Vassili Chesterkine
Maïwenn Danno

Project Report
15.093 Optimization Methods

February 8, 2023

Context

Bluebikes is Boston's bike-sharing system, giving access to more than 4000 bikes across just under 450 stations in 11 municipalities of the Boston area. With close to 500,000 recorded trips in August 2022, the system is rapidly growing in popularity. While Bluebikes are, in theory, a very convenient way to get around the city, the reality is more complex. During peak hours, some stations are full, while other stations are empty, making Bluebikes very unreliable as users are not able to pick up a bike at their origin, or return it at their destination.

A viable solution is to use vans to move bikes from one station to another, freeing up docks at full stations and refilling empty stations. For example, relocating bikes from MIT to surrounding residential areas enables more students to bike to class. In fact, Bluebikes currently employs 4-5 rebalancing vans to redistribute bikes based on real-time data.

In this project, we aim to provide Bluebikes better rebalancing strategies, in order to maximize the possible number of rides while keeping rebalancing costs reasonable. Considering estimated demand, our mixed-integer optimization model determines efficient rebalancing strategies using vans to relocate bikes, eventually reducing unmet demand by 47%.

Data

We leverage individual-level trip data provided by the company, which contains an exhaustive list of all Bluebikes trips since 2015. Additionally, we use real-time system information provided as part of the General Bikeshare Feed Specification program (GBFS), including station position, capacity and inventory.

Station information

As mentioned, real-time data about the state of the Bluebikes system is provided, thanks to the GBFS program. This data includes static information such as a list of existing stations and their location and capacity, as well as dynamic information, namely the number of bikes and docks available in real-time at any station. From there, we were able to extract a list of stations and their corresponding capacities, as well as bike inventory at any point of the day, which provides us with an initialization of our model.

Demand estimation

Leveraging historical trip data from October 2022, we were able to estimate the number of trips from one station to another, at any time of the day. When the system is not saturated, i.e. there are docks and bikes available, user demand should be very close to actual user trips. However, when the system is saturated, user demand is not fully expressed. In that case, counting user trips only provides a lower bound for the demand.

In this study, we considered the following rule to estimate hourly demand between pairs of stations in case of saturation. Let us denote t_{ijt} the historical number of trips and consider a given station i of capacity C_i. The saturation of the station is modeled by the fact that either the incoming or outgoing flow of bikes exceeds the capacity of the station. For instance, in the latter case:

$$\sum_j t_{ijt} \geq C_i$$

We arbitrarily estimate the unobserved outgoing demand as:

$$(\sum_j t_{ijt} - C_i) \times 0.5$$

and evenly distribute this additional demand over all outgoing edges (i, j) to obtain our estimation of demand d_{ijt} between i and any other station j. A caveat of this approximation is that the historical data provided by Bluebikes already contains the current rebalancing operations, which further complexifies the task of inferring actual demand. In this study, we did however not take this issue into account since there was no immediate way to circumvent it.

Figure 1 shows the 30 stations with the highest estimated daily demand for incoming and outgoing trips, with the size of the circles proportional to the estimated demand. This reveals that several stations in the MIT/Harvard area are the most popular, which aligns with the information provided on the Bluebikes website.

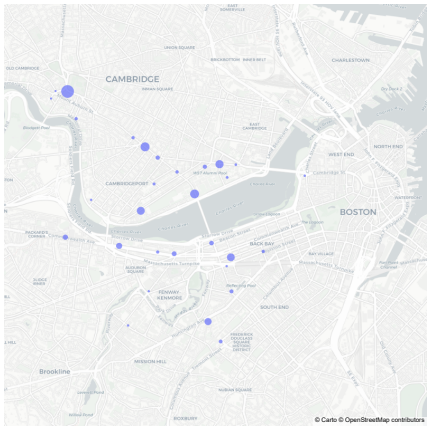

Figure 1: Estimated total demand for top 30 stations

Formulation

This section explains the formulation of our optimization problem. In order to reduce its computational complexity, we used a granularity of hours. Hence, all the decision variables and parameters are aggregated on an hourly basis.

First, the inputs of our model are the following:

- n: number of stations
- K, S: number and capacity of vans
- C_i: capacity of station i
- d_{ijt}: demand for travels from station i to j at hour t
- D_{ij}: distance between station i and station j
- X_{ij}: binary variable which indicates whether a van can travel from station i to j

We defined our decision variables so as to capture station inventory, user trips and van routing information:

- n_{it}: number of bikes available at station i at hour t
- w_{ijt}: number of user trips from station i to station j at hour t
- x_{ijkt}: equal to 1 if van k moves bikes from station i to j at hour t
- y_{ijkt}: equal to 1 if van k travels *empty* from station i to j at hour t (next rebalancing preparation)
- z_{ijkt}: number of bikes that van k moves from station i to station j at time t

The following sections detail our objective function and constraints of the optimization model.

Objective function

We aim at solving a bi-objective problem, using a trade-off parameter λ. On the one hand, we aim at minimizing unmet demand, that is the gap between estimated demand and actual users trips. On the other hand, we aim at minimizing the rebalancing costs. A simplified way to model these costs is to model the distance traveled by vans during rebalancing trips. The corresponding objective function is presented below:

$$\min_{x,y,z,w,n} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{t=1}^{24}(d_{ijt} - w_{ijt}) + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{t=1}^{24}\sum_{k=1}^{K} D_{ij} \times (x_{ijkt} + y_{ijkt}) \tag{OBJ}$$

Constraints on stations

Since n_{it} represents the inventory of the number of available bikes in a given station i at hour t, it must remain between 0 and the capacity C_i. To ensure this, we enforce $\boldsymbol{n} \in \mathbb{N}^{n \times 24}$ and add the following constraint:

$$\forall i, \forall t \quad n_{it} \leq C_i \tag{S1}$$

In addition, flow balance must always be maintained, taking into account the inventory of bikes and the number of bikes picked up or dropped off by users or vans at each station and time. The inventory n_{it} of the station i at hour t can be thus be derived from the inventory at hour $t-1$ with the following formula:

$$\forall i, \forall t > 0 \quad n_{it} - n_{it-1} = \sum_{j=1}^{n} w_{jit} - \sum_{j=1}^{n} w_{ijt} + \sum_{j=1}^{n}\sum_{k=1}^{K} z_{jikt} - \sum_{j=1}^{n}\sum_{k=1}^{K} z_{ijkt} \tag{S2}$$

with n_{i0} being initialized using real Bluebikes data.

Constraints on users trips

As in a capacitated network flow problem, we simulate the number of user trips between each pair of stations (i, j) at each hour t by using the corresponding estimated demand, such that user trips never exceed demand. We thus enforce the following constraint:

$$\forall i, \forall j, \ \forall t \quad w_{ijt} \leq d_{ijt} \tag{U}$$

Constraints on rebalancing trips

Adding the rebalancing dynamic is analogous to formulating a routing problem for all K vans. During each time step t, a van k is allowed two trips: a first trip to transport bikes from station i to station j, and a second trip from j to i' to prepare the next trip with bikes. The first trip is encoded by the binary variable x_{ijkt}, which equals 1 if van k is traveling from station i to station j at time t. z_{ijkt} corresponds to the number of bikes transported on this first trip and is limited by the capacity of the van S - as well as the inventory at station i, which is captured by constraint S2. The second trip is similarly described by a variable $y_{ji'kt}$, but this time the van is empty. In case it is not efficient to perform a rebalancing trip at time t, van k can stay in place at station i, indicated by $x_{iikt} = 1$ with zero cost. These dynamics are expressed by the following constraints:

$$\forall k, \forall t \quad \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijkt} = 1 \tag{V1}$$

$$\forall k, \forall t \quad \sum_{i=1}^{n}\sum_{j=1}^{n} y_{ijkt} = 1 \tag{V2}$$

$$\forall i, \forall k, \forall t > 1 \quad \sum_{j=1}^{n} x_{ijkt} \leq \sum_{l=1}^{n} y_{likt-1} \tag{V3}$$

$$\forall i, \forall k, \forall t \quad \sum_{l=1}^{n} y_{jlkt} \leq \sum_{i=1}^{n} x_{ijkt} \tag{V4}$$

$$\forall i, \forall j, \forall k, \forall t \quad z_{ijkt} \leq S x_{ijkt} \tag{V5}$$

Equations V1 and V2 ensure movement of the vans for each type of trip, equations V3 and V4 guarantee that vans leave from the station they arrived at previously, and equation V5 enforces the capacity constraint of vans.

Finally, we added feasibility constraints on rebalancing trips, so that vans can only use authorized edges (i, j):

$$\forall i, \forall j \quad x_{ijkt} \leq X_{ij} \tag{F1}$$

$$\forall i, \forall j \quad y_{ijkt} \leq X_{ij} \tag{F2}$$

Here, $\boldsymbol{X}$ could be chosen to allow all trips (i.e. $X_{ij} = 1, \ \forall i, j$), or to effectively constrain vans to stay in pre-defined neighborhoods.

Complete optimization problem

$$\min_{x,y,z,w,n} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{t=1}^{24}(d_{ijt} - w_{ijt}) + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{t=1}^{24}\sum_{k=1}^{K} D_{ij} \times (x_{ijkt} + y_{ijkt}) \tag{OBJ}$$

s.t.

$$n_{it} \leq C_i \qquad \forall i, \forall t \tag{S1}$$

$$n_{it} - n_{it-1} = \sum_{j=1}^{n} w_{jit} - \sum_{j=1}^{n} w_{ijt} + \sum_{j=1}^{n}\sum_{k=1}^{K} z_{jikt} - \sum_{j=1}^{n}\sum_{k=1}^{K} z_{ijkt} \qquad \forall i, \forall t \tag{S2}$$

$$n_{i0} = \tilde{n}_{i0} \qquad \forall i \tag{S3}$$

$$w_{ijkt} \leq d_{ijkt} \qquad \forall i, \forall j, \forall k, \forall t \tag{U}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijkt} = 1 \qquad \forall k, \forall t \tag{V1}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} y_{ijkt} = 1 \qquad \forall k, \forall t \tag{V2}$$

$$\sum_{j=1}^{n} x_{ijkt} \leq \sum_{l=1}^{n} y_{likt-1} \qquad \forall i, \forall k, \forall t > 1 \tag{V3}$$

$$\sum_{l=1}^{n} y_{jlkt} \leq \sum_{i=1}^{n} x_{ijkt} \qquad \forall i, \forall k, \forall t \tag{V4}$$

$$z_{ijkt} \leq S x_{ijkt} \qquad \forall (i, j), \forall k, \forall t \tag{V5}$$

$$x_{ijkt} \leq X_{ij} \qquad \forall i, \forall j \tag{F1}$$

$$y_{ijkt} \leq X_{ij} \qquad \forall i, \forall j \tag{F2}$$

$$\boldsymbol{n} \in \mathbb{N}^{n \times 24}, \boldsymbol{w} \in \mathbb{N}^{n \times K \times 24}$$

$$\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^{n \times n \times K \times 24}, \boldsymbol{z} \in \mathbb{N}^{n \times n \times K \times 24}$$

Results

Figure 2 plots a sample itinerary provided by a simulation of our optimization model on the 30 most active stations in Boston. Red lines indicate trips where vans carry bikes for rebalancing and blue lines indicate trips where vans are empty, preparing for a future rebalancing trip. In particular, the first two trips relocate bikes from Central Square to two MIT stations, most likely to anticipate upcoming high

demand after classes. These data-driven rebalancing strategies could be assigned to van drivers ahead of the day, instead of spontaneously deciding on rebalancing trips based on real-time data, which is often only marginally efficient.

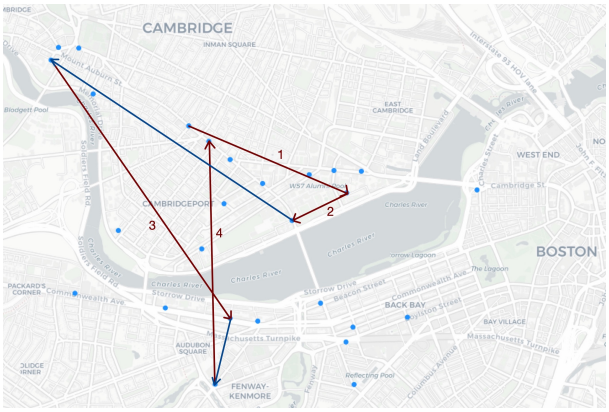

Figure 2: Van 1 itinerary sample (simulation on 30 top stations on October 10$^{\text{th}}$, using 5 vans)

Figure 3 presents a simulation of activity at the Ames St at Main St station in the MIT area, on October 10$^{\text{th}}$, 2022. The graphs compare estimated demand (*dark blue*) for outgoing and incoming trips with the actual number of trips, with (*light blue*) or without (*red*) the use of our proposed solution.

The left graph displays all outgoing trips, i.e. leaving the station. We see that our solution proposes to bring 20 bikes into the station at 4 PM, accounting for the upcoming peak in demand. With this simulation, our rebalancing operations allow a decrease in unmet demand, from 49% - without rebalancing - to 15%.

Similarly, the right graph represents all incoming trips, i.e. arriving into the station. Here, our model suggests removing bikes from the station in the morning in order to free up docks and consequently allow more users to drop off their bike. Concretely, this would allow more students to bike to class in the morning, since more docks would generally be available at that time of the day.

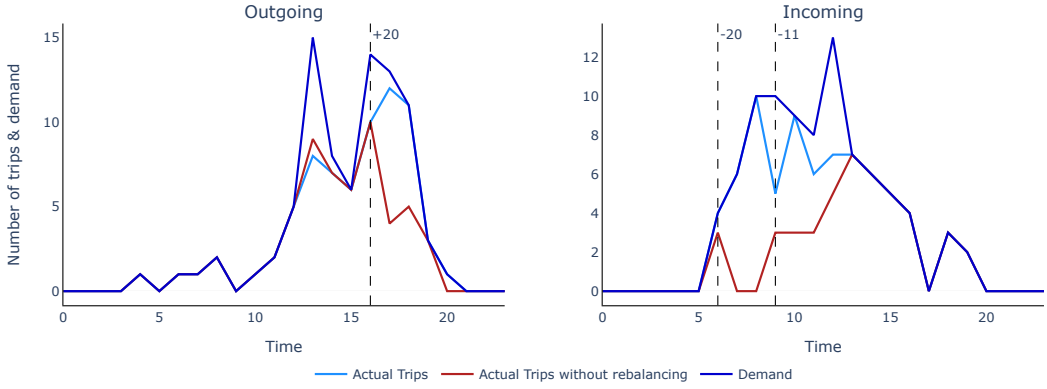

Figure 3: Incoming and outgoing bikes simulation at Ames St at Main St station

Overall, our solution enables a 47% reduction in unmet demand, and allows almost 500 additional trips daily, compared to an average of 15000 daily trips in October 2022. Figure 4 presents the results of our simulations for different values of the number of vans K. It is important to recall that the data used here already includes the current rebalancing trips performed by Bluebikes, and that consequently the results are skewed. Still, the underlying idea of this plot gives practical insights for the implementation of our solution.

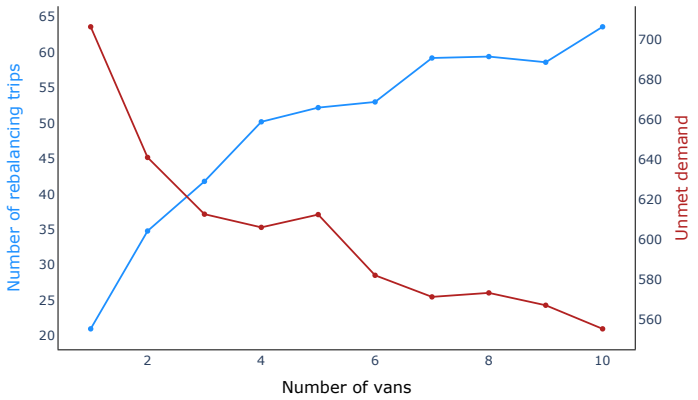

Figure 4: Demand as a function of the number of vans

Intuitively, increasing the number of vans progressively reduces the unmet demand by increasing the number of rebalancing trips. However, the incurred costs increase accordingly. Past a certain point, allowing additional rebalancing trips seems to have a marginal impact on unmet demand and might therefore not be relevant. This provides an understanding of the trade-off that would have to be made by Bluebikes when implementing our solution.

Next steps

One of the main challenges in this project is the estimation of the demand. Because we primarily focused on the formulation of the underlying optimization problem, we settled for a simplistic and arbitrary procedure to estimate demand. In fact, designing a more realistic estimation would be a project in and of itself, and might even require a proper market analysis. This would be a decisive step towards the improvement of our model.

Furthermore, demand estimation for all edges of the Bluebikes network, even if carried out thoroughly, would merely be an approximation and would in particular be subject to uncertainty in every simulation. For instance, roadwork or traffic jams in a neighborhood might incentivize even a small number of commuters to use Bluebikes instead of their car, which would in turn provoke perturbations that could propagate to the whole network, rendering our suggested strategy inefficient. In fact, using a suggested itinerary on subsequent days - unseen by our optimization model - which should in theory have similar demand, we observe that the reduction in unmet demand is closer to 25% as compared to the previously mentioned 47%. Concretely, this means that our simulation is not robust enough against fluctuations in demand.

Additionally, it would be interesting to scale our formulation, not only considering more stations for rebalancing trips, but also increasing the frequency of the latter. In this study, we used a time limit of 1000 seconds on our models, since the sizable number of integer variables likely made the branch-and-bound exploration very computationally-intensive. After the time limit, the optimality gap was typically around 10%. An interesting first step would be to run our optimization model with more computing power, for example on the MIT Engaging cluster. If this brings satisfactory results, including more stations would likely yield further decreases in unmet demand. In fact, Bluebikes stations in residential areas might not have the most traffic amongst all Bluebikes stations, but could be used to relocate bikes at rush hour, relieving high-traffic stations and enabling more bike commuting. Finally, authorizing more freedom in scheduling rebalancing trips, for instance allowing new trips as soon as the former trip is completed instead of waiting for the next hour, would likely further improve the performance of the model. In fact, most trips suggested by the current version of our model are less than 15 minutes. Even considering loading and unloading time, hourly granularity is a very conservative, yet simplifying, assumption, which wastes useful time that could be used to make Bluebikes a even more enjoyable experience.