning or evening rush hours, as such stations may experience an excess demand for empty slots; increasing the capacity of these stations can potentially further minimize the optimal objective value by decreasing unmet demand for vacant docks.

5 Conclusion

Bikeshare system imbalance is a significant cause of customer dissatisfaction, especially during rush hours. Through two integer programming models that minimize the sum of instances where a customer seeks to rent a bike but finds their desired origin station empty as well as cases where a customers wants to drop off a bike at a full-capacity station, I was able to find an optimal morning and evening rush hour bike allocation across all Bluebikes stations in Boston using historical trip data. The results of both models inform operators of rebalancing strategies that best match customer demand and help them identify which stations lose, gain, and self-balance bikes during a specific rush hour period based on each station's optimal fill percentages. Operators can also prioritize which stations to expand in terms of dock capacity by reoptimizing the proposed integer programs using different parameter combinations.

To expand my current models, several paths for future work can be taken. As mentioned previously, one of the primary limitations of the model is the challenge of estimating the net flow parameter from bikeshare ridership data; because only successful trips are recorded, it would be of interest to estimate the true underlying demand for bikes and slots so the optimal solutions returned by the proposed models better reflect reality. Additionally, since the proposed integer programs are deterministic, it does not capture the randomness in a bikesharing system; future work could explore using Markov chains or other stochastic models to simulate bike movement throughout the system and optimize operations. Finally, although the results of the integer programs in this project inform operators of the optimal number of bikes to place at each bikeshare station before the start of morning and evening rush hours, it does not take into account the operational costs of achieving this assignment that minimizes unmet user demand for bikes and empty docks. In the future, it would be useful to formulate linear and mixed-integer programs to find routes that minimize the cost of transporting bikes, subject to constraints such as van availability and capacity.

6 APPENDIX

Figure 3: Alternative evening rush hour optimal solution to Model A

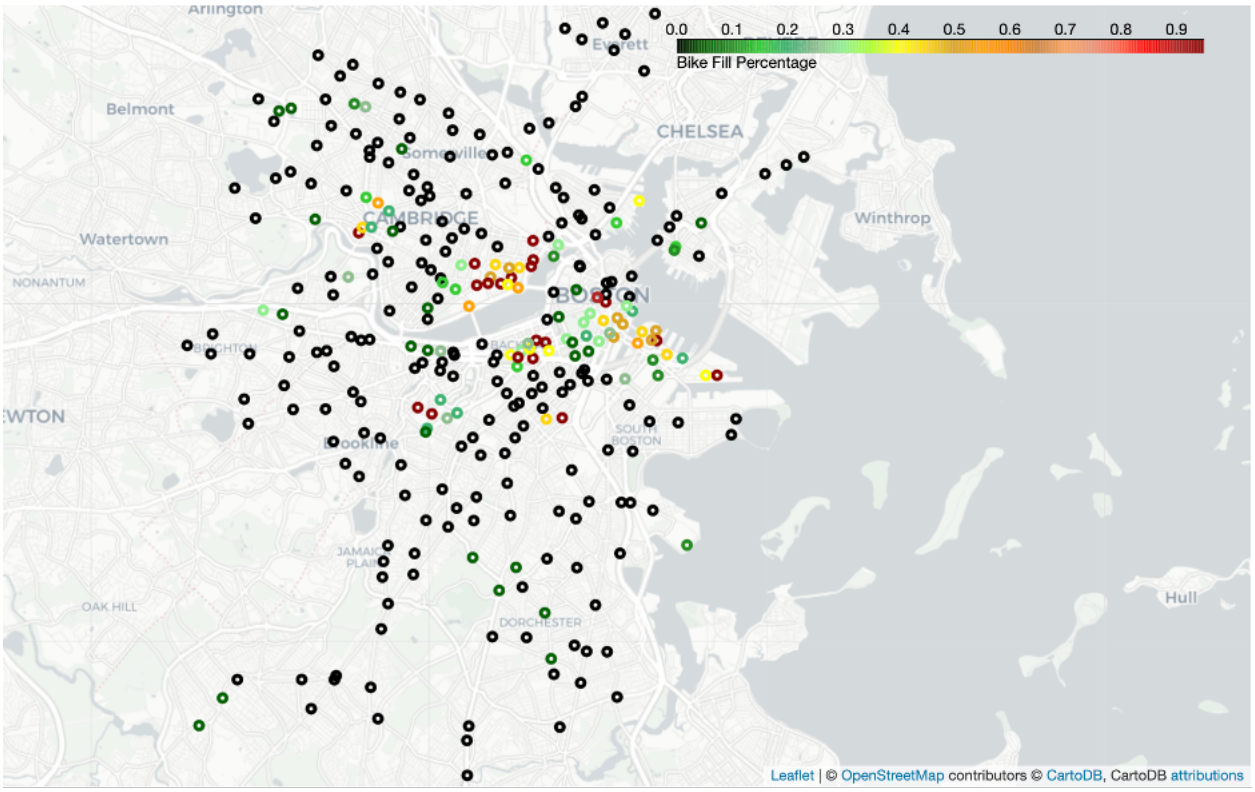
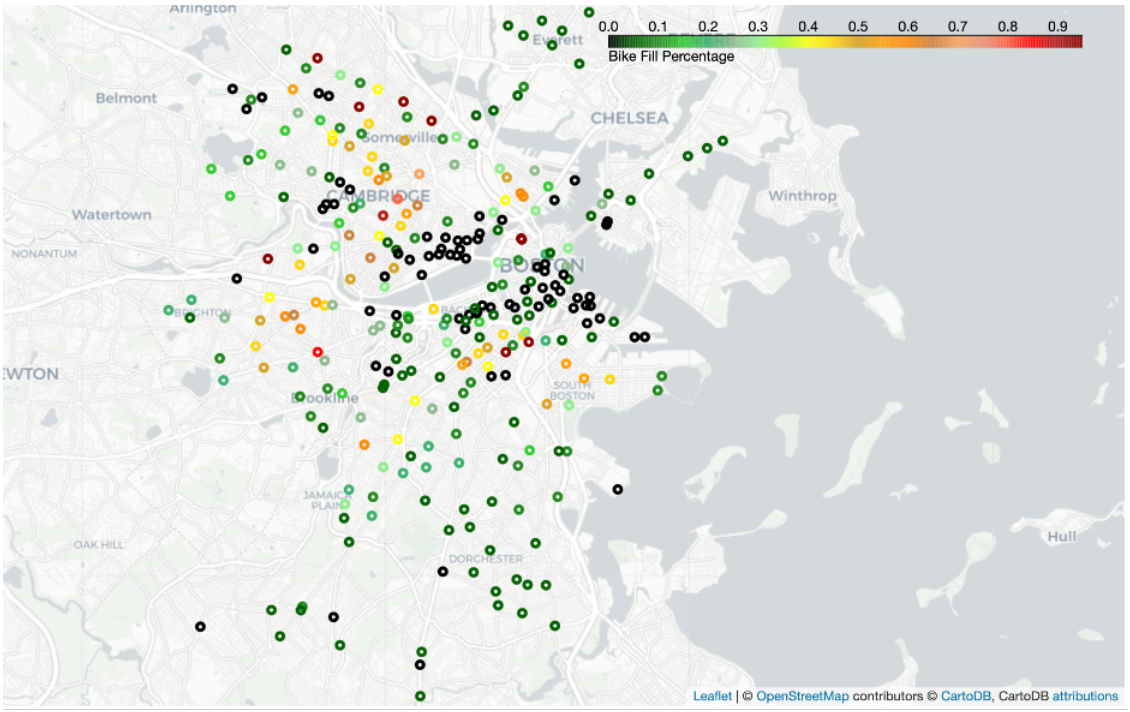
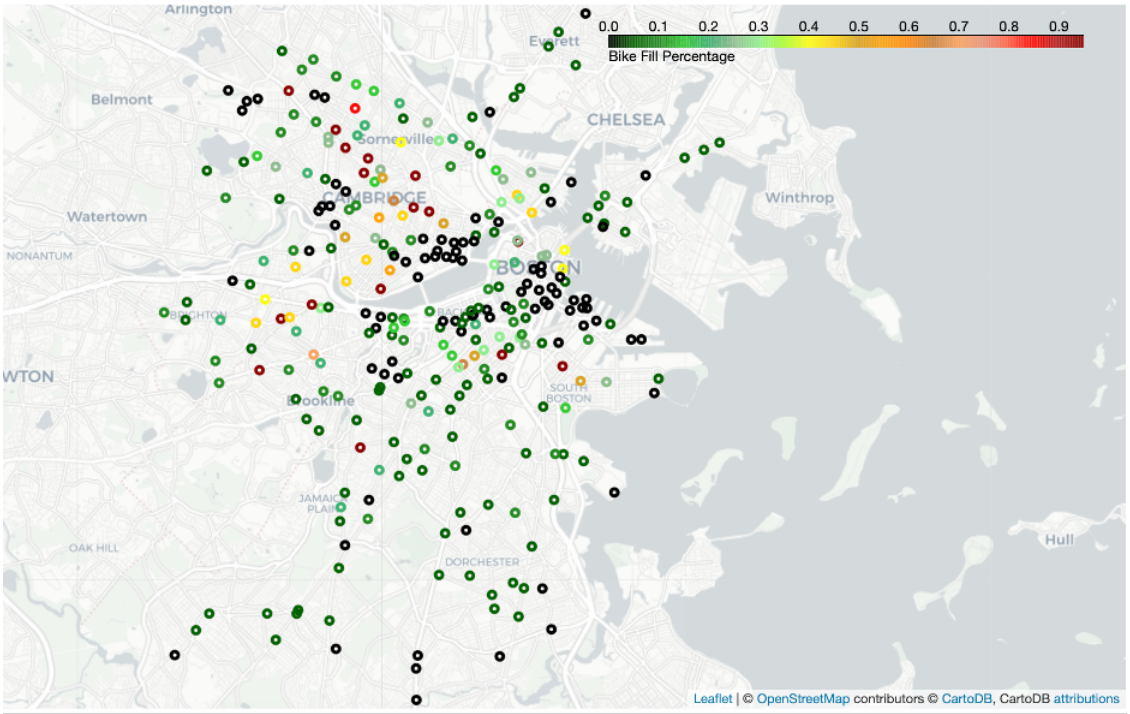


Figure 4: Seasonal effect on optimal Bluebikes allocation (Model B) across all stations during morning rush hours

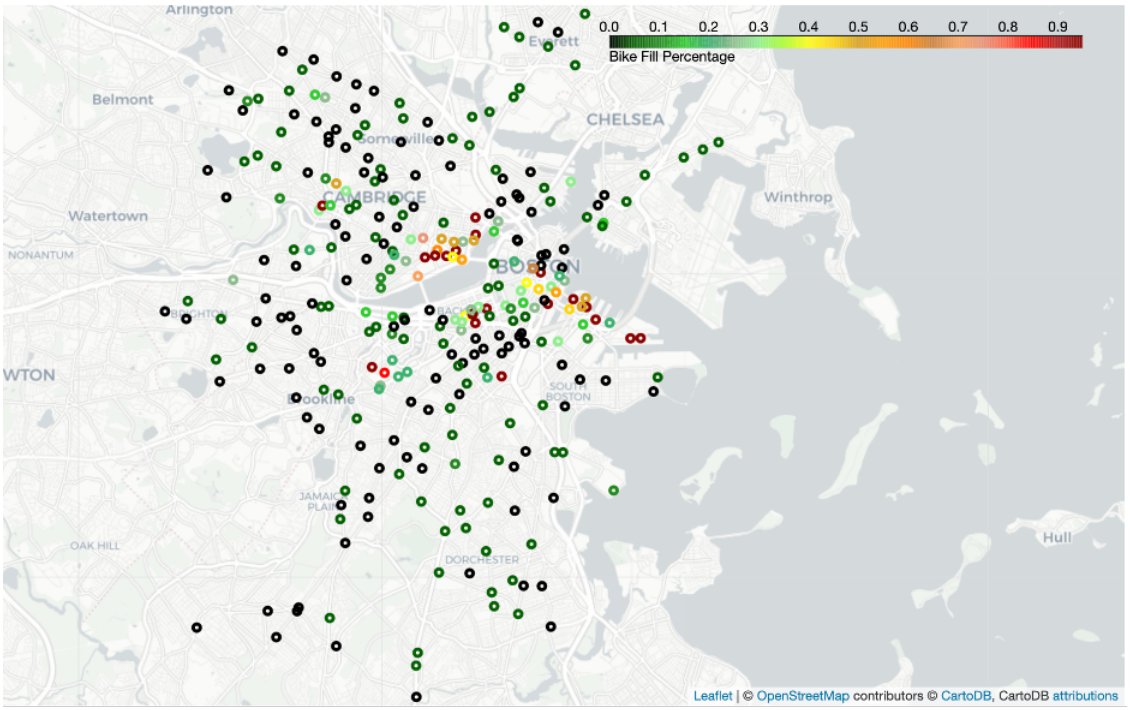


(a) Morning Rush Hour (Warmer Months)

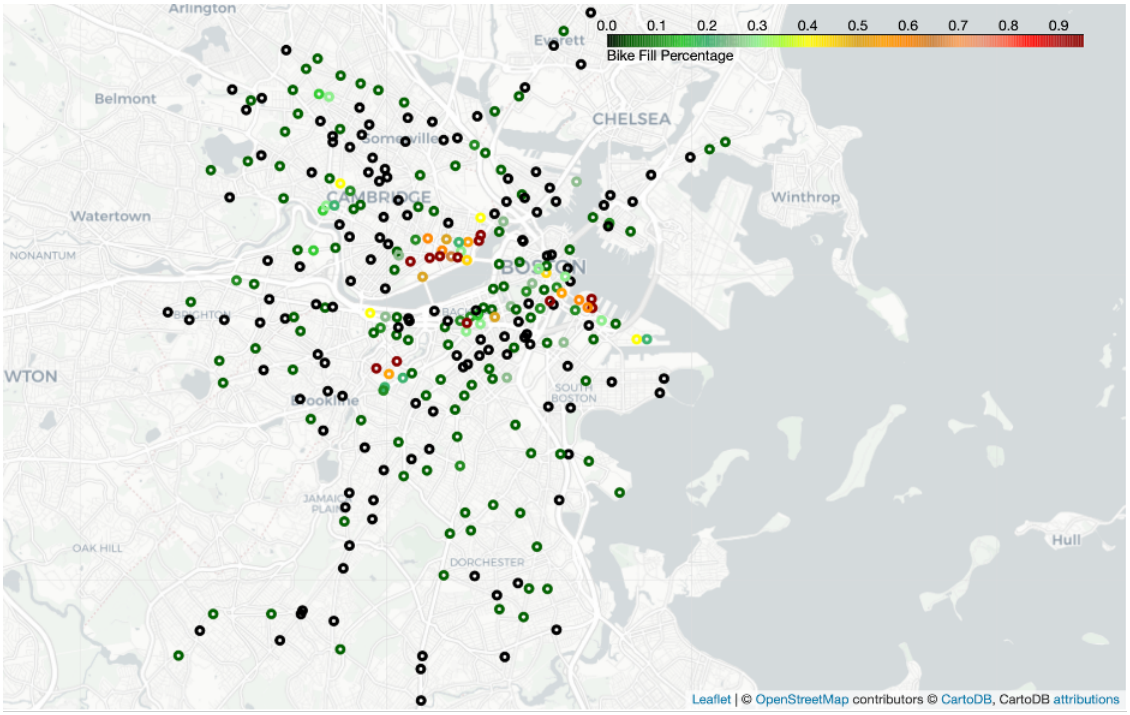


(b) Morning Rush Hour (Colder Months)

Figure 5: Seasonal effect on optimal Bluebikes allocation (Model B) across all stations during evening rush hours



(a) Evening Rush Hour (Warmer Months)



(b) Evening Rush Hour (Colder Months)