**Research Summary**

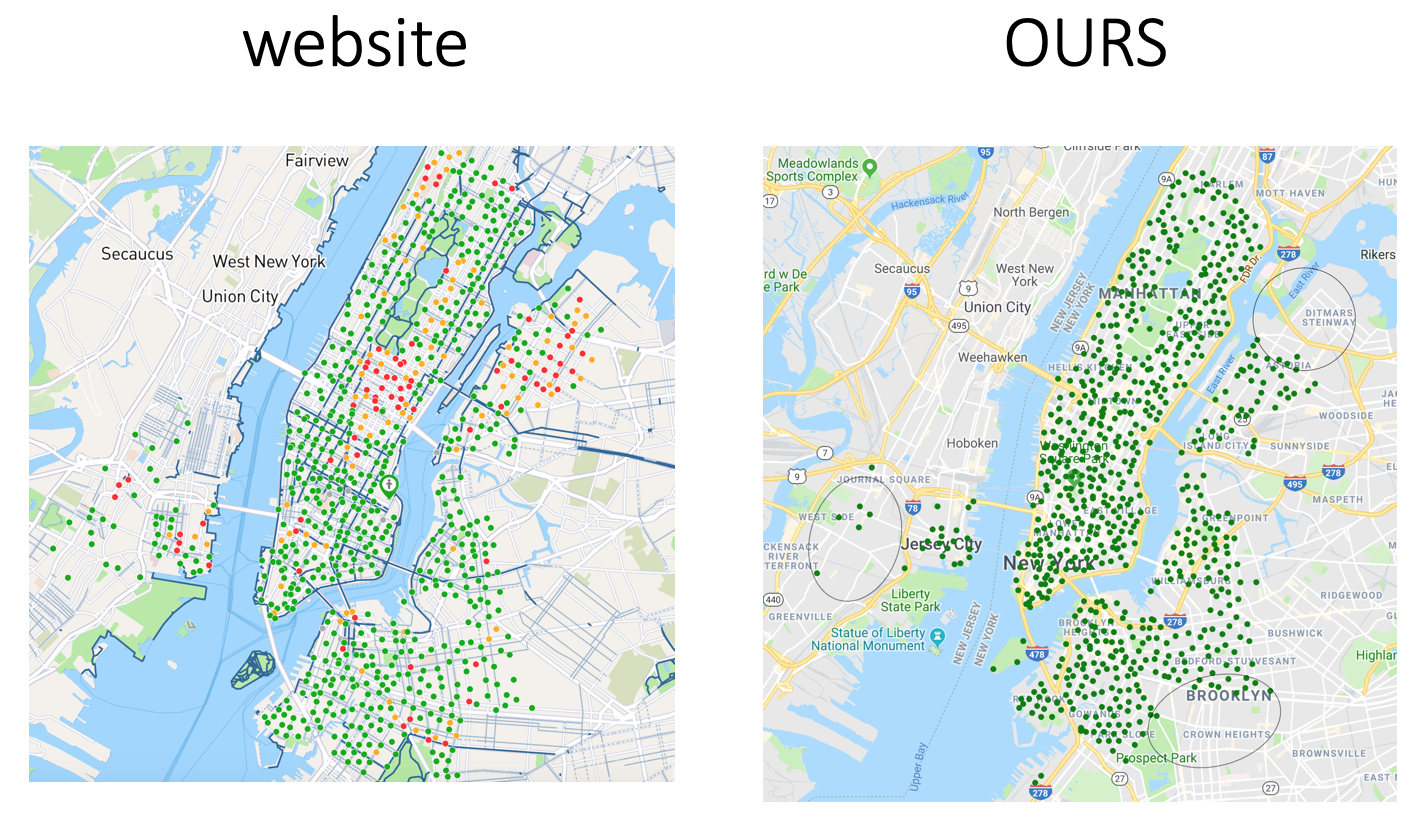
**Wenyuan Ma & Jiahe Wang**

Through this semester we focused on exploring different approaches to select electrified stations to minimize the out-of-battery events based on the following assumptions:

* In the future Citi Bike may replace all regular bikes with electrified bikes in NYC;
* Citi Bike may choose a certain percentage of stations to be charging stations to optimize the system performance;
* E-bike speed in NYC is 10mph;
* Depletion rate is 40% per hour and it takes around 3 hours to fully charge a depleted e-bike;
* If there is no e-bike available in the origin station or the destination station is full, then the trip is canceled (demand lost); Otherwise, an e-bike with highest state of charge (SOC) gets picked up;
* A depleted e-bike can still work as a regular bike with a lower speed (around 4.4 mph);

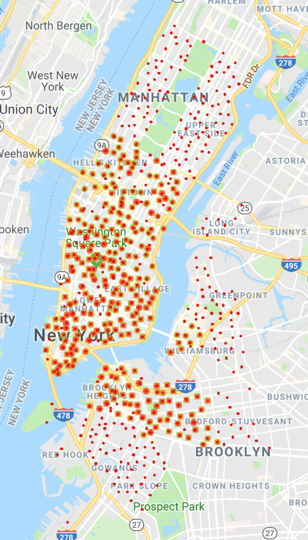
The simulation set the common random seed = 1)

Stations considered in our research (plot on the right):



(We collected stations from June 2017 till September 2017. The stations missing (circled) might be built after 09/2017.)

The usage heatmap for the stations we were considering is: (note: usage = trips in + trips out + trips failed in + trips failed out, we got the measurements from a 20-week simulation in steady state assuming all stations are electrified.)

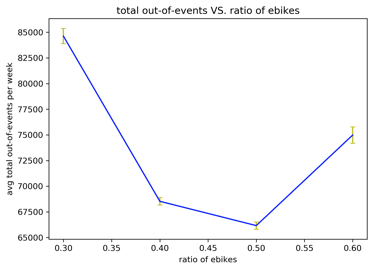


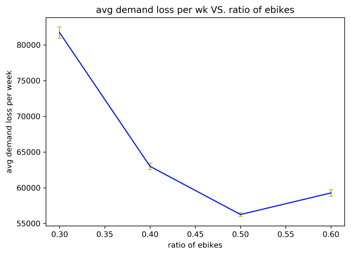
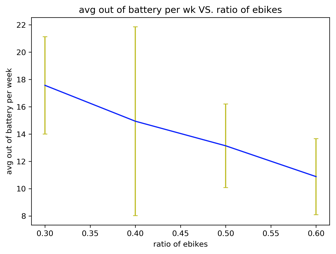
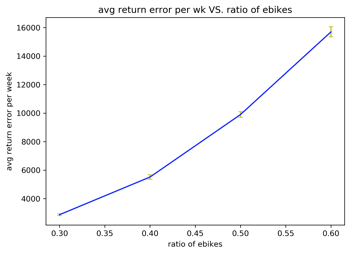
Definitions:

* Out-of-events = demand lost + out-of-battery events + failed return events;
* Demand lost: No e-bikes got picked up (due to empty origin station or full destination station at the start of a trip);
* Failed return: The destination station is full by the time an e-bike arrives;
* Out-of-battery: The SOC = 0 at the end of a trip;
* Trips in/out (for any station): Total number of trips into/out of the station;
* Trips failed in (for any station): Failed return for the station;
* Trips failed out (for any station): Demand loss for the station;

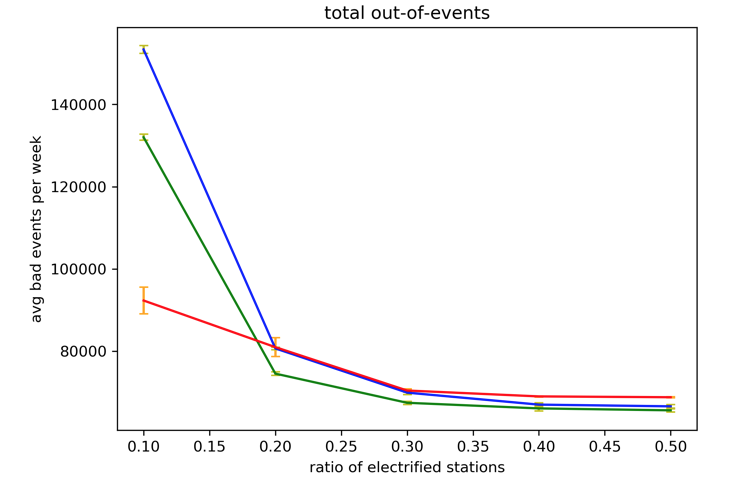
We have the following findings:

* The optimal ratio of e-bikes is 0.5. At 50% level, the total out-of-events is minimized.





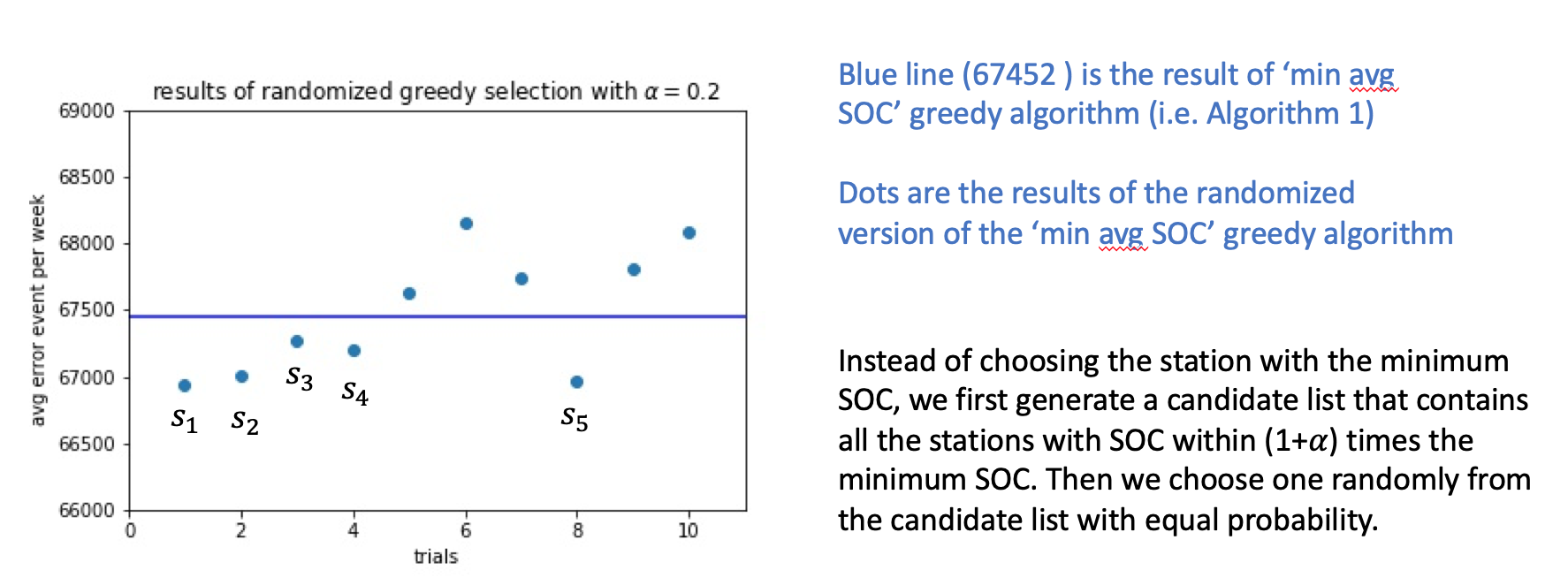
* We tried 2 greedy algorithms plus benchmark:
  + Algorithm1: Select a station with the least average SOC greedily.
  + Algorithm2: Pick the destination station with the max number of out-of-battery events as an e-station and repeat greedily.
  + Algorithm3: Benchmark Algorithm: Choose 10 stations with largest usage as candidate stations. Then, among these 10 stations, pick the one resulting least out-of-events.

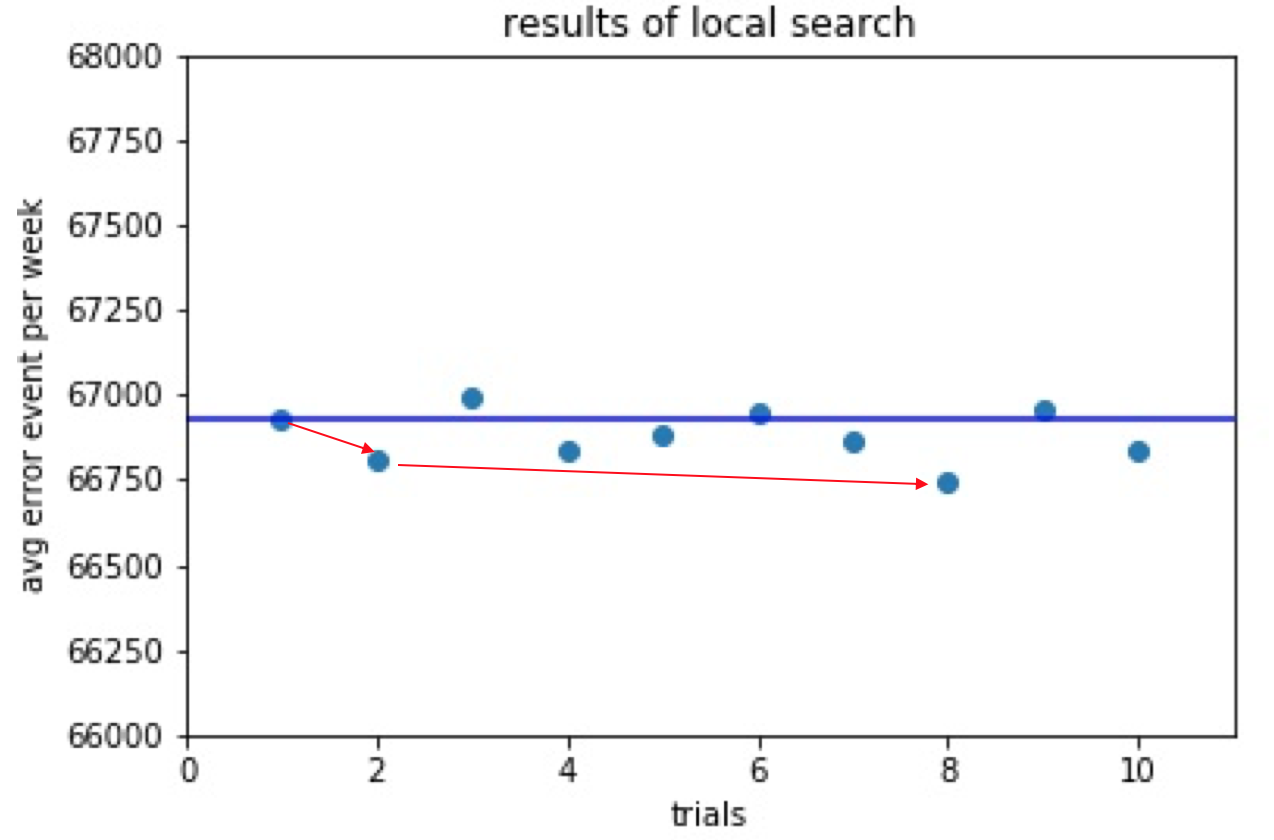


(green: Algorithm 1; blue: Algorithm 2; red: Algorithm 3)

Conclusion: 30% of charging stations is a good ratio.

* Since algorithm 1 gave the best result, we then explored a randomized greedy algorithm associated with least average SOC and a local search algorithm:



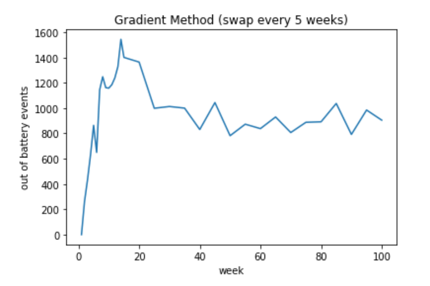


We have tried randomized greedy selection for 10 times and evaluated their performance. We identified 5 results which have fewer error events than previous methods (S1, S2, S3, S4. S5). S1 is the best among the five. Our local search starts with S1 and in each iteration, it swaps a randomly chosen station in current solution with another randomly chosen station outside the solution. We further restrict that the station to be swapped in should belong to the union of S1, S2, S3, S4. S5 (we consider stations in those five sets as “good” candidate stations). After the swapping, we evaluate the new solution. If it has better performance, we do a “move” in the local search, which means in the next iteration, we will perform swapping on the new solution. In the graph above, we did 10 trials and made 2 “moves”.

* Starting from the best 30% selection result of local research, we took *the amount of charge a station can potentially provide given it is electrified* as gradient to modify the e-station subset.
  + Procedure for the gradient approach:

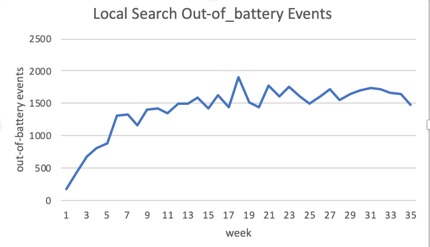
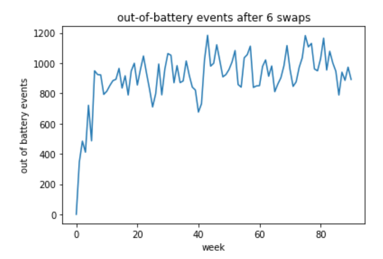
1. Simulate the system for 15 weeks (warm-up period) based on the 30% subset from local search;
2. Simulate another 5 weeks and rank electrified stations and non-electrified stations respectively based on their gradients from the performance in the 5th week.
3. Make the 1st “swap decision”, i.e. the electrified station with the smallest gradient will be swapped to be non-electrified and the non-electrified station with the highest gradient will be swapped to be electrified station.
4. Repeat the 5-week-1-decision process for several iterations.

* Illustration of the procedure:



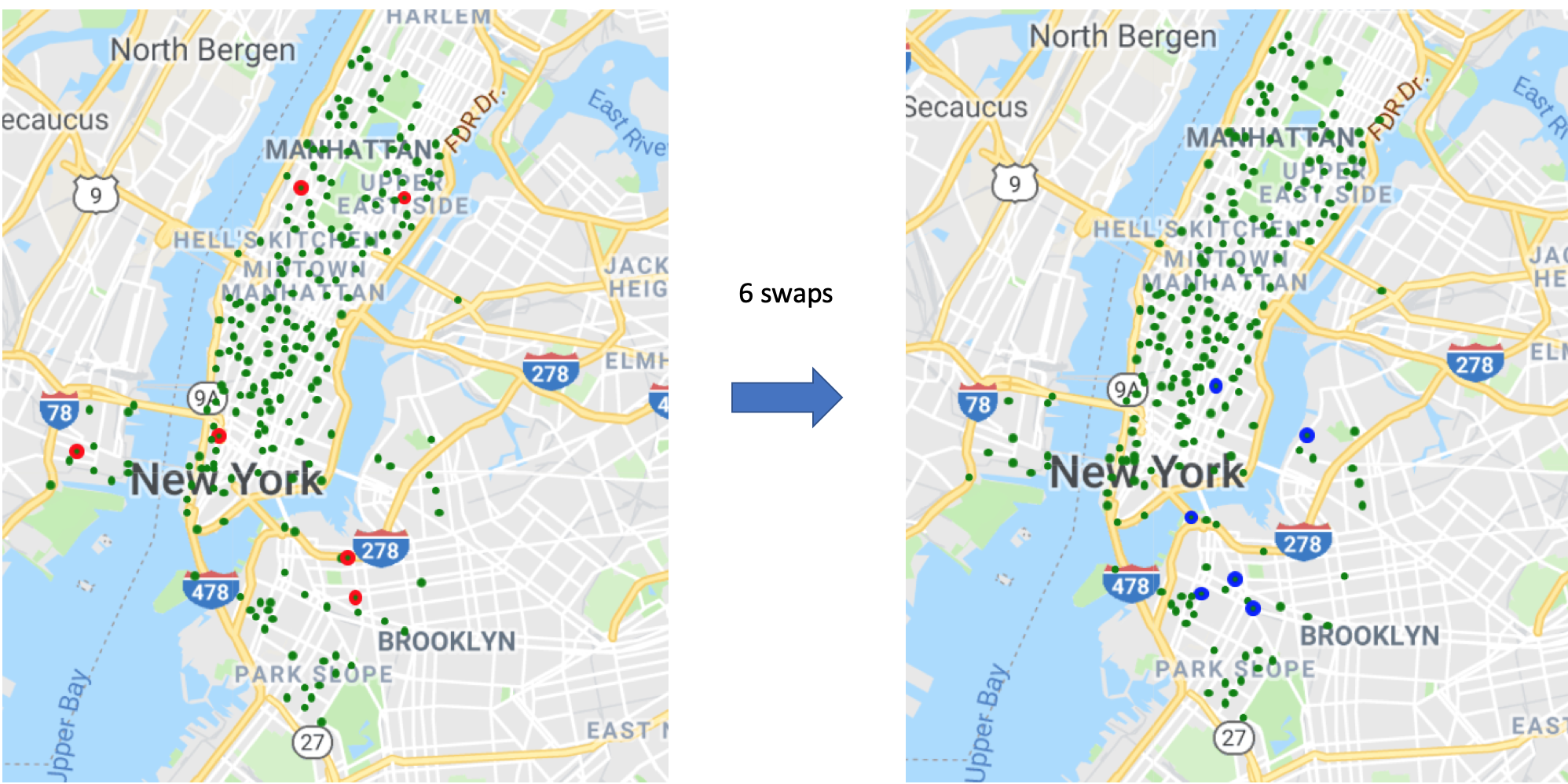
(We found the gradient method can improve the local search result significantly and at the 50th week, 782 out-of-battery events (result of 6 swaps))

Performance of the original subset from local search VS. Modified subset (after 6 swaps)

Conclusion: With the subset after 6 swaps, there are around 900-1000 out-of-battery events per week on average now. Improved from the local search subset performance!

Visualization of the 6 swaps:



Future TODOs:

* To see whether some stations’ size (capacity) should increase to address failed return problem. Especially for stations near subway/ bus stations, (considering our research on multimodal transportation), the return error may be pretty high.
* Try more algorithms to reduce out-of-battery events.
* To address demand loss problem, may examine some stations with large demand loss. Maybe we should increase their capacities or send staff to move some e-bikes to there, leading to a logistic problem to solve/optimize.
* Use different gradient measurements for electrified and regular stations. For e-stations, use the current gradient measurement, which is the total amount of electricity it provides. For regular stations, use the number of out-of-battery events that originate from that station.
* Reduce the percentage of e-stations from 30% to 20%
* Model human behavior, like incentive to let people return bikes to charging stations