Finding the fastest navigation rout by real-time future traffic estimations

Nadav Voloch[1], Noa Voloch-Bloch[2]

1. Department of Computer Science, Ben Gurion University of the Negev, Be'er-Sheva, Israel

2. Voloch Knowledge Horizons (Afikey Yeda), Israel

***Abstract* — Computing the fastest rout in traffic is a long researched problem, with several solutions to it. Most of these solutions consider the current status of the road in a given time, including vehicles that are currently in the planned rout. The problem with this approach is that vehicles that are currently placed in a point that will be reached within a certain amount of time, will no longer be there at the arrival time. This creates a problem, that even exists in the most popular navigation application such as Waze and Google Maps, where the ETA is constantly changing during the travel. In this research we suggest a different approach, that finds the fastest traffic rout according to the future positions of vehicles at the designated points in the rout. This approach will give a much more efficient navigation paradigm, in which the estimated travel times are realistic and accurate.**

***Index Terms* — Fastest rout, traffic control, navigation applications, smart traffic, transportation communication.**

I. Introduction

In the past decade, Geographical navigation applications have become a significant part of our everyday lives. If in earlier times they were in use only for military or commercial transportation use, nowadays every basic smartphone device has a built-in navigation system in it, usually in more than one form of application. The algorithms used in these applications become more and more sophisticated, and if a decade ago they were based solely on the given map, today the use is also based on traffic data, given by the users themselves, that generate a real-time environment of users (vehicles) and roads. With this given data, the task of finding the fastest rout between a source and a target, has become more accurate, but also more difficult. Finding the fastest rout is no longer a simple task of finding the shortest rout between a source to a target point, but other factors are to be considered, such as traffic loads in different points of the rout. These traffic loads are usually computed in real-time, as the status of the road is very dynamic by nature, since vehicles are constantly moving, changing the loads in different points.

Navigation applications such as Waze rely on social monitoring of the traffic, having their users as real-time data sources, giving their locations in different parts of the road.

This creates a smarter computation of the routs, that takes into consideration the status of the road in the calculation of the ETA, in purpose of achieving the best time from source to a target point. However, there is an inherently problematic aspect in this form of calculation since the current status of the road is not necessarily its status when the vehicle is getting closer to the target point. There could be a lot of other vehicles getting to the same target point from a different source, making the traffic load much bigger, making the ETA longer than expected. We can see a simple example of the problem in Fig.1, where the rout of the car that departs from Metropolis does not see traffic near Narnia, but the cars that depart from Gotham City are the future traffic load near Narnia, making the real ETA longer than the original one.

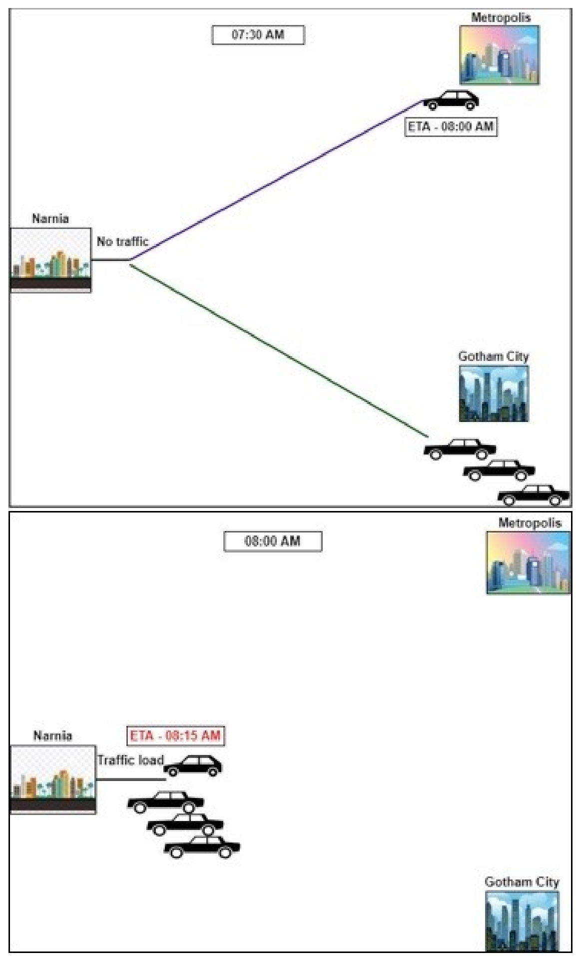

Fig. 1. Gap between planned ETA and actual ETA due to future traffic load not seen in the original rout.

In this paper we suggest a solution for this problem, that involves a smarter configuration of the designated rout that will be based on future estimated traffic status, thus giving a more real-time accurate picture that will help find the actual fastest rout from source to target, giving a more accurate ETA.

II. Background and Related Work

The shortest path problem is the problem of finding a path between two vertices in a graph such that the sum of the weights of its constituent edges is minimized as summarized in [1].

A solution for the shortest path problem is the well-known Dijkstra's algorithm, which finds the shortest path between nodes in a graph, conceived by Edsger W. Dijkstra ([2]). The algorithm was optimized in many research papers such as [3].

In our suggested solution we must take into consideration all different paths from a source point to a target point. For finding all possible paths between source and target nodes, there are several efficient algorithms, such as the Ford-Fulkerson algorithm ([4]), the Edmonds–Karp algorithm ([5]) for computing the maximum flow in a flow network in $O(V E^2)$ time, V being the number of points on the road, E being the number of different roads, and the Dinic algorithm ([6]), also for a maximum flow network, that achieves a better time of $O(V^2 E)$. In our solution we need to find all the paths to determine which path (rout) will be the optimal one from a source point to the target point, which is the same problem covered by the Dinic algorithm mentioned above.

In [7] there is a combined solution that uses the Dinic algorithm for finding all of the paths between a source to a target and finds the most efficient path considering the points as weighted ones that hold values for the Knapsack optimizing problem.

Navigation systems for vehicles were the research topic of many papers, beginning with early ones such as [8], [9] and [10], that relate to topic theoretically, with the technology of navigation by applications being relatively primitive at the time, considering software and hardware that were not that viable.

The research done in the past few years is a much more accurate one, considering the vast technological changes happening in this field of automatic navigation systems. [11] describes a strategy considering the interaction between power systems and traffic network. [12] presents a holistic solution for system-wide load balancing creating an optimizes method for navigation. In our previous work ([13]) we used Large Distributed Dynamic (LDD) graphs for detecting the fastest path in a dynamic traffic graph, which is the basis for this research, taking into consideration the different states in traffic according to different points in time. This creates a much more accurate picture since in uses a dynamic infrastructure, that resembles real-time traffic changes. In the past few years traffic load research has also flourished.

In previous work ([14]) an interesting research is presented, that involves the novel approach of putting social priorities in a smart junction traffic load estimation, so that the distributed time will reflect social fairness. This research direction continued also in [15], where real-time parameters were added, making the algorithm more realistic and accurate, and in [16], where a prototype simulator was built, giving better results for the simulation of the algorithm.

In [17] the algorithm was further developed, including an auction-based decision making for the timing of the junction.

All these research papers mentioned above were the basis of our current research, that handles the problem of non-accurate ETAs because of problematic navigation configurations. Our approach will give a much more efficient navigation perception, in which the estimated travel times are realistic and accurate. This will help getting the real fastest rout of travel and managing traffic loads much more effectively.

III. The Real-Time Navigation Model

As mentioned above, in this paper we suggest a model, that involves a smarter configuration of the designated rout that is based on future estimated traffic status, giving a more real-time accurate picture that will help find the actual fastest rout from source to target, giving a more accurate ETA.

An example for such an approach is presented in Fig.2, where there are two different routes for the car that departs from Metropolis. In the first option the calculated rout at 07:30 AM does not see traffic near Narnia, giving an ETA of 08:00 AM, but the cars that depart from Gotham City are the future traffic load near Narnia (at 08:00 AM), making the real ETA longer than the original one (08:15 AM). On the other hand, the second option shows an ETA of 08:20 AM, because it sees the traffic load in this rout near Narnia, but on 08:00 AM these cars are already gone, thus the real ETA is 08:05 AM. Our model gives the more accurate ETA, because it calculates the route with the future traffic predictions, not considering the current traffic of distant points, because it is not relevant in the given time. The actual traffic load at the arriving point will be of cars that are as distant from the target as our vehicle is in a given time.

The manifestation of this process is based on an algorithm we have devised, that takes all of the given paths and vehicles in the environment of the designation point and estimates the traffic load in different points according to the vehicles' arrival time to these points. For this purpose, we will first need to calculate all paths possible for the vehicle that departs from the source point to the target point. This is done by the Dinic algorithm ([6]) mentioned above. Then for each path we take the vehicles that are in a distance that resembles our vehicle from the target point. These vehicles are the actual traffic load in the proximity of the target point. We configure which path has the lowest amount of these vehicles for our choice of rout.

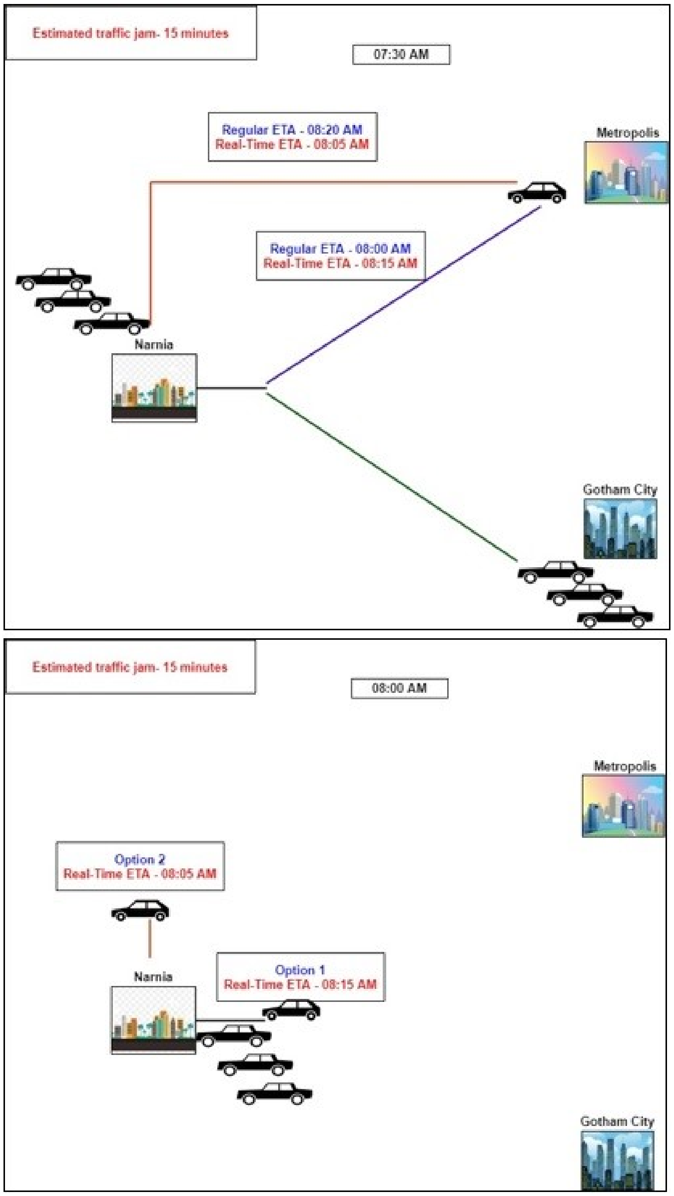

Fig. 2. The difference between the planned ETA and actual ETA in different times of the routs. Option 2 is the one chosen given our approach of calculating future traffic.

In this model we look at all of the roads as a single graph *G*, having the vertices as points on the road. We then find all of the paths in *G*, and for each path we calculate the attribute of total edge weight (tew_{i}), that is the number of vehicles in the same distance from the target as the source. After these assignments we wish to find the path that has the lowest value of tew_{i}, for the purpose of finding the road that has the fewest vehicles near the target point. At this point we also need to take into consideration the shortest path, with no relation to the vehicles in it, because it also has a major impact on the ETA. After tackling both aspects, we need to combine them, and find the optimal path in terms of ETA, thus finding the actual real-time fastest rout. We also refer to the average length of the vehicle, to estimate the traffic loads.

The algorithm is as follows:

Finding the fastest path with future traffic estimations **(Graph *G*, Edges *E*, Vertex *source*, Vertex *target*, Average length of Vehicle *Vehicle_Length*):**

1. Create a list of all paths *source* to *target*:
   1.1. $L^{path} = Dinic\ (G, source, target)$
2. For each path $L_i^{\ path}$ where $1 \leq i \leq |\ L_i^{\ path}\ |$ set attribute of total edge weight (tew_{i}):
   2.1. $tew_{\mathrm{i}} = \sum_{j=0}^{|E|} w^E_{\ j}\ (L_i^{\ path})$
3. For each path $L_i^{\ path}$ find the distance d_i between *source* to *target*:
   3.1. $d_i\ (L_i^{\ path}) = |L_i^{\ path}|$
4. $minETA\_Path = L_0^{\ path}$; $min\_realTime\_distance = d_0$
5. For each path $L_i^{\ path}$:
   5.1. if $(d_i + tew_{\mathrm{i}} * Vehicle\_Length) < min\_realTime\_distance$
   5.2. $min\_realTime\_distance = (d_i + tew_{\mathrm{i}} * Vehicle\_Length)$
   5.3. $minETA\_Path = L_i^{\ path}$
6. Return *min_realTime_distance*, *minETA_Path*

As we can see in the algorithm, the *min_realTime_distance* is calculated with the addition of the planned traffic load of the vehicles that approach the target point. The algorithm that computed the minimal travel time (as a derivative of the distance) from all of the optional paths.
In the next section we can see some simulative results for travel times with and without the model, for comparison purposes.

IV. Simulation Results

For this research we have devised a simulation that will show several cases of random routs from source to target. The results are presented in Table I.
As we can see in the table, in this part of the simulation there were 7 different routs (A-G), with different distances (d_i's) from the target point, and different tew_{i}'s (meaning the number of estimated vehicles near the target around the ETA).
The average length of vehicle was 3 meters, and the average speed of the vehicles was 60 km/h. We can see that the fastest Travel time, without taking into consideration the number of estimated vehicles near the target is in Rout A, where a supposed travel time is 1 hour, but this is without taking into consideration the tew_{i} factor that is relatively high (900), so the real-time travel estimation is actually 1 hour, 2.7 minutes. We can also see that the best rout is Rout F, in which the real-time travel estimation is 1 hour, 1.6 minutes, despite the fact the road is longer than Rout A, but the future traffic is significantly lower (200), thus it achieves the best real-time travel estimation.

TABLE I: SIMULATION RESULTS FOR REAL-TIME ROUT CALCULATIONS, *Vehicle_Length* = 3m, *Average_Speed*= 60 km/h

Rout	d_i	tew_i	Travel length	Real-Time Travel length
A	60	900	1 hour	1 hour, 2.7 minutes.
B	65	700	1 hour, 5 minutes.	1 hour, 7.1 minutes.
C	70	100	1 hour, 10 minutes.	1 hour, 10.3 minutes.
D	68	500	1 hour, 8 minutes.	1 hour, 9.5 minutes.
E	63	600	1 hour, 3 minutes.	1 hour, 4.8 minutes
F	61	200	1 hour, 1 minute.	1 hour, 1.6 minutes.
G	64	300	1 hour, 4 minutes.	1 hour, 4.9 minutes

V. CONCLUSION AND FUTURE WORK

In this research we presented a model that finds the fastest traffic rout according to the future positions of vehicles at the designated points in the rout, thus creating a much more accurate picture of the traffic. This model gives a much more efficient navigation paradigm, in which the estimated travel times are realistic and accurate. This model can help getting the real fastest rout of travel and managing traffic loads much more effectively. We have devised an algorithm to calculate these routes and times and presented simulative results for it. In future aspects of this research, other parameters can be handled in these calculations, such as different types of vehicles, that have different length and speed. This will create a more dynamic environment that is closer to real life traffic. Other developments can be building a comprehensive prototype for traffic simulations that will run the parameters handled in this research. All together this model gives a more real-time accurate picture of traffic that will help find the actual fastest rout from source to target, giving a more accurate ETA

REFERENCES

[1] Abraham, Ittai; Fiat, Amos; Goldberg, Andrew V.; Werneck, Renato F. (2010) "Highway Dimension, Shortest Paths, and Provably Efficient Algorithms". ACM-SIAM Symposium on Discrete Algorithms, pages 782-793.

[2] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". Numerische Mathematik 1: 269–271.doi:10.1007/BF01386390.

[3] Mehlhorn, K., Sanders, P. (2008). "Algorithms and Data Structures: The Basic Toolbox" .199-200. Springer.

[4] Ford, L. R.; Fulkerson, D. R. "Maximal flow through a network" . Canadian Journal of Mathematics. 8: 399–404. (1956).

[5] Edmonds, Jack; Karp, Richard M. (1972). "Theoretical improvements in algorithmic efficiency for network flow problems". Journal of the ACM. Association for Computing Machinery. 19 (2): 248–264.

[6] Dinic, Y.,"Algorithm for solution of a problem of maximum flow in a network with power estimation".(1970). Doklady Akademii nauk SSSR. 11: 1277–1280.

[7] Voloch, N. (2017) Optimal paths of knapsack-set vertices on a weight-independent graph. WSEAS Transactions on Computers, 16, 163-171.

[8] Claussen, H. (1991). Vehicle navigation systems. In Modern Cartography Series (Vol. 1, pp. 225-235). Academic Press.

[9] Dance, F. J., & Thoone, M. L. (1997). Vehicle Navigation Systems: Is America Ready? (No. 970169). SAE Technical Paper.

[10] Burnett, G. E. (2000). Usable vehicle navigation systems: Are we there yet. Vehicle electronic systems, 3-1.

[11] Shi, X., Xu, Y., Guo, Q., Sun, H., & Gu, W. (2020). A distributed EV navigation strategy considering the interaction between power system and traffic network. IEEE Transactions on Smart Grid, 11(4), 3545-3557.

[12] Liu, Q., Kuboniwa, J., Kameda, S., Taira, A., Suematsu, N., Takagi, T., & Tsubouchi, K. (2015, December). Traffic navigation using positioning information and channel quality map for system-wide load balancing. In 2015 Asia-Pacific Microwave Conference (APMC) (Vol. 1, pp. 1-3). IEEE.

[13] Voloch, N., Voloch-Bloch, N., & Zadok, Y. (2019). Managing large distributed dynamic graphs for smart city network applications. Applied Network Science, 4(1), 1-13.

[14] Barzilai, O., Voloch, N., Hasgall, A., Steiner, O. L., & Ahituv, N. (2018). Traffic control in a smart intersection by an algorithm with social priorities. Contemporary Engineering Sciences, 11(31), 1499-1511.

[15] Barzilai, O., Voloch, N., Hasgall, A., Steiner, O. L., & Ahituv, N. (2018). Traffic control in a smart intersection by an algorithm with social priorities. Contemporary Engineering Sciences, 11(31), 1499-1511.

[16] Fine, Z., Brayer, E., Proshtisky, I., Barzilai, O., Voloch, N., & Steiner, O. L. (2019, November). Handling traffic loads in a smart junction by social priorities. In 2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS) (pp. 1-5). IEEE.

[17] Barzilai, O., Giloni, A., Voloch, N., & Steiner, O. L. (2020). Auction Based Algorithm for a Smart Junction with Social Priorities. Transport and Telecommunication, 21(2), 110-118.

.