AUTONOMOUS VEHICLES IMPACT ON RIDE-HAILING

Sai Krishnan Thiruvarpu Neelakantan sthiru5@uic.edu

Praveen Chandrasekaran pchand34@uic.edu

Aditya Ramakrish aramak24@uic.edu

Vidhyasagar Udayakumar vudaya2@uic.edu

GUIDANCE OF

PROF. OURI WOLFSON, PHD

TA: AISHWARYA VIJAYAN

COMPUTER SCIENCE COLLEGE OF ENGINEERING

UIC

Abstract

Company ABC approached us to do a performance analysis on their fleet of cars used for ride hailing. They would like to know what would be a more profitable approach for the company as well as best for the customers by doing a comparison of a crowdsourced fleet of vehicles vs an autonomous one. We need to factor in a few parameters that would help us make a decision on the number of cabs to operate, the algorithm used to assign cabs to resources and so on.

A few parameters that consider are agent search time, resource wait time and profitability for the company. We would look for the approach that would reduce the expiration percentage, agent search time and resource wait time, thus maximising the profit for the company.

Contents

1. Introduction

Ride hailing became one of the most used commercials after the introduction of Global Positioning System (GPS). To date, one of the large scale bodies for ride hailing services are Uber and Lyft which provide a direct link between the drivers and passengers. Since the introduction of ride hailing services and carpooling services, its impact on reducing the number of vehicles, reducing traffic congestion, and reducing pollution has been phenomenal.

In 2017, the global ride-hailing service was valued at $36,000 million and it is projected to reach around $126,000 million by 2025. With a CAGR of 16.5%, the market is to see a tremendous growth, proving it to be an industry worth the investment of time and research.

The rapid growth of this service can be attributed to the rise in the trend of on-demand transportation services of millennials along with the creation of employment opportunities and reduction in car ownership.

The history of ride hailing isn't exactly too far back, but it is essentially an iteration of the standard taxi services. Currently, North America is the highest contributor to the market of ride hailing services followed by Asia Pacific. There is however speculation that there will be a lot of changes in this trend due to expansions in the field. All factors considered, it is safe to say that car manufacturers are definitely looking to secure their businesses through partnerships and new ventures in ride hailing services.

There are still certain expectations that this market faces like the extension of autonomous vehicles available in the ride hailing industry.

Here, in this project we are consuming the yellow taxi records in order to build a model that gives out results efficiently with respect to the passenger and the driver.

The system is built with two algorithms to make the passengers and drivers get their intended mode of benefits. These algorithms are developed, keeping in mind that both the passengers and the drivers are benefitted, and not just the company.

1.1 Data

The first step was data collection. To test the demo, data was collected for May 17th, 2016 and May 20th, 2016. According to the data for May 17th, 2016, the total number of requests was 326,612. The maximum waiting time of the resource i.e. the customer was 600 seconds. This was the duration that the customer had to wait before the vehicle assigned to her reached her. The assignment periods were of two kinds: 30 seconds and 60 seconds.

1.2 Pre processing

- Initial analysis was conducted by filtering out requests between 8 am and 10 pm.
- We have seen an average of 330000 requests per day and 316 requests per minute.
- Choose a benefit factor that would consider the destination to drop off a customer also to assign cabs to customers. Benefit calculated as the ratio between time taken to pick up resources by the trip cost.
- Made amendments on the COMSET solution to accept a pool of resources and then do the matching of agents and resources.

2. Assumptions and Constraints

For the algorithm under consideration, there were some assumptions and constraints set. They were:

- Considered trips starting and ending in Manhattan.
- Resources are always stationary.
- Throughout the simulation, the number of agents will be fixed.
- Speed at each road segment = speed_limit * ((map_average_trip_duration)/(TLC_average_trip_duration))
- Random intersection for empty cruise in search of resources.
- Until expiration, the unassigned resources are taken to the next pool.

2.1 Datasets evaluated :

1) Experiments on: Jan 2015 - Dec 2015
2) Demo on: May 17, 2016 & May 20, 2016
3) Initial analysis on: June 1, 2016

2.2 Configurable Parameters:

- Number of Agents, Default value set: 5000
- Resource Expiration Time, Default value set: 10 Mins
- Assignment Period, Default value set: 30 Sec
- Assignment Algorithm, Default value set: Fair (based on shortest pick up time)

3. Algorithms Evaluated

3.1 Fair Assignment:

Matching the resources in a given pool with the closest available agent based on time is assigned as fair.

The input is the list of available vehicles and the requests received in each pool. The edges connecting the two lists would be the time taken for each agent to reach the resource.

The output then returns the matching based on the shortest time taken for a resource to reach an agent.

3.2 Optimal Assignment:

For optimal assignment, the Hungarian algorithm was used to perform a matching that would best optimize a benefit factor which is calculated as the ratio of the pickup time to total trip fare i.e. lower the benefit factor , the more profitable for an agent to pick up the resource.

The input was the list of available vehicles and the requests received in each pool. The edges connecting the two lists would be the benefit measure between that agent and the resource. The output finds the combination with maximum total benefit for the pool with bipartite matching and returns the result.

3.3 Performance Improvement:

The distance between the intersections can be pre computed. This can then be stored in a hash map, so that the performance can be increased for scalability.

4. Experiments Conducted

Experiments Conducted	Parameter Setting
2015 Year Trend by Month	1. Number of Agents: 5000 2. Resource Expiration Time: 10 Mins 3. Assignment Period: 30 Sec
Seasonal Comparison	1. Number of Agents: 5000 2. Resource Expiration Time: 10 Mins 3. Assignment Period: 30 Sec
Day/Night Comparison	1. Number of Agents: 5000 2. Resource Expiration Time: 10 Mins 3. Assignment Period: 30 Sec
Trend by Changing the Number of Agents	1. Number of Agents: 3000, 5000, 7000 2. Resource Expiration Time: 10 Mins 3. Assignment Period: 30 Sec

Link for all the simulation results: Click **here**

4.1 Performance Metrics

Plots were generated for below performance metrics

The Wait Time of a Resource: The period of time from its introduction until its pick-up or expiration.

The Expiration Percentage: The percentage of expired resources. Each resource has a maximum lifetime starting from its introduction, beyond which the resource will be automatically removed from the system.

The Search Time of an Agent: The amount of time from when the agent is labeled empty until it picks up a resource. An agent experiences multiple search times, each corresponding to one assignment.

4.2 Tech Stack Used

Language: Java, Python

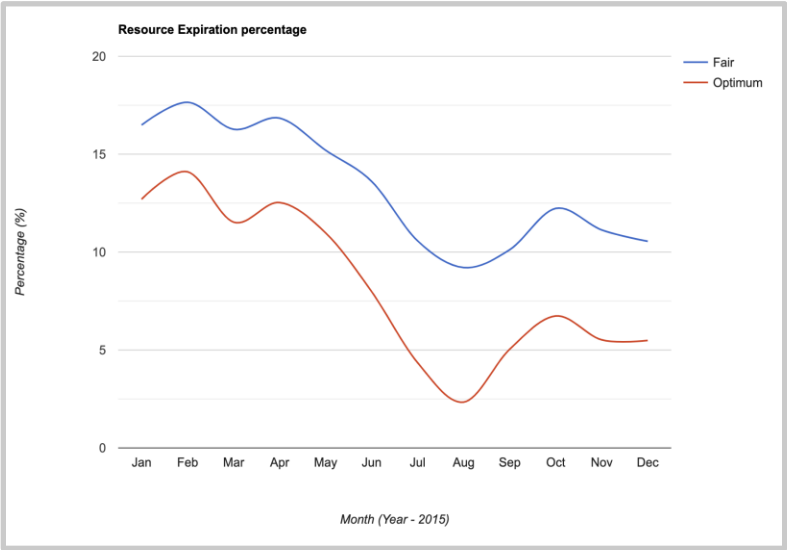
Software: Intellij, Jupyter Notebook

3rd party packages: Hungarian Algorithm (link)

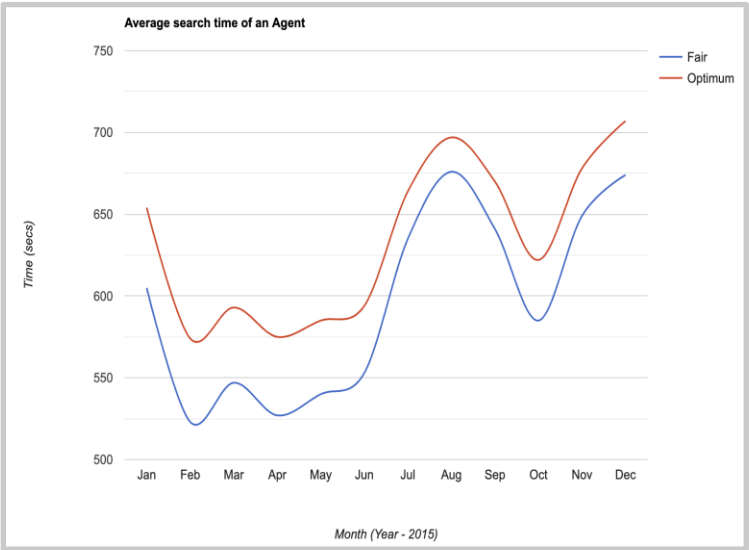
Framework: COMSET

5. Results

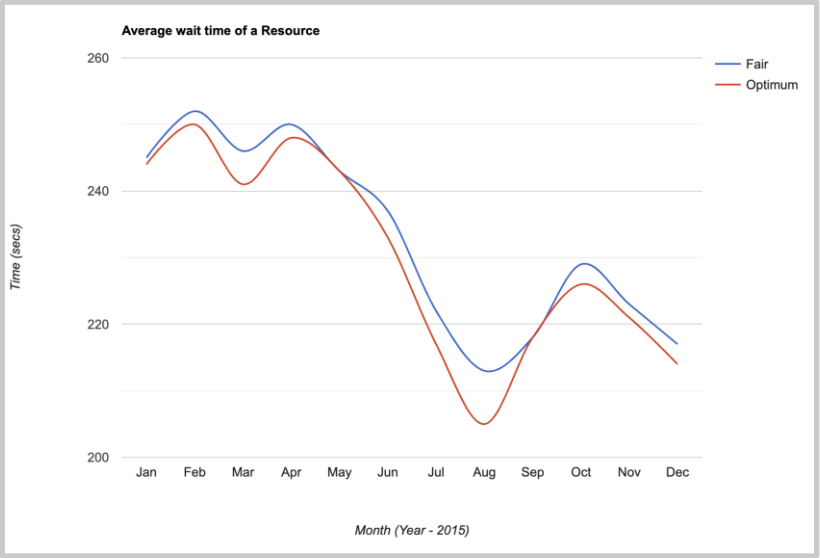
5.1 2015 Year trend by month and seasonal comparison



RESOURCE EXPIRATION PERCENTAGE



AVERAGE SEARCH TIME OF AGENT



AVERAGE WAIT TIME OF A RESOURCE

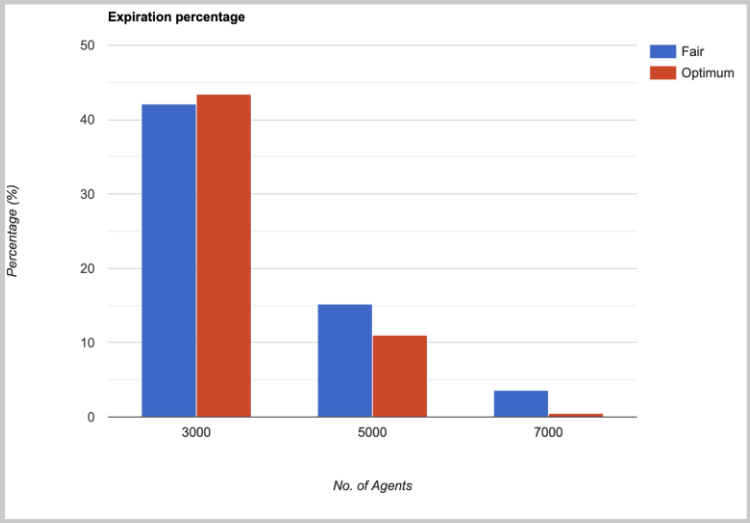
It is evident from the graph that there are two major factors that are influencing namely temperature and vacation months. Ride hailing or cab utilization is much lesser in summer and during vacation months.

People prefer using their own vehicles during summer. Therefore, the month of August shows a lesser expiration percentage due to less volume of cab hailers. December, being a vacation month, also has a lesser expiration percentage compared to other months in the winter season.

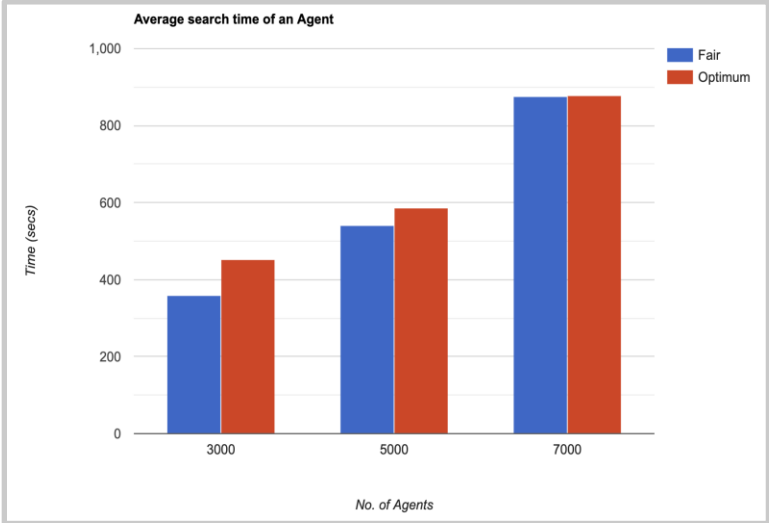
The anomaly in August is also verified in the graphs below as it shows a higher average agent search time and a lower average resource wait time.

In all the three plots, the results for both Fair & Optimum follow a similar trend.

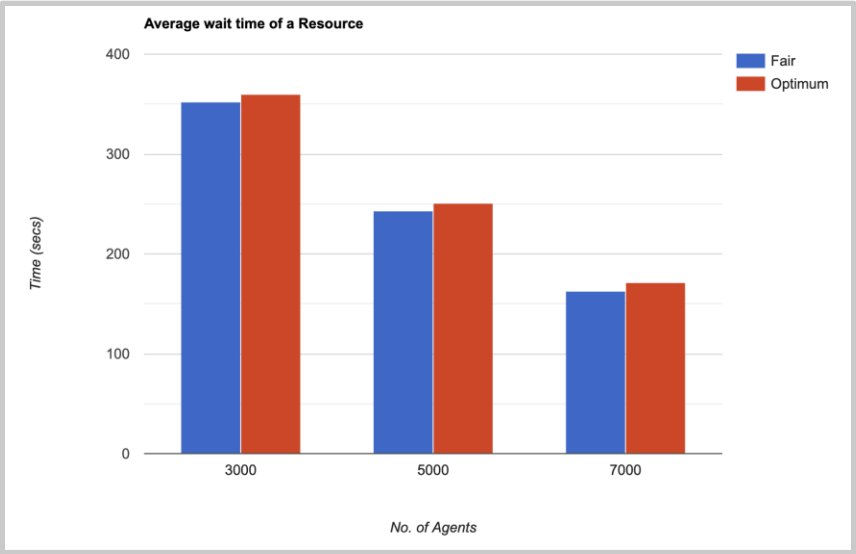
5.2 Trend by changing the number of agents



RESOURCE EXPIRATION PERCENTAGE



AVERAGE SEARCH TIME OF AGENT

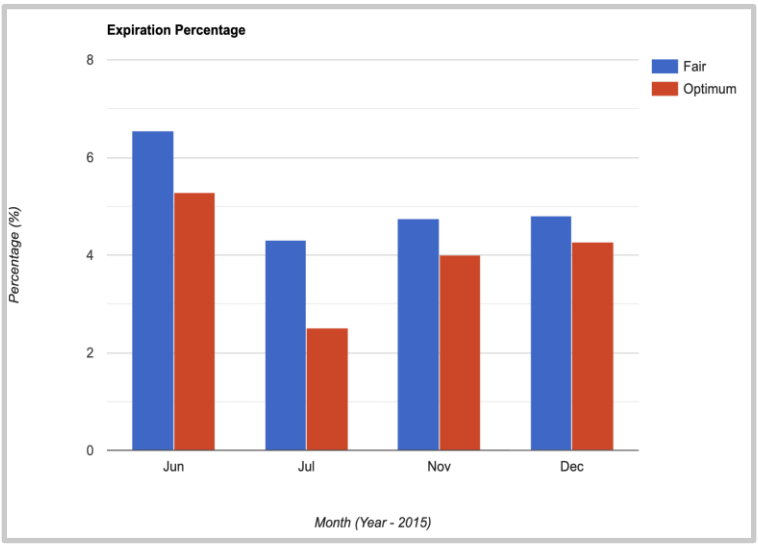


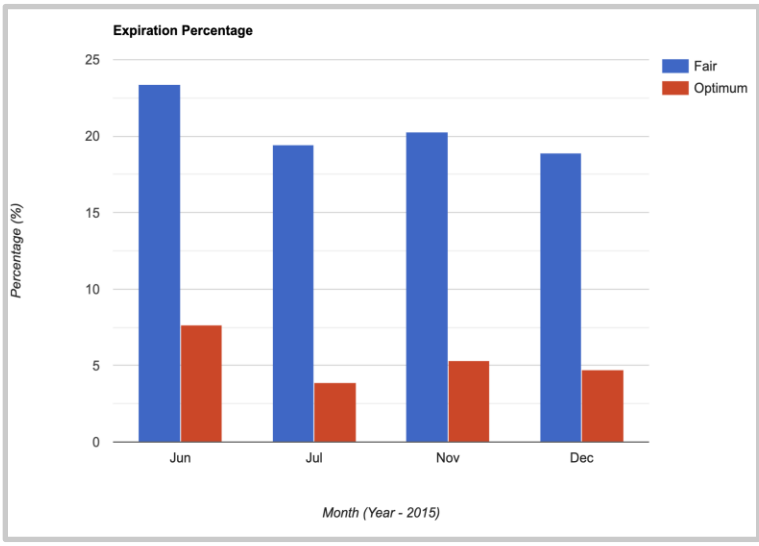
AVERAGE WAIT TIME OF A RESOURCE

From the graph, it was observed that the number of agents play an important role in deciding the profit of the company. With an increase in the number of agents, there tend to be more assignments.

However, it is also possible that increasing the number of agents could cost the company more financially. The below graph helps us make a more informed decision. It shows that having 7000 cabs operating in the city has a higher number of cabs driving without resources, thus resulting in lower profitability for the company.

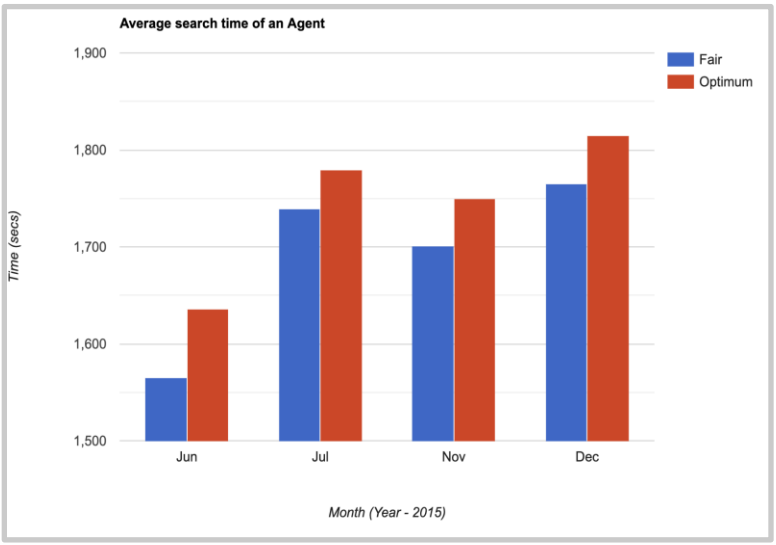
5.3 Day/Night Comparison

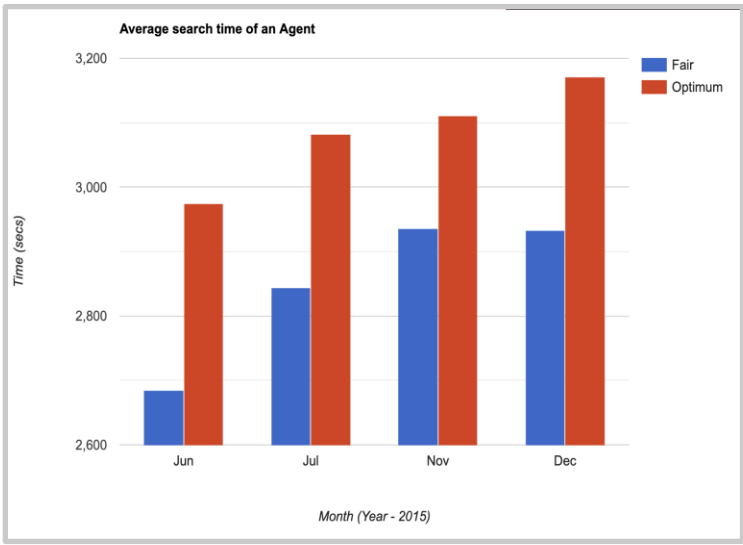




Day **Night**

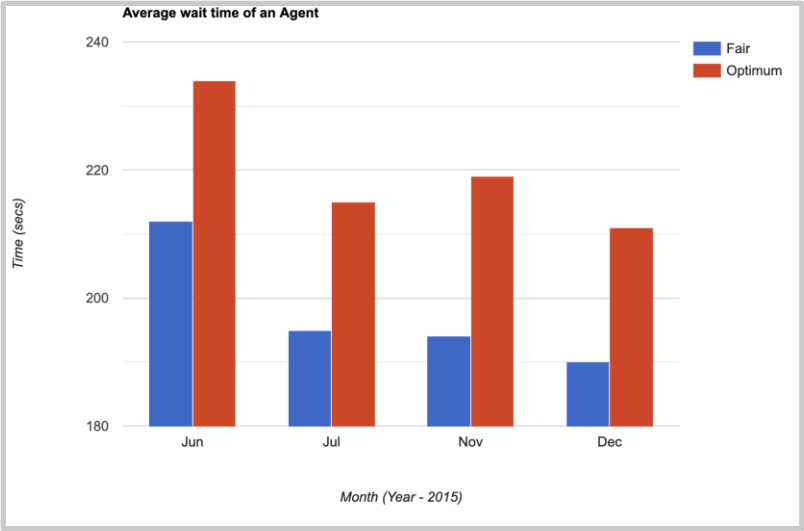
EXPIRATION PERCENTAGE

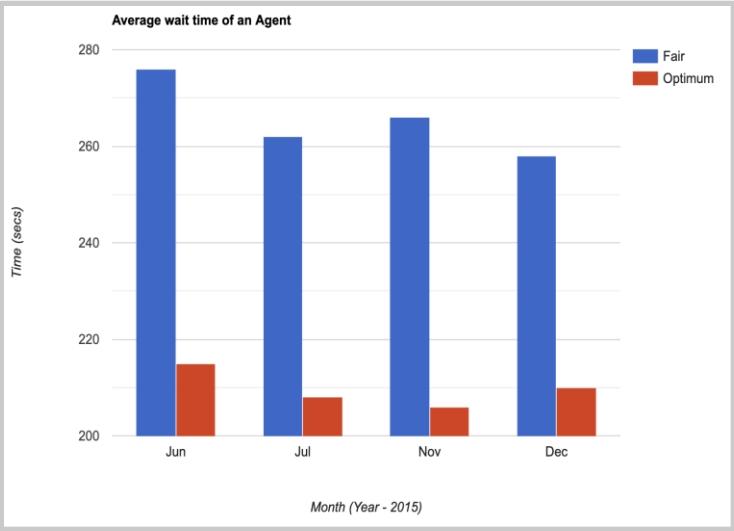




Day **Night**

AVERAGE SEARCH TIME OF AGENT





Day **Night**

AVERAGE WAIT TIME OF A RESOURCE

As is evident from the graph, cab utilization during the day is almost evenly strong, however during the night, ride hailing is less prominent and shows a low expiration percentage.

During night, most of the cab utilization is usually between the hours 6 PM and 10 PM after which there is a huge drop in the number of resources. Whereas, during the day it's equally spread out through the hours.

The graph also shows that the optimum algorithm has pretty similar expiration percentages throughout Day and Night handling high demands in a short period in such a way to make more profit.

Generally, in our plot we see that Optimum Algorithm has higher search time than Fair. This is followed in the above plots as well.

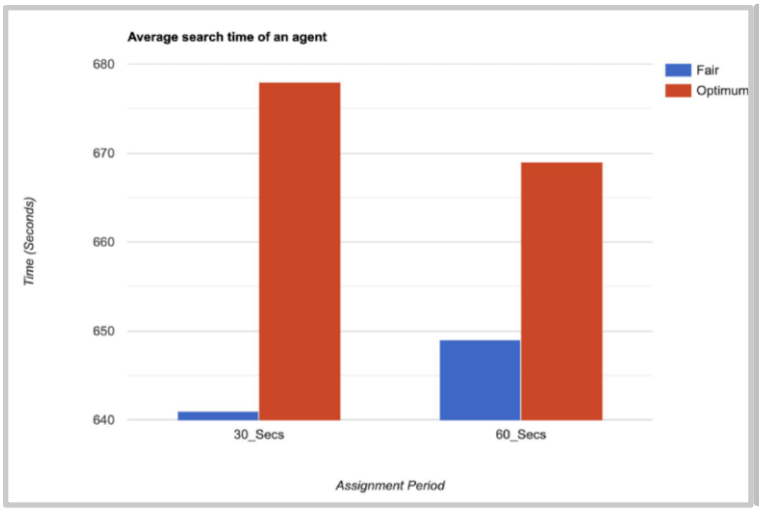
The increase in difference during the night can be due to the fact that after 10 PM, there would be minimal number of resources since most of the requests are concentrated between 6 PM and 10 PM.

The optimal approach has a higher agent search time at night due to lesser number of requests at that time and a significantly lower resource wait time as compared to the fair approach.

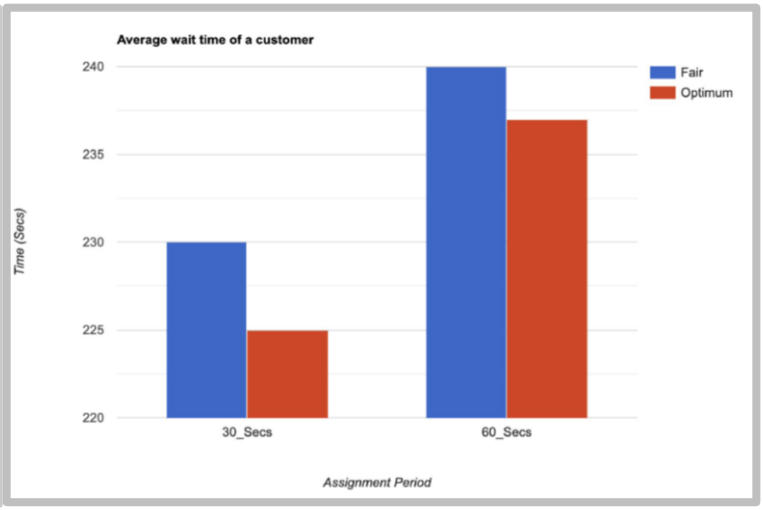
5.4 DEMO RESULTS:

Data set A

Experiment Parameters	Values
Number of Agents	5000
Resource Expiration Time	10 Mins
Assignment Period	30 Sec, 60 Sec
Dataset	May 17th 2016



AVERAGE SEARCH TIME OF AGENT

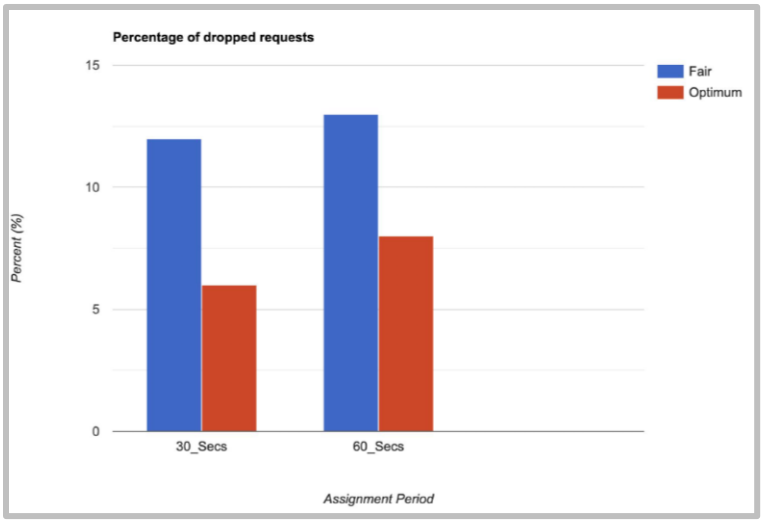


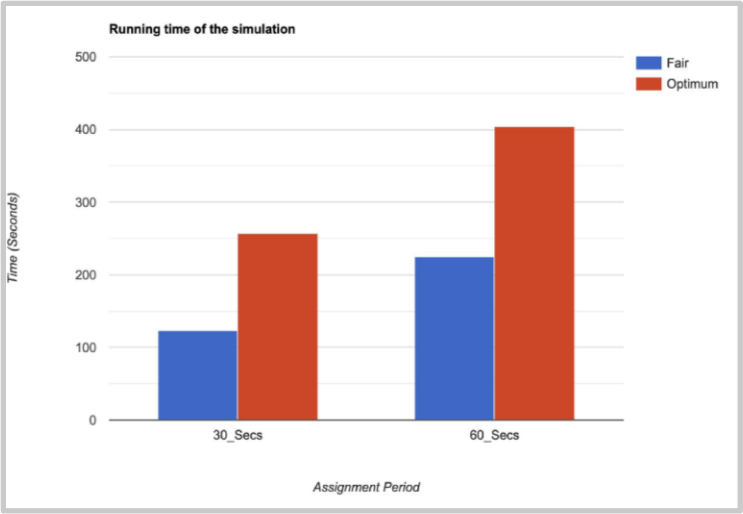
AVERAGE WAIT TIME OF A RESOURCE

The results for the average search time of an agent showed that the search time was much higher in the optimum algorithm than in the fair algorithm. The average search time also increases with the pool size.

The results for the wait time for the customer were contrary to the average search time. The wait time was higher for the fair algorithm than the optimum algorithm.

However, similar to the average search time, the wait time also increases with the pool size.



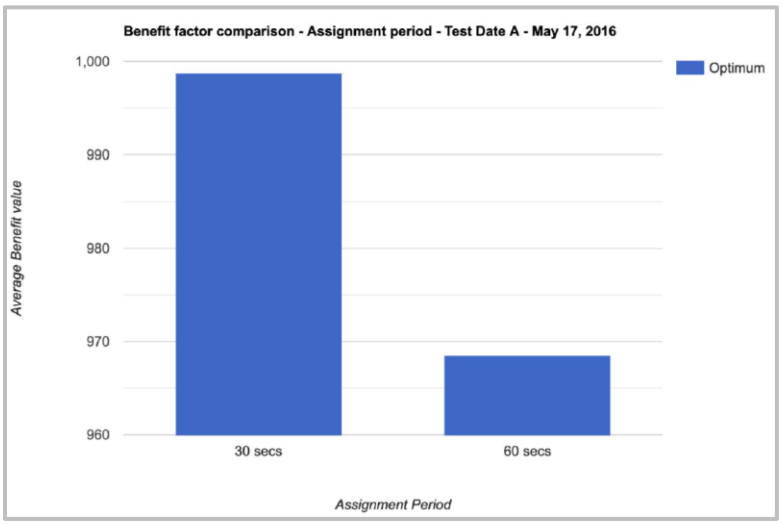


RESOURCE EXPIRATION PERCENTAGE

RUNNING TIME OF SIMULATION

The results of the dropped requests also showed that it was lesser for the optimum algorithm than for the fair algorithm. The percentage of dropped requests also increased with an increase in pool size.

The running time of the algorithm also showed results where the fair optimum was much lesser than that of the fair algorithm.



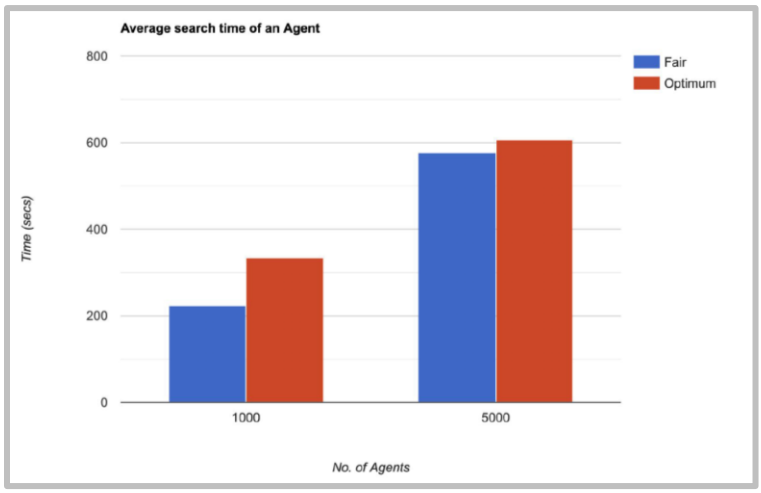
BENEFIT FACTOR

The average benefit per agent in the optimum algorithm showed results that the 60 second assignment was much lesser than the 30 second window.

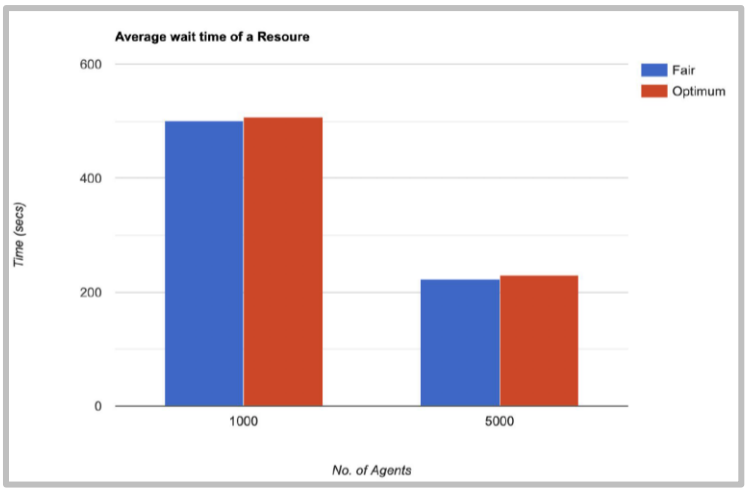
These results are inversely proportional to the results of the expiration percentage.

Data set B

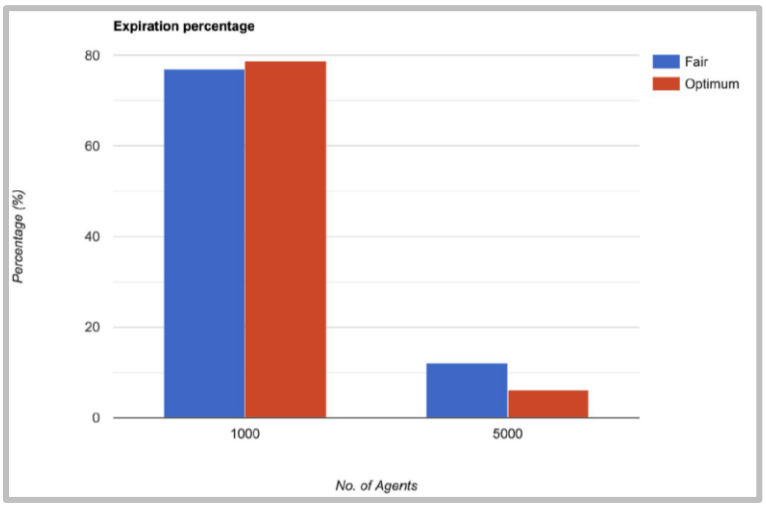
Experiment Parameters	Values
Number of Agents	1000, 5000
Resource Expiration Time	10 Mins
Assignment Period	30 Sec
Dataset	May 20th 2016



AVERAGE SEARCH TIME OF AGENT



AVERAGE WAIT TIME OF A RESOURCE



RESOURCE EXPIRATION PERCENTAGE

6. Issues raised during final presentation

Results showed consistent trends for different days. They also matched with other teams using the same parameters.

Our optimal approach showed a significantly lower expiration percentage and a higher agent search time as compared to the fair approach.

The only issue raised was the reason why Optimal has a higher search time than Fair. This is the area with scope for future work as there is a need for a deeper analysis to find if such an anomaly exists and the conditions under which this happens.

7. Collaboration and project management

Task	Ownership
Data pre-processing and filtering	Vidhyasagar
Understanding COMSET and amending to accept pool of resources	Aditya
Fair Algorithm	Praveen
Optimal Algorithm	Sai Krishnan
Benefit Calculation	Sai Krishnan
Performance Improvement	Aditya
Code Refactoring and Merging	Vidhyasagar
Graphical Representation of Results	Sai Krishnan/Praveen
Running the algorithm for different number of agents	Vidhyasagar
Running the algorithm for different resource pools	Aditya
Running the algorithm for the entire year and doing a monthly comparison	Sai Krishnan
Running the algorithm and doing a day vs night comparison	Praveen
Comparing results and presentation and Report	All

Collaboration tool used: Google Hangouts, GitHub

8. References

1. https://github.com/Chessnl/COMSET-GISCUP/
2. https://en.wikipedia.org/wiki/Hungarian_algorithm
3. https://www.geeksforgeeks.org/hungarian-algorithm-assignment-problem-set-1-introduction/
4. https://en.wikipedia.org/wiki/Matching_(graph_theory)
5. https://tapptitude.com/blog/ride-sharing-apps-traffic-solution/
6. https://www1.nyc.gov/site/nypd/stats/traffic-data/traffic-data-collision.page
7. https://www1.nyc.gov/html/dot/html/about/datafeeds.shtml
8. https://github.com/KevinStern/software-and-algorithms/blob/master/src/main/java/blogspot/software_and_algorithms/stern_library/optimization/HungarianAlgorithm.java
9. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page