Rebalancing citi bike Stations

CONSTRUCTING A GRASP ALGORITHM TO OPTIMALLY REALLOCATE BIKES

Ellen Chen, Shangdi Yu, Daniel Freund, David B. Shmoys

INTRODUCTION

- Maintaining a successful bike-sharing system involves constant rebalancing tasks throughout the day.
- Customers move bikes around system
- Asymmetric demands
- Truck drivers must reallocate bikes so there are available bikes and opens docks at every station.
- We improved the effectiveness of a truck's rebalancing tasks by using a GRASP algorithm to search for near-optimal solutions for bike reallocation.

GRASP ALGORITHMS

- A **GRASP** (greedy randomized adaptive search procedure) is a metaheuristic algorithm for optimization problems.
- Algorithm Steps:
  - Run greedy randomized algorithm to construct base solution
  - Use local search to improve each solution.
  - Run many iterations and choose the best solution found

USER DISSATISFACTION FUNCTIONS (UDF)

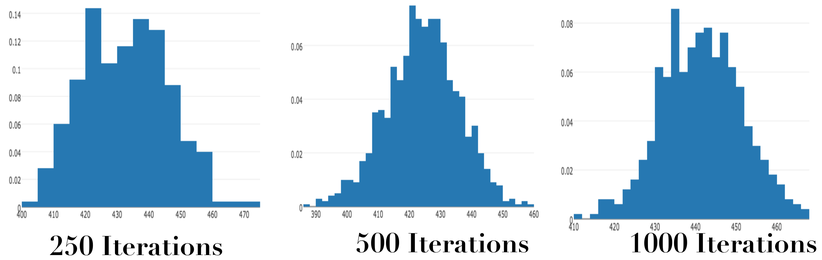
- Our greedy heuristic was constructed around UDFs.
- There is a unique UDF for every bike station at every half hour interval.
- A UDF for station X at time T maps each value in [0, n], where n is the total number of docks at X, to a value C.
- C represents the expected number of customers unable to pick up/drop off a bike in the next hours given that there are currently X bikes at the station.

SAMPLE RESULTS

- Our GRASP demonstrated improvements in comparison to a simple greedy algorithm.

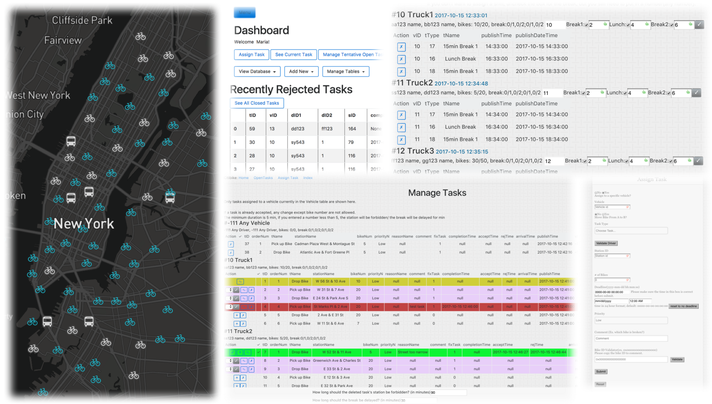
	Greedy	GRASP
Case 1:	360.21	385.08
Case 2:	259.53	272.80
Case 3:	350.67	374.63

- These histogram show how the algorithm improves over increased iterations.

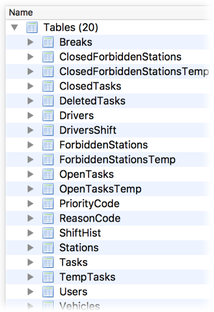


250 Iterations | 500 Iterations | 1000 Iterations

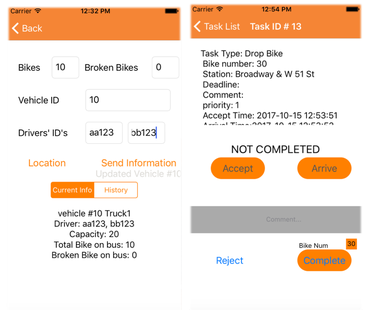
SYSTEM IMPLEMENTATION



Dispatcher Side



Database



Rebalancer Side

ALGORITHM APPROACH

- **Greedy Heuristic:**
  - Using the UDFs, we found C, the optimal number of bikes that should be at every station at the current time.
  - Calculate the time, T, it would take to drive to each station, and move the appropriate number of bikes from or to the station.
  - The optimality of each task was ranked by the value of $\frac{C}{T}$
- **Randomization Method:**
  - Retrieve the best 5 moves defined by the greedy heuristic
  - Weigh each task as 2^n, where n is equal to $\frac{C}{T}$
  - Use a random weighted selection to chose from the top-5 tasks.
- Our implementation ran ~10 iterations per minute on a system with >660 stations and 4 trucks.

INTERFACE DESIGN

We created a web and mobile application to implement and visualize the algorithm and assign tasks.

- **Web app features for dispatchers**
  - Customize algorithm inputs
  - Modify the existing schedule and the schedule generated by the algorithm
  - Create tasks for the algorithm to assign.
  - Set stations as forbidden.
  - Schedule breaks for drivers
  - Track/manage vehicles/ stations/ tasks
- **Mobile app features for rebalancers**
  - Update vehicle and driver information
  - Accept/ Reject/ Report complete tasks
  - Begin/ End shifts

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under grant CCF-1522054 and would not have been possible without the Citi Bike research group at Cornell University and the many people at Motivate International.