Predictive Rebalancing for Bluebikes: A Two-Stage Optimization Approach

Kara Chou, Ray Law, Amber Su

7 December 2025

1 Introduction

As of its launch in 2011, the Bluebikes bike-sharing system has enabled over 27 million trips across the Greater Boston area. With roughly 5,500 bikes and 600 stations, Bluebikes has become an essential component of our transportation infrastructure. Ridership only continues to grow, emphasizing the importance of continuing to improve this system.

Operating a network of this scale comes with many logistical challenges. Notably, demand is highly stochastic—demand varies widely by time of day and location, leading to chronic station imbalances. Some stations are regularly empty, leaving riders without available bikes. Other stations are completely full, preventing riders from docking at their destination. These imbalances result in substantial user dissatisfaction and directly translate into lost ridership.

Although bikeshare operators increasingly incorporate demand forecasting and planned rebalancing, managing station imbalance remains challenging due to the stochastic and spatially uneven nature of rider flows. In reality, from our experiences, there still exists multiple instances throughout the day where user demands cannot be satisfied. This project develops a pre-operation, predictive rebalancing schedule formulated as a two-stage optimization problem. The goal is to determine initial bike allocations and truck routing plans that minimize user dissatisfaction while limiting rebalancing costs, providing a principled baseline before real-time adjustments occur.

By analyzing historical demand patterns and modeling station-level imbalances, we seek to generate an optimized rebalancing plan improving system reliability while reducing unnecessary labor and transit costs. Ultimately, our goal is to provide a data-driven framework that supports a more efficient and dependable Bluebikes system.

2 Data

We are using the publicly available Bluebikes System Data (`bluebikes.com/system-data`), which provides comprehensive historical information about bike-sharing activity.

- **Bluebikes Comprehensive Trip Histories:** This primary dataset provides granular records for all genuine user trips. Crucially, it includes start and end times, duration, and the unique station and bike IDs. The system's filtering of short (< 60 seconds) or maintenance-related trips ensures a clean view of rider behavior. This data is essential for estimating demand and netflow across the network.
- **Bluebikes Stations Dataset:** This static data provides the physical infrastructure details necessary for capacity planning, including station name, ID, geographic coordinates, and, most importantly, the total dock capacity. These capacities serve as the bounds for our Subproblem 1 feasibility constraints. We use the geographic coordinates information from this dataset to calculate the distances between stations for our Subproblem 2, coupled with the routing algorithm in Open Street Map.

- **Bluebikes Real-time Data (GBFS feed):** The real-time General Bikeshare Feed Specification (GBFS) is crucial for acquiring the initial bike inventory at each station at the designated rebalancing start time. This provides the ground truth required for Subproblem 2 to calculate the necessary bike movements.

The combination of historical demand and real-time inventory is the backbone of our two-stage approach.

3 Formulation

3.1 Subproblem 1

This subproblem solves for the optimal number of initial bikes at each station to minimize *user dissatisfaction* and *netflow deviation*.

Parameters

- Let the total number of stations be S.
- Let the total number of timesteps be T.
- Let the total number of bikes be B.
- Station In-Weight and Out-Weight (W_i^{in} and W_i^{out}): For each station $i = 1, \ldots, S$, let w_i^{in} and w_i^{out} be the average total daily rides into and out of station i, respectively. The weights are defined as

$$W_i^{in} = \frac{w_i^{in}}{\sum_{k=1}^{S} w_k^{in}}, \qquad W_i^{out} = \frac{w_i^{out}}{\sum_{k=1}^{S} w_k^{out}}.$$

- Station Capacity $C_i \in \mathbb{Z}_{\geq 0}$: The total number of docks at a given station i.
- Netflow $N_{it} \in \mathbb{R}$: The estimated netflow (total rides into station minus total rides out of station) at a station i during time period t.
- **Hyperparameter:** $\lambda > 0$ is a weight parameter that controls how strongly we penalize deviations from the expected netflow.

Decision Variables

- Let $b_{i1} \in \mathbb{Z}$ be the number of bikes at station i for $i = 1, \ldots, S$ during time $t = 1$.
- Let $b_{it} \in \mathbb{R}$ be the number bikes at station i during time t for $i = 1, \ldots, S$ and $t = 2, \ldots, T$.
- Let $n_{it} \in \mathbb{R}$ represent the feasible netflow at station i during time t to ensure the number of bikes at station i is feasible.
- Let $z_{it} \in \mathbb{R}_{\geq 0}$ represent the deviation from the derived netflow n_{it} from the expected netflow N_{it}.
- Let $x_{it}^{in} \in \{0, 1\}$ represent whether or not station i is "nearly full" ($> 90\%$ full) at time t.
- Let $x_{it}^{out} \in \{0, 1\}$ represent whether or not station i is "nearly empty" ($< 10\%$ full) at time t.

Constraints

- The sum of the initial number of bikes at each station must equal B.

$$\sum_{i=1}^{S} b_{i1} = B.$$

- The number of bikes at each station must always be within station capacity.

$$0 \leq b_{it} \leq C_i \quad \forall i = 1, \ldots, S, t = 1, \ldots, T.$$

- The number of bikes at each station must accurately represent the netflow.

$$b_{i(t+1)} = b_{it} + n_{it} \quad \forall i = 1, \ldots, S, t = 1, \ldots, T-1.$$

- By construction,

$$z_{it} \geq n_{it} - N_{it} \quad \forall i = 1, \ldots, S, t = 1, \ldots, T$$
$$z_{it} \geq N_{it} - n_{it} \quad \forall i = 1, \ldots, S, t = 1, \ldots, T$$

- A station is nearly full if the remaining capacity is ≤ 0.1.

$$x_{it}^{in} \geq 0.1 - \left(1 - \frac{b_{it}}{C_i}\right) \forall i = 1, \ldots, S, t = 1, \ldots, T$$

- A station is nearly empty if its capacity is ≤ 0.1.

$$x_{it}^{out} \geq 0.1 - \left(\frac{b_{it}}{C_i}\right) \forall i = 1, \ldots, S, t = 1, \ldots, T$$

- We also must satisfy our construction constraints of our decision variables.

$$b_{i1} \in \mathbb{Z} \; \forall i = 1, \ldots, S$$
$$b_{it} \in \mathbb{R} \; \forall i = 1, \ldots, S, t = 2, \ldots, T$$
$$n_{it} \in \mathbb{R} \; \forall i = 1, \ldots, S, t = 1, \ldots, T$$
$$z_{it} \in \mathbb{R}_{\geq 0} \; \forall i = 1, \ldots, S, t = 1, \ldots, T$$
$$x_{it}^{in}, x_{it}^{out} \in \{0, 1\} \quad \forall i = 1, \ldots, S, t = 1, \ldots, T$$

Objective Function: Let "User Dissatisfaction" be the sum over all stations of the number of time steps where a station is nearly empty (e.g. 10% full) or nearly full (e.g. 90% full), weighted by their station in/out weight. This represents potential unmet demand due to a station being too full/empty to allow for docking/riding. We want to minimize user dissatisfaction. At the same time, we want to ensure our feasible netflow (where the number of bikes at each station are always within capacity) do not deviate too much from the estimated netflow. Therefore, we also add a penalization term for the deviation, weighted by our hyperparameter λ. Formally, our objective function is as follows:

$$\text{minimize} \sum_{t=1}^{T} \sum_{i=1}^{S} (W_i^{in} x_{it}^{in} + W_i^{out} x_{it}^{out} + \lambda z_{it}).$$

Formulation: We can summarize this formulation as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T}\sum_{i=1}^{S}(W_i^{in}x_{it}^{in} + W_i^{out}x_{it}^{out} + \lambda z_{it}) \\
\text{subject to} \quad & \sum_{i=1}^{S} b_{i1} = B \\
& 0 \le b_{it} \le C_{it} && \forall i = 1,\dots,S, t = 1,\dots,T \\
& b_{i(t+1)} = b_{it} + n_{it} && \forall i = 1,\dots,S, t = 1,\dots,T-1 \\
& z_{it} \ge n_{it} - N_{it} && \forall i = 1,\dots,S, t = 1,\dots,T \\
& z_{it} \ge N_{it} - n_{it} && \forall i = 1,\dots,S, t = 1,\dots,T \\
& x_{it}^{in} \ge 0.1 - \left(1 - \frac{b_{it}}{C_i}\right) && \forall i = 1,\dots,S, t = 1,\dots,T \\
& x_{it}^{out} \ge 0.1 - \left(\frac{b_{it}}{C_i}\right) && \forall i = 1,\dots,S, t = 1,\dots,T \\
& b_{i1} \in \mathbb{Z} && \forall i = 1,\dots,S \\
& b_{it} \in \mathbb{R} && \forall i = 1,\dots,S, t = 2,\dots,T \\
& n_{it} \in \mathbb{R} && \forall i = 1,\dots,S, t = 1,\dots,T \\
& z_{it} \in \mathbb{R}_{\ge 0} && \forall i = 1,\dots,S, t = 1,\dots,T \\
& x_{it}^{in}, x_{it}^{out} \in \{0,1\} && \forall i = 1,\dots,S, t = 1,\dots,T
\end{aligned}$$

3.2 Subproblem 2

Given the optimal number of initial bikes at each station to minimize *user dissatisfaction* and *netflow deviation* from subproblem 1, this subproblem seeks to achieve the desired inventory configuration as closely as possible while simultaneously minimizing the operational cost of rebalancing.

Setup

We model the Bluebikes system as a directed graph $G = (V, E)$.

- Nodes: Each Bluebike station corresponds to a node $i \in N = \{1, \dots, n\}$. We also include a depot node 0 where trucks start and end their routes. $V = \{0\} \cup N$.
- Edges: There is a directed edge between two stations if they are within driving range R:

  $$E_d = \{(i,j) \in V_b \times V_b, i \neq j, d_{ij} \le R\},$$

  where d_{ij} denotes the distance between station b_i and station b_j. Additionally, each station is connected to the depot:

  $$E = E_d \cup \{(0,i),(i,0), \forall i \in 1,\dots,n\}.$$

Parameters

- d_{ij}: distance between stations b_i and b_j
- c: unit cost per distance traveled by a truck
- m: unit cost for loading or unloading one bike
- C: number of Bluebikes each truck can carry at a time
- K: total number of trucks
- i_i: the initial number of bikes at station b_i
- e_i: the desired number of bikes at station b_i, from Subproblem 1

- V: the maximum number of times that a station can be visited
- **Hyperparameter:** $\lambda > 0$ is a weight parameter that controls how strongly we penalize deviations from the optimal bike initializations.

Decision Variables

- Let $x_{ijk} \in \{0, 1\}$ represent whether or not truck k traveled from station i to j.
- Let $y^+_{ik} \in \mathbb{Z}_{\geq 0}$ represent the number of bikes being dropped off at station i by truck k.
- Let $y^-_{ik} \in \mathbb{Z}_{\geq 0}$ represent the number of bikes being piked up at station i by truck k.
- Let $z_{ijk} \in \mathbb{Z}_{\geq 0}$ represent the number of bikes on truck k from station i to station j.
- Let $v_{ik} \in \{0, 1\}$ represent whether or not truck k visited from station i.
- Let $u_{ik} \in \mathbb{R}, 0 \leq u_{ik} \leq n$ represent the relative ordering of station i visited by truck k.
- Let $\delta^+_i, \delta^-_i \in \mathbb{R}_{\geq 0}$ represent slack variables measuring initializations that trucks cannot reach

Constraints

- Each truck must start and end at the depot:

$$\sum_{i \in N} x_{0ik} = 1, \quad \sum_{i \in N} x_{i0k} = 1 \quad \forall k = 1, \ldots, K$$

- For every station and every truck, inbound edges must equal outbound edges:

$$\sum_{j \in V:(i,j) \in E} x_{ijk} = \sum_{j \in V:(j,i) \in E} x_{jik} \quad \forall i = 1, \ldots, n, \ \forall k = 1, \ldots, K$$

- If a truck leaves station i, then it must have visited station i:

$$v_{ik} = \sum_{j \in V:(i,j) \in E} x_{ijk} \quad \forall i = 1, \ldots, n, \ \forall k = 1, \ldots, K$$

- No station can be visited by more than V truck:

$$\sum_{k=1}^{K} v_{ik} \leq V \quad \forall i = 1, \ldots, n$$

- To prevent subtours in the route constructed by each truck, we introduce a visit-ordering constraint, a station may only receive a positive ordering value if it is visited by truck k:

$$u_{ik} \leq n \cdot v_{ik} \quad \forall i = 1, \ldots, n, \ \forall k = 1, \ldots, K$$

- To eliminate subtours among stations for each truck, we impose the Miller–Tucker–Zemlin (MTZ) constraints. For every edge (i, j) (excluding the depot), and for truck k, if truck k traverses edge (i, j), then we force station j to appear strictly later in the visit order.

$$u_{ik} - u_{jk} + n \cdot x_{ijk} \leq n - 1 \quad \forall (i, j) \in E, \ \forall k = 1, \ldots, K$$

- If a truck doesn't visit station i it cannot pick up or drop off bikes:

$$0 \leq y^+_{ik} \leq C\, v_{ik}, \quad 0 \leq y^-_{ik} \leq C\, v_{ik} \quad \forall i = 1, \ldots, n, \ \forall k = 1, \ldots, K$$

- For each station's final bike count should match its target e_i, allowing deviations only through slack variables that are penalized in the objective.

$$i_i + \sum_{k=1}^{K} y^+_{ik} - \sum_{k=1}^{K} y^+_{ik} + \delta^+_i - \delta^-_i = e_i \quad \forall i = 1, \ldots, n$$

- The number of bikes on a truck on edge (i, j) can only be positive if the edge (i, j) is traveled by truck k, and can be maximum at truck capacity.

$$0 \leq z_{ijk} \leq C\, x_{ijk} \quad \forall (i,j) \in E,\ \forall k = 1, \ldots, K$$

- Bike-flow conservation on truck k at station i by requiring that the number of bikes arriving on the truck minus the number dropped off plus the number picked up must equal the number of bikes departing on the truck.

$$\sum_{j:(j,i)\in E} z_{jik} - y^+_{ik} + y^-_{ik} = \sum_{j:(i,j)\in E} z_{ijk} \quad \forall i = 1, \ldots, n,\ \forall k = 1, \ldots, K$$

- We must also satisfy our construction constraints of our decision variables.

$$x_{ijk} \in \{0, 1\} \quad \forall (i,j) \in E,\ \forall k = 1, \ldots, K$$

$$v_{ik} \in \{0, 1\} \quad \forall i = 1, \ldots, n,\ \forall k = 1, \ldots, K$$

$$y^+_{ik}, y^-_{ik} \in \mathbb{Z}_{\geq 0} \quad \forall i = 1, \ldots, n,\ \forall k = 1, \ldots, K$$

$$z_{ijk} \in \mathbb{Z}_{\geq 0} \quad \forall (i,j) \in E,\ \forall k = 1, \ldots, K$$

$$\delta^+_i, \delta^-_i \in \mathbb{R}_{\geq 0} \quad \forall i = 1, \ldots, n$$

$$u_{ik} \in \mathbb{R}, 0 \leq u_{ik} \leq n \quad \forall i = 1, \ldots, n,\ \forall k = 1, \ldots, K$$

Objective Function: Our goal is to minimize both the total cost of rebalancing and the gap between the achieved station status and ideal station status from the frist subproblem. This includes minimizing:

- Truck Driving Cost: $\sum_{k=1}^{K} \sum_{(i,j)\in E} cd_{ij} x_{ijk}$
- Loading/Unloading Cost: $\sum_{k=1}^{K} \sum_{i \in N} m(y^+_{ik} + y^-_{ik})$
- Slack penalty: If trucks cannot achieve target inventory e_i, we penalize the deviation by a factor of λ, where a larger λ represents a stronger enforcement: $\lambda \sum_{i=1}^{K} (\delta^+_i + \delta^-_i)$.

Formally, our objective function is as follows:

$$\text{minimize} \sum_{k=1}^{K} \sum_{(i,j)\in E} cd_{ij} x_{ijk} + \sum_{k=1}^{K} \sum_{i \in N} m(y^+_{ik} + y^-_{ik}) + \lambda \sum_{i=1}^{K} (\delta^+_i + \delta^-_i)$$

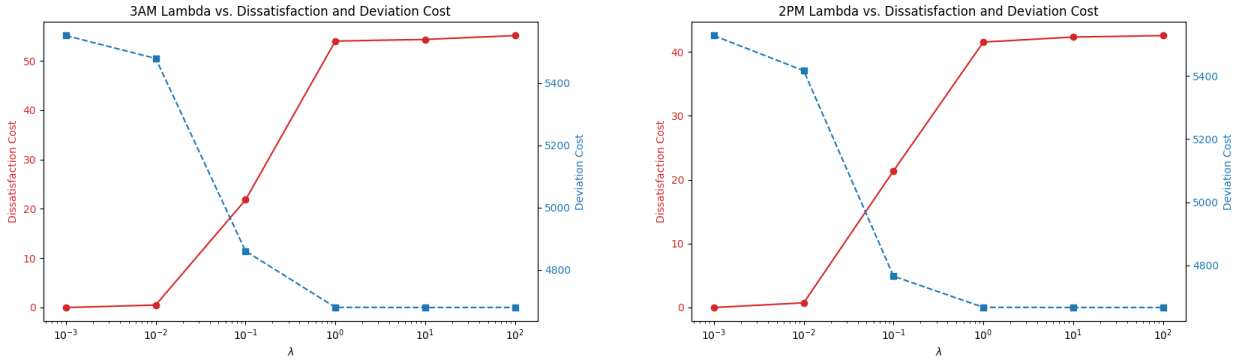
Formulation: We can summarize this formulation as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{K} \sum_{(i,j)\in E} cd_{ij} x_{ijk} + \sum_{k=1}^{K} \sum_{i\in N} m(y_{ik}^{+} + y_{ik}^{-}) + \lambda \sum_{i=1}^{K} (\delta_i^{+} + \delta_i^{-}) \\
\text{subject to} \quad & \sum_{i\in N} x_{0ik} = 1 && \forall k = 1,\ldots,K \\
& \sum_{i\in N} x_{i0k} = 1 && \forall k = 1,\ldots,K \\
& \sum_{j\in V:(i,j)\in E} x_{ijk} = \sum_{j\in V:(j,i)\in E} x_{jik} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& v_{ik} = \sum_{j\in V:(i,j)\in E} x_{ijk} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& \sum_{k=1}^{K} v_{ik} \le V && \forall i = 1,\ldots,n \\
& u_{ik} \le n \cdot v_{ik} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& u_{ik} - u_{jk} + n \cdot x_{ijk} \le n - 1 && \forall (i,j) \in E,\ \forall k = 1,\ldots,K \\
& 0 \le y_{ik}^{+} \le C\, v_{ik} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& 0 \le y_{ik}^{-} \le C\, v_{ik} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& i_i + \sum_{k=1}^{K} y_{ik}^{+} - \sum_{k=1}^{K} y_{ik}^{+} + \delta_i^{+} - \delta_i^{-} = e_i && \forall i = 1,\ldots,n \\
& 0 \le z_{ijk} \le C\, x_{ijk} && \forall (i,j) \in E,\ \forall k = 1,\ldots,K \\
& \sum_{j:(j,i)\in E} z_{jik} - y_{ik}^{+} + y_{ik}^{-} = \sum_{j:(i,j)\in E} z_{ijk} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& x_{ijk} \in \{0,1\} && \forall (i,j) \in E,\ \forall k = 1,\ldots,K \\
& v_{ik} \in \{0,1\} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& y_{ik}^{+}, y_{ik}^{-} \in \mathbb{Z}_{\ge 0} && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K \\
& z_{ijk} \in \mathbb{Z}_{\ge 0} && \forall (i,j) \in E,\ \forall k = 1,\ldots,K \\
& \delta_i^{+}, \delta_i^{-} \in \mathbb{R}_{\ge 0} && \forall i = 1,\ldots,n \\
& u_{ik} \in \mathbb{R}, 0 \le u_{ik} \le n && \forall i = 1,\ldots,n,\ \forall k = 1,\ldots,K
\end{aligned}$$

4 Results

4.1 Subproblem 1

We ran the model 12 times, using 10-minute buckets, six test values of lambda ($\lambda = 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}$) and two initial times (3 AM and 2 PM). The time limit of each run is set to 90 minutes. The resulting costs per lambda are shown below:



(a) 3AM **(b) 2PM**

Figure 1: λ vs. Dissatisfaction and Deviation Cost for Two Time Periods

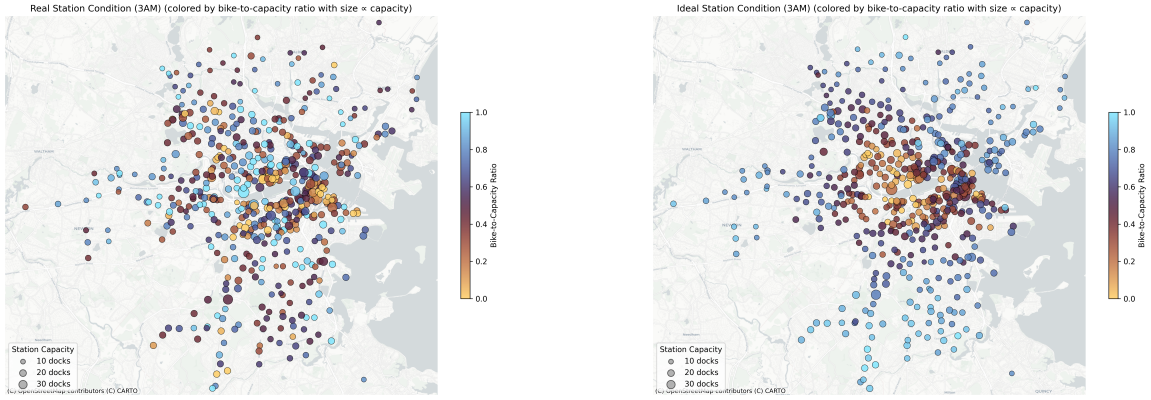
To generate more accurate results, the model should prioritize minimizing deviation cost because deviation cost measures how much the feasible net flow differs from the estimated net flow. Thus, higher lambda values should be prioritized. As shown above, when lambda is at least 1, the dissatisfaction cost levels off, so this is the limit for the model's accuracy.

The ideal ratio between the initial bike distribution and the capacity of each station at 3 AM (for $\lambda = 1$), alongside the actual ratio pulled from live Bluebikes data, is shown in Figure 2. The model fills stations in residential areas and empties stations in work areas, which aligns with expected travel patterns: riders typically leave residential neighborhoods in the morning and head toward work areas, then return in the evening.

The following table shows the costs associated with both distributions. In both runs, we used 10-minute buckets, an initial time of 3AM, and $\lambda = 1$. In the control, we set the initial bike distribution to the actual distribution from live data. In the ideal case, we let the model choose the optimal initial bike distribution.

Cost	Control Value	Ideal Value	Percent Improvement
Dissatisfaction	54.93 timestamps	54.00 timestamps	1.69%
Deviation	5925.13 ride-timestamps	4678.99 ride-timestamps	21.0%

Our model reduces the dissatisfaction cost by 1.69% and the deviation cost by 21.0%. This means that the initial bike distribution found by our model both leads to less user dissatisfaction and is more compatible with the expected flows of bikes throughout the day.



(a) Actual 3AM Distribution **(b) Ideal 3AM Distribution**

Figure 2: Comparison of ideal vs. actual station fill ratios at 3 AM.

To better illustrate where the system is over- or under-supplied relative to the model's ideal configuration, we plot the difference between the two distributions in Figure 3. Warm colors indicate stations that have more bikes than ideal, while cool colors indicate under-supplied stations.

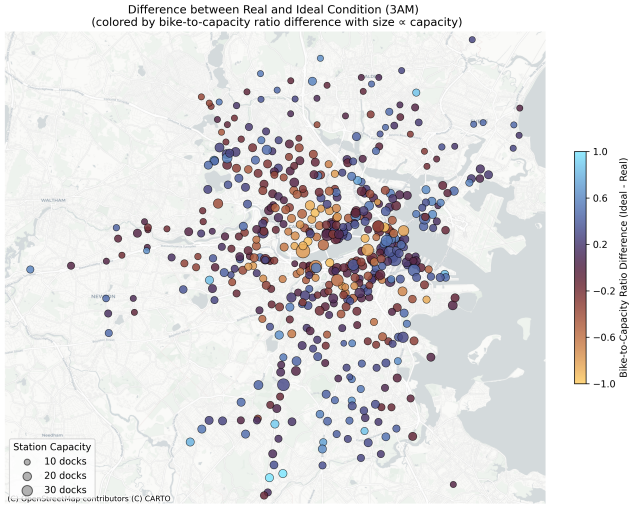


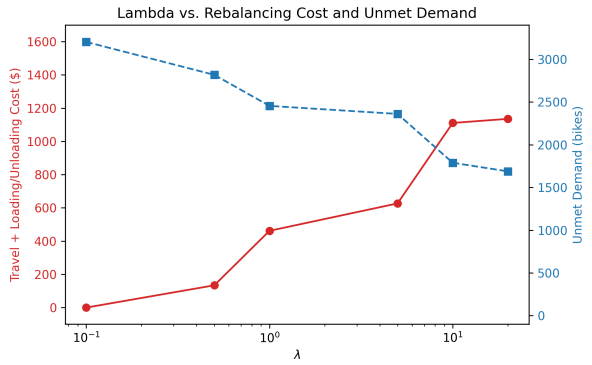
Figure 3: Difference between the ideal and actual station fill ratios at 3 AM.

4.2 Subproblem 2

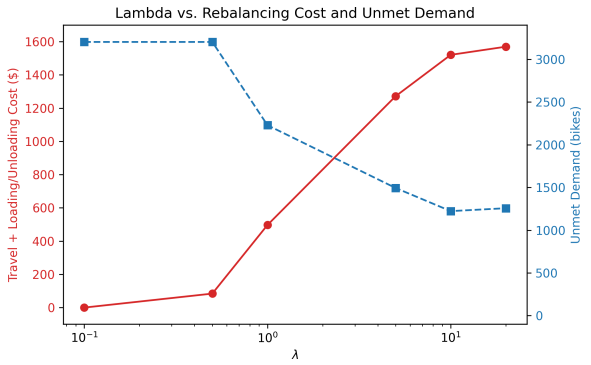
We estimate the fleet size to be five trucks, each with a 30-bike capacity as reported by Bluebikes. The distance cost is set to \$0.95 per km, and the loading/unloading cost to \$0.40 per bike, based on information gathered from news, reports and interviews.

We conducted two sets of experiments: (1) using 5 trucks with each station allowed to be visited at most by one truck, and (2 using 15 trucks with a visit limit of three per station. We evaluated both settings across multiple values of $\lambda = (0.1, 0.5, 1, 5, 10, 15)$.

We plotted the rebalancing cost and unmet demand over different choices of λ of the two scenarios.



(a) 5 Trucks + 1 Max Visit

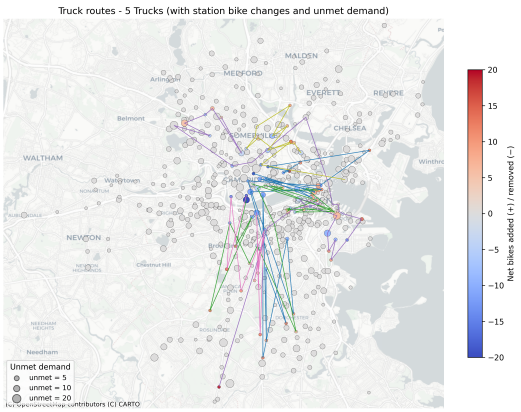


(b) 15 Trucks + 3 Max Visit

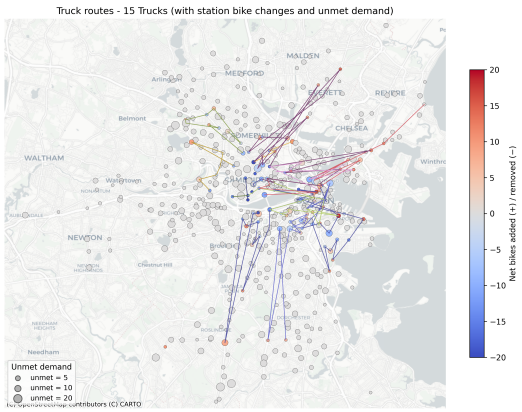
Figure 4: λ vs. Rebalancing Cost and Unmet Demand for Two Routing Scenarios

Increasing the number of trucks and allowing multiple visits per station drastically improves the system's ability to satisfy rebalancing demand, as reflected in the lower unmet demand in the 15-truck, 3-visit setting. While the 5-truck case seems to reach a feasibility limit: even when unmet demand is heavily penalized, the system cannot meaningfully reduce imbalance because of the lack of sufficient operational capacity. In contrast, the 15 truck case remains responsive to higher penalties, leading to additional reductions in unmet demand at the expense of higher rebalancing costs. These results suggest that if a single-time-step rebalancing strategy is desired, greater upfront investment in truck number and operational capacity is necessary; while such investment may reduce the cost for continuous rebalancing over the course of the day.

Figure 5 shows the routes for the two settings when $\lambda = 1$ (i.e. the unmet demands are not heavily penalized).



(a) 5 Trucks + 1 Max Visit ($\lambda = 1$)



(b) 15 Trucks + 3 Max Visit ($\lambda = 1$)

Figure 5: Truck Routing

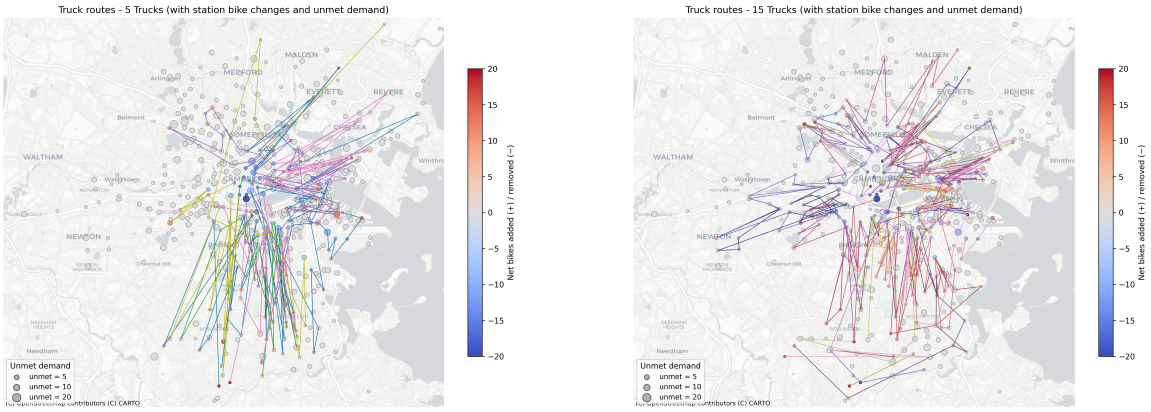
With only 5 trucks, each vehicle is forced to cover long, complex tours that span multiple peripheral clusters, reflecting the tight capacity constraints and limited routing flexibility. In contrast, when up to 15 trucks are made available, the workload becomes more spatially localized: the optimizer effectively deploys 11 trucks, each serving a smaller region with shorter routes. Because $\lambda = 1$ does not strongly penalize unmet demand, the model has little incentive to activate the remaining trucks, which explains why the not all trucks are not

utilized.

We also report the routes for the two settings when $\lambda = 10$ (i.e. the unmet demands are relatively heavily penalized) (Figure 6).

Compared to $\lambda = 1$, since the unmet demands are more heavily penalized, the routing for 5 trucks becomes overly complicated and sprawling. For 15 trucks, the coverge significantly expands and trucks travel farther into the periphery. Even so, only 13 of 15 trucks are actually used. Given the sparsity of the unmet demands over the periphery, it is still unjustified for the optimizer to deploy more trucks for extra routing.

Overall, the routes highlight a structural mismatch between the spatial pattern of demand and the routing configuration. Because the city center needs to be emptied while peripheral stations need to be filled, trucks are forced into long, complicated tours that do not look operationally intuitive. Under this spatial imbalance, the standard operational assumption we use (small trucks with limited carrying capacity) naturally produces long, fragmented, and sometimes unintuitive tours. This assumption is consistent with common industry practice and also shaped by transportation constraints (e.g., vehicle size limits, curb access rules, and downtown congestion regulations), but it fundamentally restricts how efficiently the system can respond to imbalances. The resulting routes suggest that simply adding more trucks does not fully resolve the issue. Instead, increasing capacity per truck may be better-suited: for example, allowing larger-capacity vehicles where policy permits, or better configure the truck for denser bike loading.



(a) 5 Trucks + 1 Max Visit ($\lambda = 10$) **(b) 15 Trucks + 3 Max Visit ($\lambda = 10$)**

Figure 6: Truck Routing

5 Impact

The successful implementation of this two-stage predictive rebalancing framework yields significant, quantifiable benefits across rider satisfaction, operational efficiency, and environmental sustainability.

5.1 Elevated Rider Satisfaction

One of the most tangible benefits of optimized rebalancing is a reduction in situations where riders encounter empty or full stations. These events are highly frustrating: empty stations block trip starts, while full stations prevent trip completions and force riders to search for alternative docking points. By modeling this in Subproblem 1's objective, we directly target these pain points.

This anticipatory balancing increases the likelihood that bikes are available where expected high-demand is

projected (e.g., residential areas in the morning) and that docks are open where high-inflow is anticipated (e.g., the city).

5.2 Operational Cost Reduction

Rebalancing is typically the most expensive non-fixed operational cost for bike-share systems, encompassing labor, fuel, and vehicle maintenance.

By generating a pre-optimized, multi-route schedule in Subproblem 2, we provide a good baseline plan for bike distribution. By front-loading much of the necessary redistribution, this approach can reduce the operational stress of continuous rebalancing throughout the day while keeping total costs minimized, providing a clearer baseline from which real-time adjustments can be made.

5.3 Enhanced Environmental Sustainability

Optimization of truck routes directly translates to a reduction in vehicle miles traveled by the rebalancing fleet. This results in a direct decrease in fuel consumption and associated greenhouse gas emissions.

Moreover, a more reliable bike-share system promotes greater mode-shift from private vehicles to Bluebikes, contributing to broader city-level goals for congestion reduction, air quality improvement, and urban sustainability.

6 Future Work

There are many directions we would like to extend this project.

6.1 Parameter Estimation

Despite producing feasible rebalancing routes, a key limitation of our current framework lies in the net flow estimation. The station-level net flows derived from historical averages often deviate noticeably from the flows that would actually make the system feasible within a single time step. As a result, the optimization sometimes compensates for inaccurate inputs rather than reflecting operational needs. Improving these estimates is likely a prerequisite for achieving more meaningful and reliable rebalancing plans.

Instead of relying solely on historical averages, we'd like future models to incorporate hourly weather forecasts (rain probability, temperature, snow), seasonal ridership patterns, and day-of-week differences in predicting our model parameters. Using time-series analysis or machine learning models (e.g., deep neural networks or Gradient Boosting) would allow for a more dynamic and accurate prediction of station-level netflows and station weights. This would significantly improve the robustness of the optimal initial bike allocation.

In addition, a structured sensitivity analysis across different parameter configurations (such as varying netflow forecasts, station capacities, or imbalance penalties) would help identify which configurations exert the greatest influence on system-wide performance. Understanding these "high-leverage" stations can guide operational prioritization and help diagnose when the planning model becomes infeasible under different scenarios.

6.2 Multi-Period Rebalancing

The current pre-operation (solved once per day) model is the first step. A more advanced system would incorporate real-time feedback.

We want to eventually implement a dynamic, multi-period optimization strategy. By dividing the day into several planning horizons (e.g., morning, midday, evening), our system could execute the morning plan,

then re-optimize the remaining daily plan based on the actual observed netflow and any deviations from the morning's schedule. This captures intra-day dynamics and allows for reactive adjustments within a predictive framework. This would both further reduce user dissatsifaction while also allowing for more efficient routing.

6.3 Incorporating Robustness

Finally, to ensure solutions are reliable under real-world operating conditions, we'd like our model to account for inherent uncertainty. For example, we could employ robust optimization to explicitly manage uncertainty in demand and netflow. This would generate an optimal initial allocation that performs well across a range of likely netflow scenarios, rather than just the average-case. This is particularly valuable for protecting against critical stock-outs/over-stocks during unexpected demand surges.