

Ridesharing

CS581 Database Management Systems



Jaspreet Kaur Sohal
Rashmi Arora
Sanjana Channabasappa Mallik
Yashika Goyal

Agenda

- Objective
- Dataset
- Constraints
- Demo Results
- Algorithm
- Weighted Max Matching
- Output Plots

Objective

Devise a ride-sharing algorithm to merge individual trips given constraints to determine :

- The number of miles saved
- Number of trips saved by the algorithm

Dataset

Data Source - <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

Dataset considered:

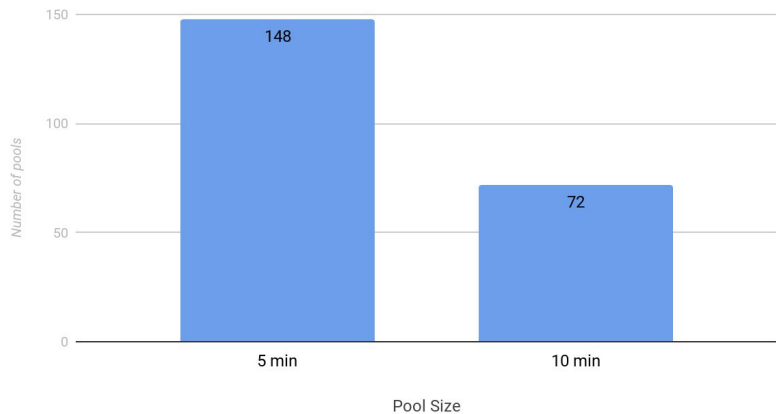
- Yellow Cab Data - January 2016 - May 2016
- Total number of trips in dataset is ~ 300,000

Constraints

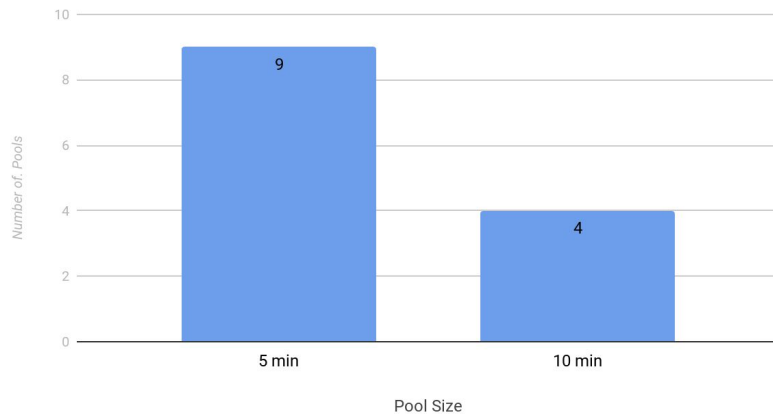
- Merging only static trips
- Maximum of 2 trips merged
- The taxi maximum occupancy is taken as 3
- Source Hub is taken as JFK
- Social Score based on parameters such as Rider's Profession and Language is assigned
- Pooling window of 5 min and 10 min is considered
- Delay time proportional to the actual trip time is considered
- Infinite number of taxis available

Demo Results

Test Data A - Number of Pools processed in 14 hours

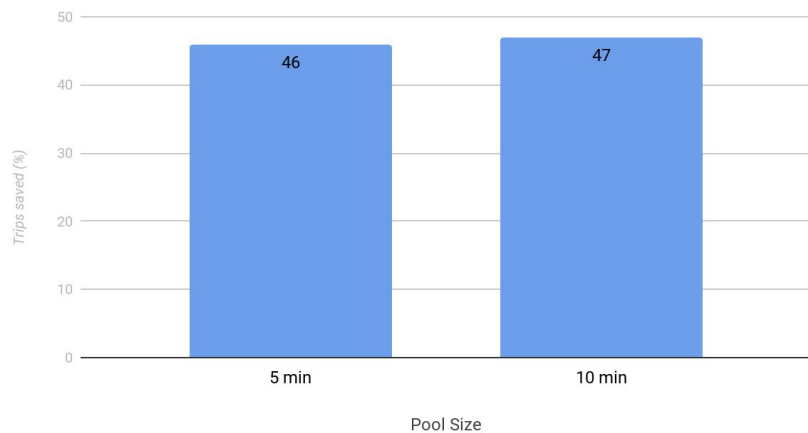


Test Data B - Number of Pools processed in 5 min

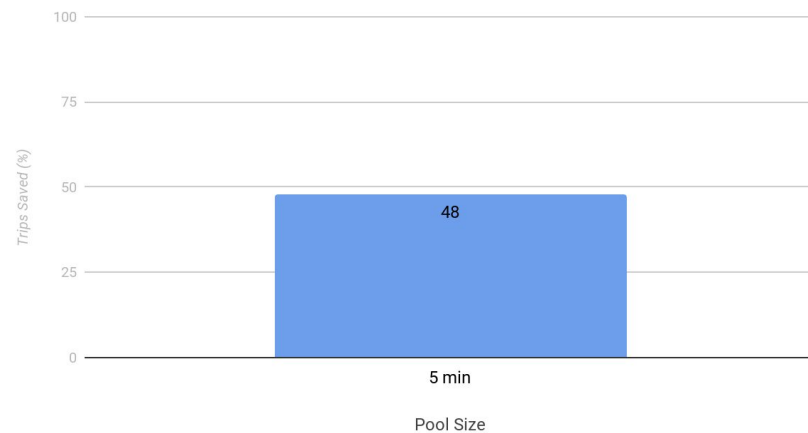


Demo Results

Test Data A - % Trips Saved



Test Data B - % Trips Saved in 5 min



Algorithm

- Requests within a chosen pool window (5, 10 min) are put into one request pool
- Shareability of the ride is decided based on Lower Bounding using Haversine distance

$$SP(A) + T_{dest(A), dest(B)} < SP(B) + Delay(B)$$

- GraphHopper API is used to compute the optimal route and distance between the source and destinations
- Weighted Graph is formed on the basis of distance saved and social score similarity measure.
- Weighted Max Matching is used to generate combined trips

Weighted Max Matching

Algorithm WEIGHTED_MAX_MATCHING

Input: the shareability network $S = (T, L)$

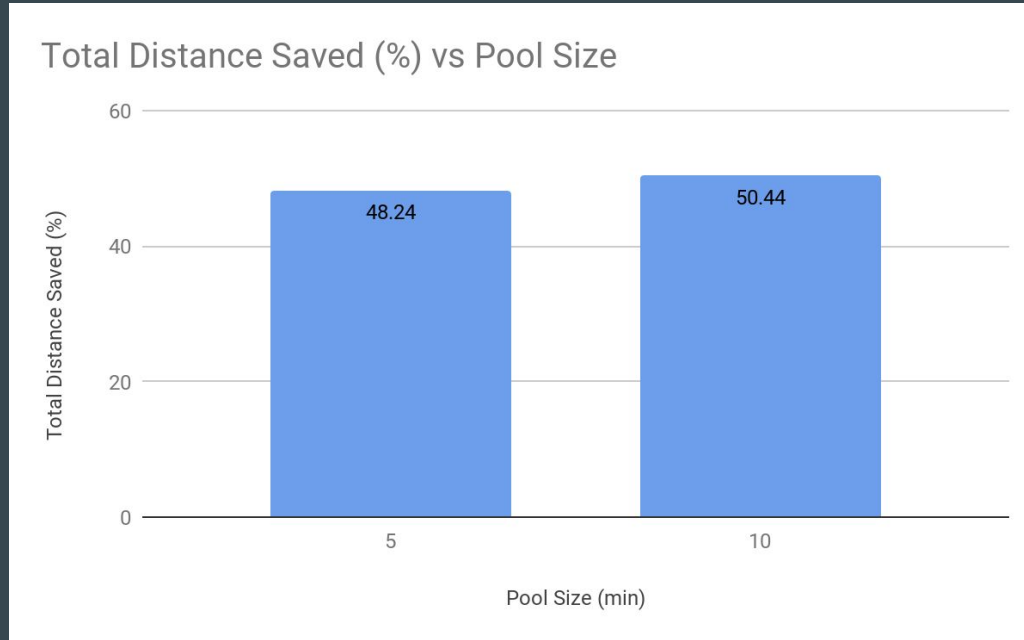
T = set of all trips

L = set of links $(T_i, T_j, \text{weight}_{ij})$

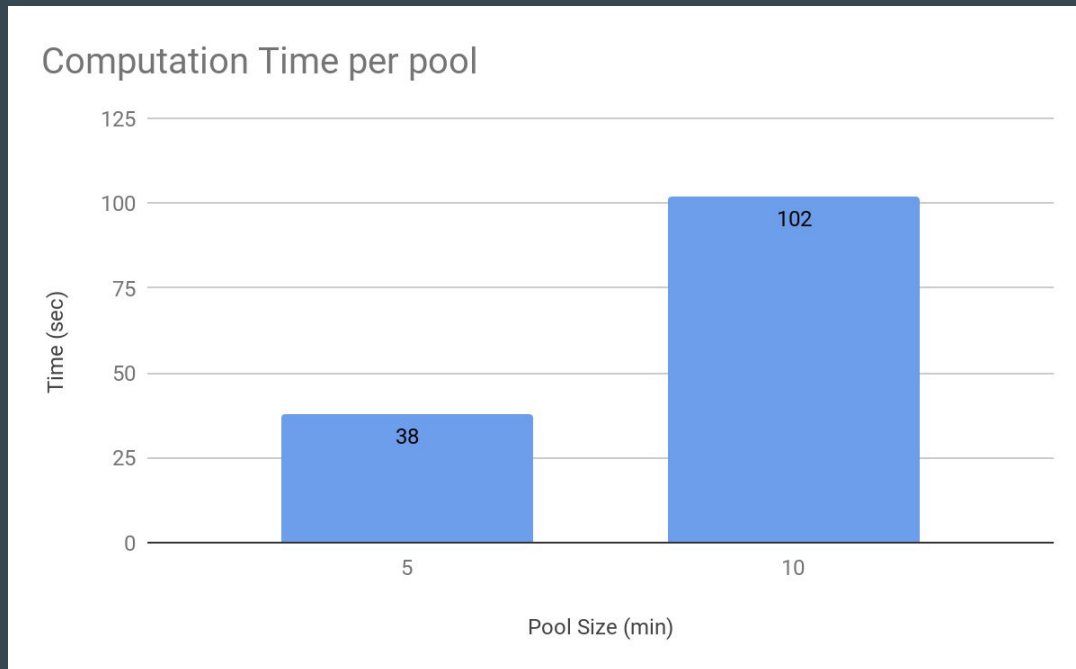
Output: the set R of (possibly combined) trips to be performed

1. $R = T$
2. Build a weighted maximum matching M_{\max} on S - **NetworkX**
3. for each $(T_i, T_j) \in M_{\max}$ do
 - $R = R \cup \{T_{ij}\}$
 - $R = R - \{T_i, T_j\}$
4. return R

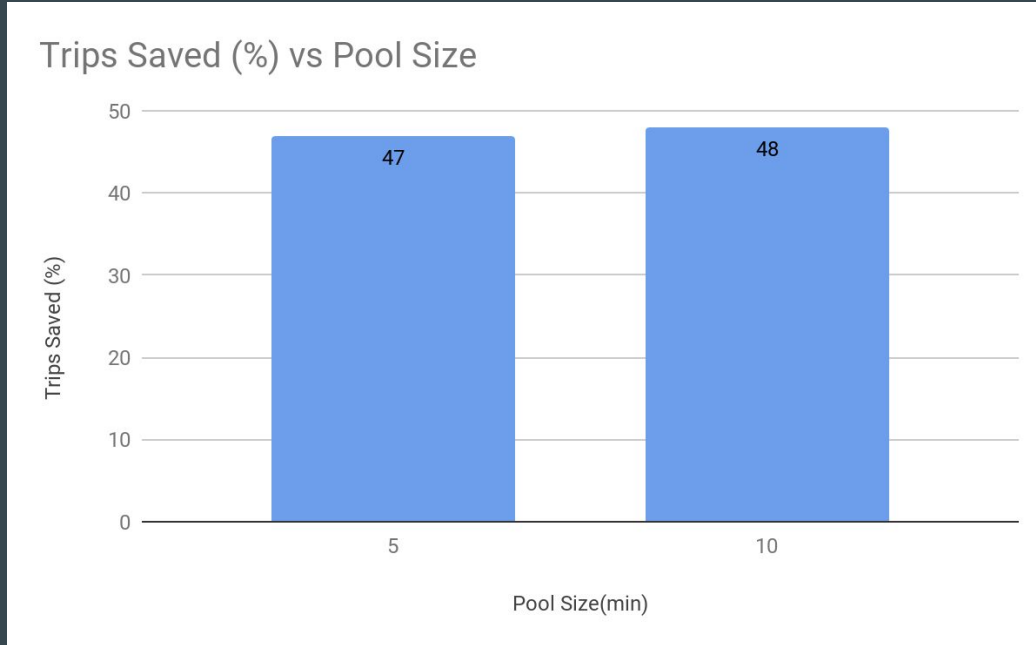
Output Plots



Output Plots

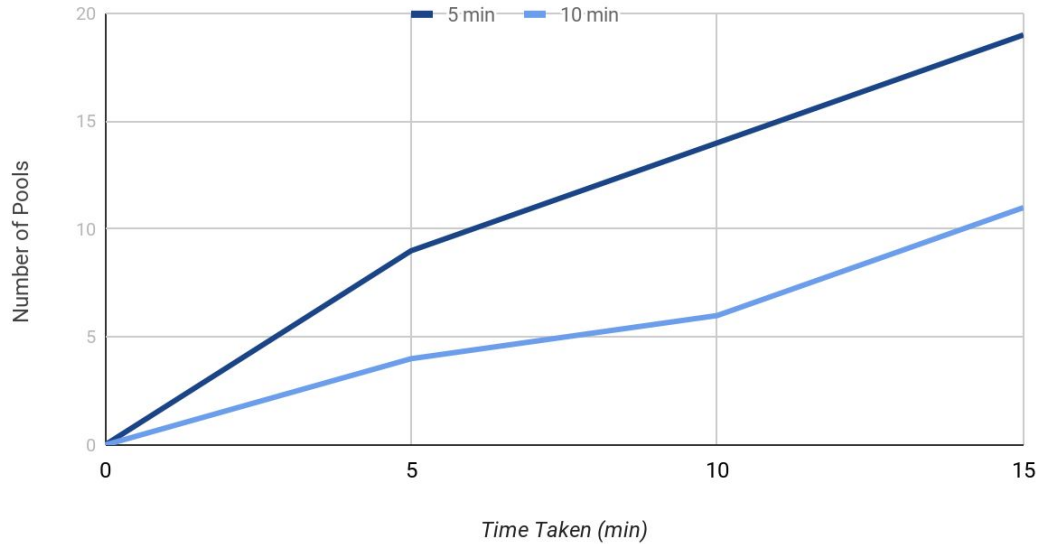


Output Plots



Output Plots

Number of Pools Processed



Questions?