Exploring Duration and Distance for Delivery Route Optimization

Saranya Venkatasubramanian
School of Computing,
Dublin City University,
Dublin, Ireland.
saranya.venkatasubramanian2@mail.dcu.ie

***Abstract*— The objective of this project is to evaluate driver's daily routes against an optimized route, taking into account bias and errors in estimating driving times. The first phase of the project compares actual driving time between global positioning system locations from different providers with estimated driving time. The analysis carried out is to perform from a single day route to aggregated level of daily route. The second phase compares the actual distance with the calculated optimized route distance computed by different route optimization providers (Google Maps, Mapbox, and GraphHopper) with historical inferences from the datasets.**

Keywords—Travelling salesman problem, Vehicle routing problem, Data analysis, Route optimazation;

I. Introduction and Motivation

There are potential cost saving opportunities in the logistic sector with customer communications and route optimization i.e. bridging the gap between drivers and their ultimate customers. This problem is addressed by a start-up, Xpreso Software Private Limited based in Dublin, Ireland. Customer communication with different drivers is dealt through implementing strategies with neural networks with driver feedback about the routes [1]. Route optimization maintains overall profitable services by cutting down on operating cost by using optimized routes for deliveries [2]. The route optimization problem is a variant of vehicle routing problem or travelling salesman problem. This project has utilized existing historical data records to explore options to optimize and understand driver routes.

Route optimization in the logistic sector refers to scheduling a route where the priority is to ensure the most profitable delivery costs through taking the shortest distance path or a path with least duration and also maintain overall service objectives such as on time delivery [4, 5]. The optimized route's goal is to reduce logistic company's operating costs, which can be minimized by total route distance and average miles between the stops. Key elements of route optimization includes the underlying road network given my different road maps service providers such as Google Maps. Mapbox, TomTom, GraphHopper, HERE or OpenStreetMaps and the degree of configurability that is setting priority on time windows [6, 7]. Different route optimization providers are available in market and they rely on Meta heuristics to find route sequence solutions. These route providers use different routing algorithms such as travelling salesman problem or vehicle routing problem. Travelling salesman problem is a mathematical solution for finding shortest distance among different cities. Vehicle routing problem deals with giving a solution to the set of routes for fleet within a given time frame.

In section II, related previous research are summarized that discusses the routing optimization problems, different routing APIs available in markets, their underlying techniques and efficient uses of APIs. In section III, the data modelling techniques that are used to introspect the dataset are discussed in detail with some initial analysis and further the project workflow is discussed. In section IV, evaluation of results and analysis are discussed in details. In section V, future work and conclusion of this project is discussed.

II. Literature Review

Ref [1, 2, 3], Authors have discussed about the underlying different scenarios were vehicle routing problem and travelling sales man problem are discussed with different optimal solutions in their research, these papers helped in understanding the different delays and insufficiencies that may occur while devising route optimization and fleet time windows..

Ref [4, 5], Authors have presented a specific window time biases and its effects in solutions of travelling salesman problem, these biases are taken into consideration while modelling and analyzing this explorer tool. Usages of these APIs are explained with different routing technologies. Ref[6], Authors have given an overview on how to efficiently use Google Maps, Mapbox and GraphHopper routing APIs, these efficient usability traits where taken into consideration while selecting these APIs when there were lots of routing APIs available in the market. Official website examples for developers were referred too.

Ref [7, 8, 9], Authors have discussed the significant use of different time windows in a fleet for route optimization using vehicle routing problem. In Ref [9], the authors have explained better, the road network into different time windows by segregating the locations within a city network. This explains the importance of usage of windows in within the fleet. This can used in route optimization of a driver in logistic sector as well, where, the drivers deliveries can be grouped in different time frames and hence enhancing his optimal fleet for the day.

III. Data Modelling and Workflow

A. Data Collection

The dataset was provided by Xpreso Software Pvt. Ltd for carrying out this project. This dataset consists of courier deliveries with details of the deliveries i.e. Driver_Name, Finish_Time, Longitude, Latitude, and Job_Id. The excel workbook consisted of 308836 rows and 5 columns as defined above. Addressing the issue of privacy, data was anonymized before it was provided.

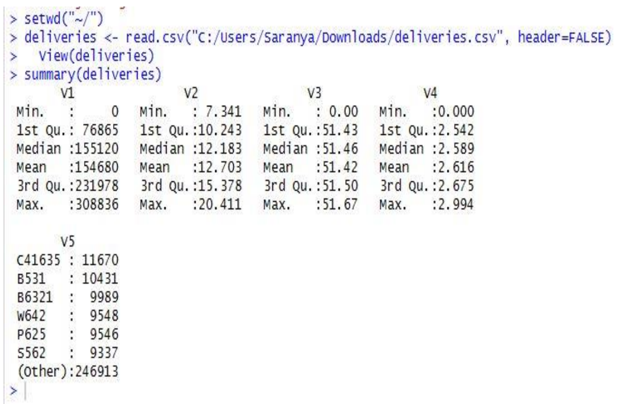
B. Initial Analysis

i. Assumptions

While considering routes for logistic services, the fleet starts from depot and it finishes with depot. Driver can randomize the fleet as per his requirements. Each driver is considered to be delivering unique couriers. The time windows for each driver is different and may change as per driver's wish. Driver visits each courier destination only once in his fleet.

ii. Data Cleansing

Dataset was summarized and plotted for initial understanding of data. It was noticed that columns named Longitude, Latitude and Finish_Time had zero as minimum value and column named Driver_Name had null value. The maximum value in the column Longitude was also noticed to be suspicious. In Fig.1 the columns V1, V2 and V4 shows the minimum values as zero.



```
> setwd("~/")
> deliveries <- read.csv("C:/Users/Saranya/Downloads/deliveries.csv", header=FALSE)
>   View(deliveries)
> summary(deliveries)
       V1               V2               V3              V4
 Min.   :     0   Min.   : 7.341   Min.   : 0.00   Min.   :0.000
 1st Qu.: 76865   1st Qu.:10.243   1st Qu.:51.43   1st Qu.:2.542
 Median :155120   Median :12.183   Median :51.46   Median :2.589
 Mean   :154680   Mean   :12.703   Mean   :51.42   Mean   :2.616
 3rd Qu.:231978   3rd Qu.:15.378   3rd Qu.:51.50   3rd Qu.:2.675
 Max.   :308836   Max.   :20.411   Max.   :51.67   Max.   :2.994

        V5
 C41635 : 11670
 B531   : 10431
 B6321  :  9989
 W642   :  9548
 P625   :  9546
 S562   :  9337
 (Other):246913
> |
```

Fig. 1 Dataset Summary

Thus, the below cleansing process was taken into consideration.

- A program was written in java programing language to group the dataset according to driver names and write them in different sheets where each sheet in the excel book consisted of distinguished drivers. It is noticed that after grouping the drivers there were 83 unique drivers in the dataset with 500+ average deliveries/day.
- Using online Google Fusion Tables tool, dataset was further checked with zeros in both the columns Longitude and Latitude in each row. The rows with zero values were deleted.
- After grouping the dataset with driver name, a java code was again scripted where, similar latitude and longitude where classified into drivers. For this model, the dataset was divided into three parts i.e. 50%-25%-25%. According to the model, prior probabilities of classifiers were calculated and further used to allocate the driver names to the respective latitude and longitude with a minimum probability being 0.65. Whenever, the class prior probability is lower than 0.65, the row was purged from the dataset.
- As mentioned earlier, Longitude column had signs missing. In each sheet of the excel workbook, the column was sorted in descending order and the rows that were without signs were multiplied by '-1' in paste special option by copying '-1' in a cell in the sheet. Thus, changing the sign of the number.
- After, cleansing process the dataset had 303489 rows.

iii. Selecting APIs

As mentioned above there are many routing service providers available in the market such as TomTom, HERE, OpenStreetMaps, etc. Criteria was to select API wherein the underlying algorithm of routing optimization is formed from either vehicle routing problem or travelling salesman problem. Most of the APIs in logistic sector use the vehicle routing variants to perform routing optimization such as TomTom, HERE, Google Maps etc. GraphHopper was one such API implemented in java programming language with travelling salesman approach. Implementation wise, GraphHopper API was easily understandable and easily available. Mapbox also had APIs in java programing language wherein the route optimization had both the options of vehicle routing algorithm as well as travelling salesman. Google maps also has implementation in java programming language.

C. Workflow

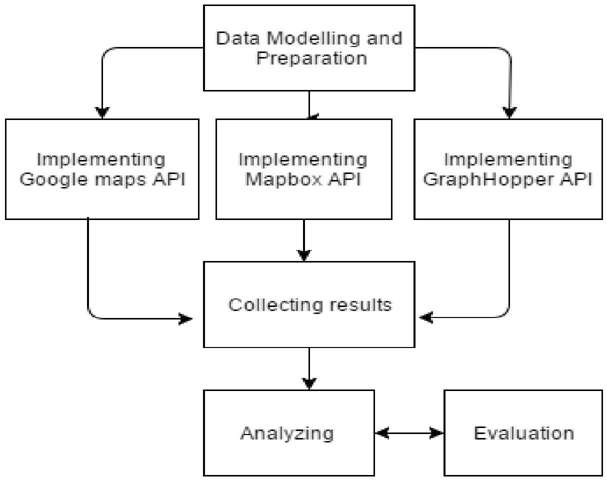


Fig. 2 Project Flowchart

The Fig.2 represents the work flow of the project, which had three main phases- Data Modelling, Implementing API and Analyzing, Evaluation

i. Implementation of APIs

There were three APIs (Google maps[1], GraphHopper[2] and Mapbox[3]) selected for creating a distance matrix and as well as calculating accurate duration taken for the same route in these different APIs. Some limitations and restriction from the service providers were addressed such as limitations on number of waypoints as a free user, maximum requests that can be sent per day, especially in Google Maps and GraphHopper. The restrictions were addressed by creating more accounts as free user and were put in loop to address the waypoints in java programing language.

ii. Collecting Results

Results collection was to be handled carefully as the output was exponential for the distance matrix. Storing them in excel workbook and threaded functionalities were used to handle each sheet and its corresponding matrix and duration calculation. Durations for each leg was stored separately and duration was also calculated with the difference in the column Finish_Time within each leg for further analysis.

iii. Analyzing

First phase of analysis was carried out by analyzing actual time difference calculated from each leg of dataset and the estimated time output given by all the three APIs. Within this phase the following things were carried out.

- Each leg estimated duration was compared with actual leg duration computed from the Finish_Time column. The time differences were ranked into 1, 2 and 3 for different APIs and results were stored for all the drivers with ranks, API name and difference between the actual time and all the other APIs.
- The optimized route given by APIs with durations was further compared with the actual time taken by the driver and results were stored as per the ranks.
- Both the above outcomes were evaluated in the evaluation phase.

Second phase of analysis was carried out for calculating each leg distance with different APIs and analyzing them with optimized route given by the distance matrix calculations of the same.

- Each leg estimated distance is compared with actual leg distance given by various APIs and the differences in the durations are ranked into 1, 2 and 3 for different APIs and results were stored for all the drivers with ranks, API name and differences between the actual times of different APIs.
- The outcomes were evaluated in evaluation phase.

iv. Evaluation

Evaluation was carried out to answer three questions as stated below:

- Which service provider user show worst driving time, with the help of ranks per leg?
- Which service provider user shows best/worst driving time, when and why it is happening?
- Which one of the driving time service is best? And worst? According to the ranks and how much is the gap?

These questions are further elaborately discussed in details with actual results and their graphical representations in the upcoming section.

IV. Results Presentation

In the Fig.3 and 7, summary of durations and distance for each leg. It shows that Mapbox and Google maps are relatively similar and GraphHopper comparatively is higher than the other. In the upcoming results, the interpretation of summary is checked.

```
> Duration_APIs <- read.csv("C:/Users/Saranya/Desktop/Ranks.csv", header=TRUE)
> summary(Duration_APIs)
 Dur_GraphHopper Dur_GoogleMaps    Dur_Mapbox
 Min.   : 1.50   Min.   : 1.10   Min.   : 1.10
 1st Qu.: 6.70   1st Qu.: 5.80   1st Qu.: 5.80
 Median :12.00   Median :10.50   Median :10.50
 Mean   :11.96   Mean   :10.54   Mean   :10.55
 3rd Qu.:17.20   3rd Qu.:15.30   3rd Qu.:15.30
 Max.   :22.40   Max.   :20.00   Max.   :22.30
>
```

Fig. 3 Durations Summary from APIs

In the Fig.4, we can see the variations in the time differences and the ranks assigned to APIs for each driver. The below figure shows that ranks assigned to different APIs are majorly due to the difference in underlying routes optimization techniques i.e. application of TSP and VPR. Travelling salesman isn't consistent with fleet constrains such as traffic_time, school_road, diversions, under_constructions, construction_diversions, etc.

Row Labels	MapBox	GoogleMpas	GraphHopper
⊞1	81518	86377	33579
⊞2	60720	72792	66756
⊞3	39978	40920	80898
Grand Total	182216	200089	181233

Fig. 4 Difference in API Ranks

[1] https://github.com/googlemaps/google-maps-services-java
[2] https://github.com/graphhopper/directions-api-java-client
[3] https://github.com/mapbox/mapbox-java

The Fig.5 and Fig.6, shows the boxplot of fleet distances/durations per leg computed by three APIs. Google maps has lowest distance calculated in each fleet per leg, whereas GraphHopper has slightly higher estimated distances. Mapbox looks similar to Google maps.

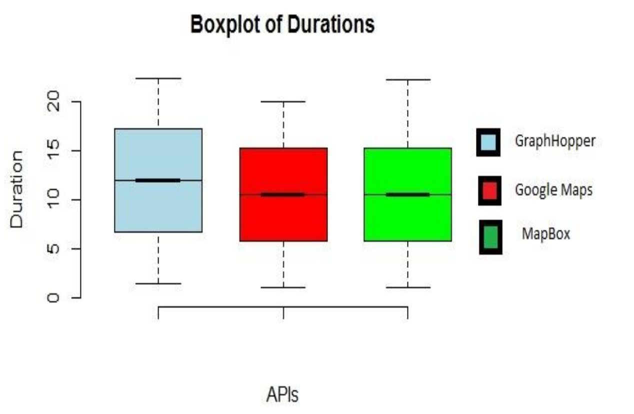


Fig. 5 Boxplot of Time Differences

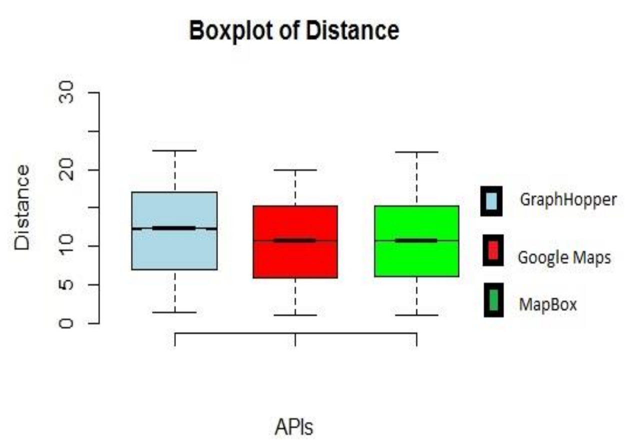


Fig. 6 Boxplot of Distance Variance

In the Fig.8, we can see the bars that show the ranks of estimated route fleet time per leg from different APIs. Google Maps has the highest number of first ranks whereas GraphHopper has highest number of third rank and also lowest number of third rank and first rank respectively. Earlier, we discussed the underlying route optimization technique used by Google maps and Mapbox is vehicle routing problem whereas Graph Hopper's route optimization is build using travelling salesman problem. GraphHopper looks consistent with better number of second ranks though travelling sales man problem with constraints of tolls, school_hours, and traffic_time doesn't seems to be giving best optimistic route.

In Fig.9, we can see the bar graph that shows the summary of the ranks of different APIs for estimated fleet distance. Google Maps (GM) is just a bit higher in terms of first and second rank with Mapbox (MB). GraphHopper (GH) seems to be very low in first rank though, in second rank it is consistent at par with Google Maps and Mapbox. Distance estimation has less affect in travelling salesman problem as we can see GraphHopper is consistent like the other two.

Vehicle routing problem gives emphasis on fleet time window and fitting the deliveries within the fleet time. The route optimization depends on fleet duration rather than the fleet leg distance. Travelling salesman problem gives significance to the distance calculated between the two destinations, which in real scenario doesn't seem to be optimal fleet. Route Optimization in logistic sector is preferred minimum duration fleets wherein most deliveries can be accommodated. The ranks in the Fig.7 and Fig.8, clearly shows that VPR gives a better route.

```
> summary(Distance)
  Dis_MapBox       Dis_GoogleMaps    Dis_GraphHopper
 Min.   : 1.50    Min.   : 1.50    Min.   : 1.50
 1st Qu.: 9.80    1st Qu.: 9.80    1st Qu.: 9.80
 Median :18.20    Median :18.30    Median :18.30
 Mean   :18.24    Mean   :18.24    Mean   :18.26
 3rd Qu.:26.60    3rd Qu.:26.60    3rd Qu.:26.70
 Max.   :35.00    Max.   :35.00    Max.   :35.00
> |
```

Fig. 7 Summary of Distance

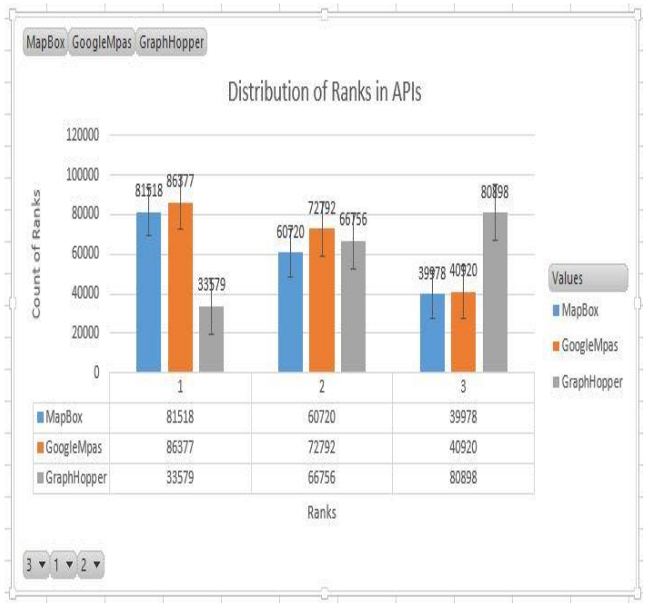


Fig. 8 Bar Graph of Ranks (Duration of Fleet)

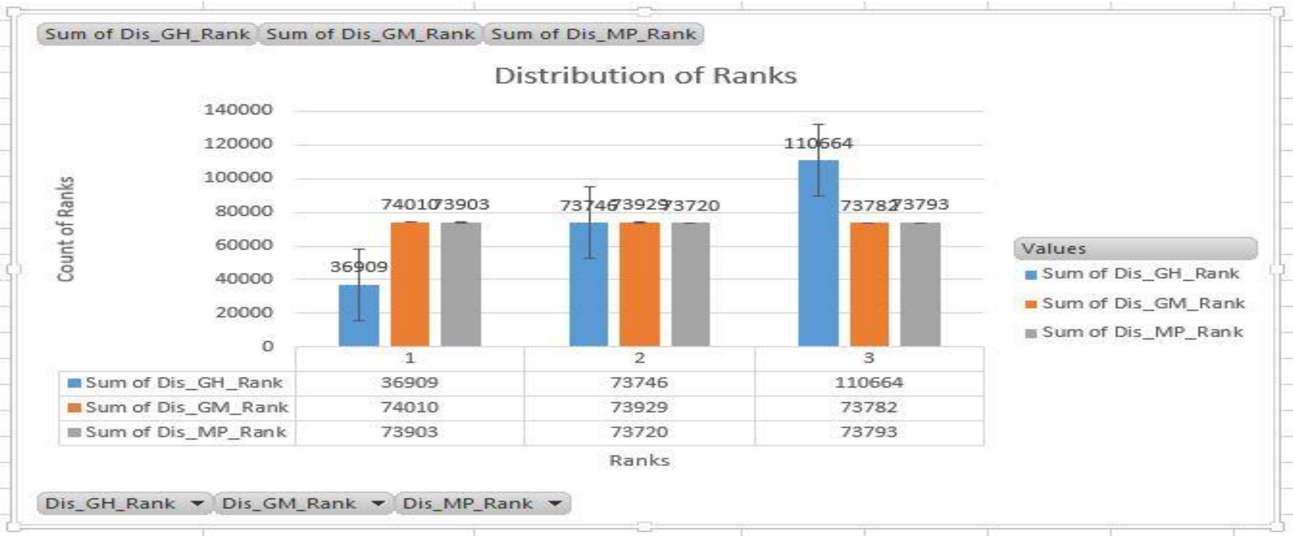


Fig.9 Bar Graph of Ranks (Distance of Fleet)

```
> head(Stacked_Groups)
  values    ind
1   15.9 Group1
2   14.6 Group1
3    5.8 Group1
4   11.0 Group1
5   13.9 Group1
6   16.3 Group1
> Anova_Results <- aov(values~ind, data = Stacked_Groups )
> summary(Anova_Results)
                Df   Sum Sq Mean Sq F value Pr(>F)
ind              2   404237  202119    6267 <2e-16 ***
Residuals   910464 29363922      32
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Fig.10 Anova Results of Fleet

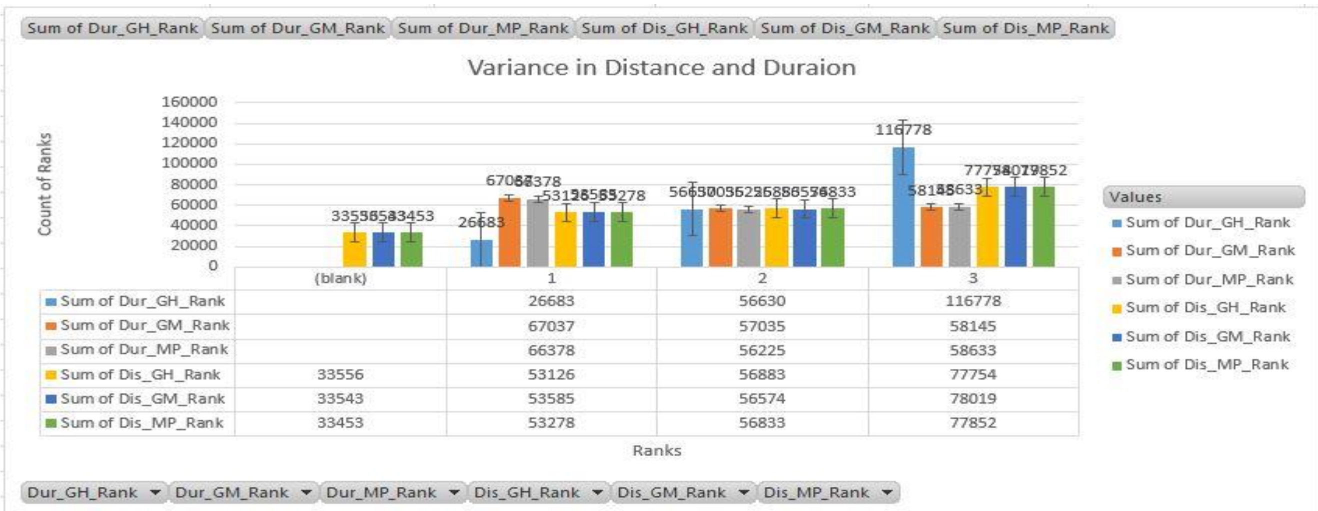


Fig. 11 Bar Graph showing variance

The Fig.10, shows the analysis of variance of all the estimated fleet leg distances and durations. This analysis was conducted the find out the significant variations among/between the group. Null hypothesis was to test if the means of the group are equal where, groups are APIs estimated fleet duration and distance. At the degrees of freedom of 2 and 910464 i.e. F (2, 910464) the f-value calculated is 6267 and p value from table is 0.000000216. The mean difference is strongly different among groups as the mean difference is more times the standard error. Hence, even though the durations and distances where calculated for the same routes, they strongly different as they are calculated with different APIs and different routing optimization technique.

The Fig.11, shows the aggregated fleet route duration and distance estimated by different APIs with the Ranks allotted to them. The aggregated level fleet i.e. the route of all the deliveries a driver does on a day. The calculated average time fleet time of a driver per day is 5 hours 15 mins and average deliveries per driver is 350. The fleet distance estimation in all three APIs were more or less similar as the differences weren't significant as in terms of per leg distance estimation. This shows that the significance is given to optimal solution on a bigger scale than in per leg optimization of route. In vehicle routing problem, each leg is checked to fit the fleet time that is already a criteria that is set.

The above results present that, optimization of fleet durations gives more optimal solution. The two APIs Mapbox and Google maps, have their underlying routing technology as that of vehicle routing problem, which works well in a logistic sector, because the fleet time and the windows in the fleet are already set and deliveries are assigned accordingly. Whereas, traveling salesman problem deals the same problem in a different scenario, it deals with minimizing the per leg fleet distance, which doesn't take duration into consideration. GraphHopper works well with distance optimal routes, but, the fleet duration is non consistent. In the figure 9, distance and durations of fleets were tested in two groups, wherein the result was the means aren't equal which shows distance and duration and not correlated or less correlated.

V. Conclusion And Future Work

Thus, the objective of the project to evaluate daily route of the driver against an optimized route calculated by different APIs taking into account the delays and biases in route networks of different APIs is completed successfully. The comparisons of different route optimization service providers showed Google Maps and Mapbox optimized routes were faster with respect to time. The analysis effect of distance on duration and duration on distance in different APIs showed the difference in underlying technique is significant.

This project can be further used in a machine learning environment to further model the error and bias of optimized routes and also the underlying techniques of route optimization can be enhanced more with the error and bias model. A study can be carried out to minimize fleet time error and this can be used to further optimize the solution and also the windows for the fleet time can be enhanced further by maximizing the window time in fleet, to optimize the route further in a given time frame. Neural networks can be used to further understand and classify the real scenario from driver's feedback, which may led to a better understanding of the read maps and can enhance the route optimization process.

Acknowledgment

Deepest gratitude and heartiest appreciation for the help and support that were extended by the following people who in a way or another have contributed in making this practicum possible.

Dr. Suzanne Little, for her words of encouragement throughout the project, her support, advices, guidance, valuable comments, suggestions and provisions that benefitted much in the completion and success of this practicum.

Mr. Fabiano Pallonetto, for his assistance with datasets, valuable inputs, motivation, his time and effort, his technical expertise that enhanced my knowledge, which helped me to accomplish this practicum successfully.

Dr. Dónal Fitzpatrick, his guidance on various practicum information and support that enhanced the completion of the practicum as per schedule.

Mr. Gary Conway, for his support and help in technical assistance with processor speed and data storage.

References

[1] G. S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," Operations Research, Vol. 21, No. 2 (Mar. - Apr., 1973), pp. 498-516.

[2] David Pisinger , Stefan Ropke, "A general heuristic for vehicle routing problems," Computers & Operations Research 34 (2007) pp. 2403 – 2435.

[3] Canhong Lin, K.L. Choy, G.T.S. Ho, S.H. Chung, H.Y. Lam, "Survey of Green Vehicle Routing Problem: Past and future trends," in Expert Systems with Applications 41 (2014) 1118–1138.

[4] César Rego, Dorabela Gamboa, Fred Glover, Colin Osterman , "Traveling salesman problem heuristics: Leading methods, implementations and latest advances," European Journal of Operational Research 211 (2011) 427–441.

[5] Keld Helsgaun, "General k-opt submoves for the Lin–Kernighan TSP heuristic," Math. Prog. Comp. (2009) 1:119–163 DOI 10.1007/s12532-009-0004-6.

[6] Geoffrey De Smet, "Vehicle routing with real road distances," OptaPlanner lead, 2014, Unpublished.

[7] Christian Prins, "A simple and efective evolutionary algorithm for the vehicle routing problem," Computers & Operations Research 31 (2004) 1985 – 2002.

[8] Jose Carlos Molinaa, Ignacio Eguiaa, Jesus Raceroa, Fernando Guerreroa, "Multi-objective vehicle routing problem with cost and emission functions," Procedia - Social and Behavioral Sciences 160 (2014) 254 – 263.

[9] Gitae Kim, Yew Soon Ong, Chen Kim Heng, Puay Siew Tan, Nengsheng Allan Zhang, "City Vehicle Routing Problem(City VRP): A Review," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, August 2015

APPENDIX

[1] https://github.com/SharuVS/RouteExplorerTool

The above link is Github repository of this project, contains java codes for different APIs, Outputs from these APIs and document which explains this project.

[2]https://drive.google.com/file/d/0B4kmbHjLp43dakFaaGVxdWI0V28/view

The above link is of the dataset used in this project.

[3]https://drive.google.com/open?id=0B_dINEjYWq4Jd0xVTFZPZEZDSEk

The above link is the papers those are specified in reference section of this project report.