Personalized Road Networks Routing with Road Safety Consideration: A Case Study in Manchester

Sophie Hayes
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
sophieh98@btinternet.com

Shen Wang
School of Computer Science
University College Dublin
Dublin, Ireland
shen.wang@ucd.ie

Soufiene Djahel
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
s.djahel@mmu.ac.uk

***Abstract*—Urban road congestion is getting worse with the increasing population and car ownership. Traditional solutions, such as increasing road capacity and dynamic control and adaptation of traffic lights, rely heavily on infrastructure support, which limits their wider adoption and practicality. Vehicle navigation systems, such as Google Maps, TomTom, and AutoNavi, are widely used due to the popularization of smartphones. However, these systems normally provide routes with either shortest travel distance or fastest current travel speed, without any consideration of the drivers' route preferences. For example, the safety level of a road is also very important as it often leads to non-recurring congestion that is more difficult to avoid. In this paper, we propose, implement, and test a personalized routing application that allows end-users to flexibly adjust their route preferences among travel distance, estimated travel time, and the safety level. We present the validation results of our application using a realistic dataset from the city of Manchester in England.**

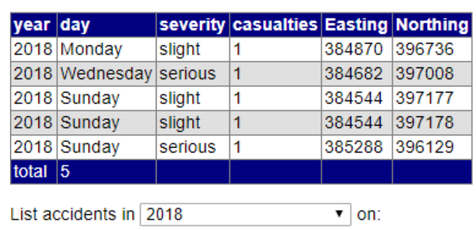
***Index Terms*—Vehicle routing, personalized routing, mobile application, road safety**

I. Introduction

Traffic congestion occurs when the volume of traffic exceeds the roadway capacity and we can distinguish two main types: recurrent and non-recurrent [1]. Recurrent congestion occurs regularly at the same time such as at peak hours in the morning and evening, whereas non-recurrent congestion is caused by randomly occurring events such as an accident or roadworks. In the UK, as the number of licensed vehicles on the roads is expected to increase to between 38 and 42 million by 2050, representing a 30-45% increase from the 2015 number, congestion is expected to continue to increase [2]. As congestion problem has social, economic, and environmental impacts, the situation will get significantly worse if efficient solutions are not developed and used.

Like many cities in the UK, congestion is a serious issue in Manchester. In 2018, it was rated the 4_{th} most congested city in the UK, with congestion costing an estimated £1,157 per person [3]. Also, 156 hours were lost in congestion which was a 2% increase from 2017 [3]. As nationally, the number of licensed vehicles is predicted to increase in the future, which will result in an increase in the congestion level, and its associated cost, as well as the number of hours lost in the future. In 2011, there were an estimated 300 cars per 1000 people in Manchester which are a 3% increase since 2001 [4]. This means that the number of cars on the roads in the area has increased which has caused congestion to increase in this period as well, leading to lost hours as well as the cost to increase over time.

Existing solutions to alleviate traffic congestion include increasing road capacity, dynamic traffic lights control systems [5], and real-time vehicle navigation systems (VNS). Compared to the first two solutions, which often require large investment and testing to apply, the VNS (e.g. Google Maps, TomTom, AutoNavi, etc.) are now widely used for end users or drivers with multiple routing choices to avoid roads that are currently congested. However, most of these navigation systems can only provide the route either with the shortest travel distance, or with the fastest current travel speed. Other routing preferences are usually not considered, such as the least number of junctions, the easiness of driving, and the road safety level, which often leads to non-recurrent congestion. Moreover, these VNS are not open sourced so that they hinder the research communities to apply further improvements.



year	day	severity	casualties	Easting	Northing
2018	Monday	slight	1	384870	396736
2018	Wednesday	serious	1	384682	397008
2018	Sunday	slight	1	384544	397177
2018	Sunday	slight	1	384544	397178
2018	Sunday	serious	1	385288	396129
total	5				

List accidents in 2018 on:

Fig. 1. Number of accidents occurred on a road in Manchester in 2018

In this paper, we designed and implemented a personalized routing system that is compatible with both iOS and Android devices. End users can input their routing preferences for the shortest travel distance, or the fastest current travel speed, or the highest safety level (i.e. using the TfGM dataset with its sample shown in Figure 1). Users can also flexibly adjust the importance among these routing metrics according to their own needs. Additionally, we have tested this personalized routing application on a section of the Manchester city center road map, and collected users' feedback, which the majority of them are very positive showing our system is rather convenient to use.

II. Related Works

In this section, we will briefly present the two most used routing algorithms, review two advanced routing systems proposed for smart cities and outlines the main routing criteria that should be used to accommodate different drivers' needs.

A. Routing Algorithms

In many GPS navigation systems, such as sat-navs and navigation apps including Google Maps, Dijkstra and A* algorithms are usually used to plan routes [6]. Both algorithms have benefits and costs as described below.

Dijkstra's algorithm was created by Edsger Wybe Dijkstra in 1959 [7] and the purpose of it is to find the shortest path within a network. In this algorithm, a route is calculated by placing all vertices of a graph in a priority queue which is then ordered based on their proximity to the source vertex. The closest vertices to the source vertex are removed and the distance of neighbouring vertices is then updated. This process is continued until the shortest path from the source has been found.

The A* algorithm [8] was created in 1968 by Peter Hart, Nils Nilsson and Bertram Raphael. To find the most optimal solutions, it combines features of uniform-cost search and pure heuristic search. To calculate the route, the algorithm starts with two lists, a closed list to record areas evaluated and a fringe list to record areas adjacent to those already evaluated. The distances traveled from the start point and the estimated distance to the goal point are also stored throughout the algorithm running. A* traverses the graph, it follows the path with the lowest known cost. A priority queue to store alternative path segments is also produced. If any part of the path being traversed has a higher cost than an alternative in the priority queue, this latter will then be traversed instead. The process continues until the goal has been reached and the path found will be the route with the lowest possible cost.

Compared to Dijkstra, A* algorithm has better performance because it uses heuristics to produce the result by estimating the distance to the goal for points when calculating the shortest route. This means it restricts the search space, and in road networks such search space is restricted to the area where traffic congestion has changed. By contrast, Dijkstra's algorithm is more suited when all nodes in a graph need to be visited. This is because it is terminated as soon as the destination node is found, whereas, the A* algorithm will only find the optimal path when the full shortest path tree has been calculated. Therefore, if all nodes are required to be visited in the route then Dijkstra's is the best algorithm to use and A* is if not all nodes need to be visited.

B. Routing Systems

In a smart city, routing vehicles from origin to destination is rather a systematic approach than just a single algorithm. A smart city is a city that connects people, information and city elements using new technologies to create a sustainable, greener city, competitive and innovative commerce, and an increased life quality. The technologies that could be used in smart cities to reduce the effects of congestion include vehicle to everything (e.g. road side unit) communication technologies, enabled dynamic traffic light control system and real time VNS. Currently, VNS do not consider unpredictable route events (i.e. usually happens when road safety level is low) that can occur, such as accidents, on an already calculated route, meaning that such route may not remain the fastest route throughout the driver's journey. An example of a routing system that combines the above technologies to achieve efficient routing is the so-called Next Road Rerouting (NRR) [9]. NRR's main goal is to find the optimal next roads to follow by the vehicles affected by a road event to bypass the blocked road segment due to this event. Once the updated route is found the concerned drivers are informed by updating the route displayed by the VNS so that the new route will be followed instead. As NRR is only calculating a new part of the route and not the entire route, it is fast and more efficient at producing the updated route (i.e. re-routing path), which is an essential feature that prevents the creation of bottleneck around the en-route event location.

Re-routing with FOg-CloUd System (ReFOCUS+) [10] is another advanced Route Guidance System (RGS). ReFOCUS+ employs Road Side Units (RSUs) to calculate traffic factors such as current and predicted congestion and travel time. Using this data, it applies re-routing to vehicles to try and reduce traffic congestion. The re-routing applied uses a multi-metric function called Road Weight Measurement, which means that many factors such as travel time, emissions and distance are included in the calculations made. NRR and ReFOCUS+ advanced routing systems were both simulated using the Simulation of Urban Mobility (SUMO) traffic simulator, and their performance were compared against other state of the art solutions. For both, they were more efficient than the existing systems they were compared to in terms of travel time. In addition, the ReFOCUS+ system was more efficient than the existing systems in fuel consumption as well. However, all these routing systems proposed for smart cities are heavily reliant on the infrastructure upgrade (e.g. RSUs, v2x communications), thus it is not easy to implement these ideas in practice to benefit the road users.

C. Routing Criteria

Common routing criteria includes travel distance and estimated travel time. To calculate a route, the map of a road network is converted to a graph with nodes and weighted edges. The nodes represent road junctions while the edges represent the roads. In terms of congestion, the weights should account for the length of the road and the traffic density [11]. Due to this, roads that have the shortest distance to the destination will not be included in the route if their traffic density is so high that it makes an alternative edge have a lower weight.

As drivers take the fastest route only for 35% of journeys according to [12], this suggests that other factors are influencing what route drivers should follow. One of these factors could be avoiding difficult junctions and motorways as there is a wider range of drivers now on the roads compared to the situation in previous years. As of 2014, 9% of licence holders were aged between 17 and 24 which are classed as young, inexperienced drivers [13]. As these drivers are less experienced, they may wish to avoid motorways and difficult junctions meaning that they do not always take the fastest route.

Another group of drivers who may choose to take the safest, instead of the fastest, route is older drivers. In 2016, approximately 12% of licence holders were aged 70 or over and this number has increased by 14% between 2013 and 2016 [2]. As this equates to 21% of all licence holders, this should be included as a metric in routing as this could potentially prevent accidents from occurring if drivers can calculate the safest route possible instead of just the quickest and shortest ones. Preventing accidents could also prevent non-recurrent congestion occurring which is another reason why this metric should be included in routing. This paper later presents how the road safety level is incorporated with our personalized routing application.

III. System Overview

This section describes our personalized routing system from developers' view and from the users' perspective. In the former, the different components of the system are introduced, while in the latter view, a full journey to use our application for planning a route is presented.

A. Developer's perspective

This system includes a user interface, a personalized routing algorithm, and a database. As shown in Figure 2, the user interface accepts users' input and pass them to the routing algorithm. Based on this input, the algorithm retrieves the corresponding map, traffic data, and events information from the database, computes the best fitted route, and returns the results to user interface to display them to the end users.

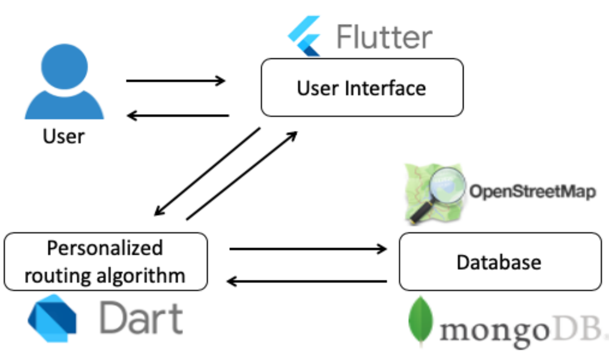


Fig. 2. System overview

The user interface of our application is configured to accept the users' input about their origin and destination locations, as well as their routing preferences (i.e. travel distance, travel time, and safety level). In addition, it also displays the routing results on the map. The user interface, as well the customized routing algorithm described later, are all developed using Flutter[1] framework and Dart programming language based on this framework. Flutter is chosen because the application developed using it can be well compatible with many commonly used platforms (i.e. mobile, web, and desktop).

We customize A* algorithm to allow multi-criteria routing, which preferences are personalized by different users. We consider three factors of a road in our routing algorithm: travel distance R_d, estimated travel time R_t, and accidents ratio R_a (i.e. safety level). R_d is calculated using Haversine formula (i.e. distance on the sphere surface rather than a plane) between its two ends (junctions). R_t is computed as the R_d over the average travel speed (data source: Transport for Greater Manchester [2]) during a certain time period.

R_a is calculated as the number of accidents occurred on a given road in the last 12 months (data source: TfGM [14]), divided by the maximum number of accidents happened on a single road in the selected area. The lower the value of R_d, R_t, and R_a the better. We aggregate these three factors as a linear combination to come up with a single weight value for each road W:

$$W = w_d \times R_d + w_t \times R_t + w_a \times R_a \quad (1)$$

where w_d, w_t, and w_a are the weight values for each of the three above mentioned factors, and the sum of them equals to 1. Note that R_d and R_t in this equation are also normalized in the range between 0 and 1 by dividing them by the maximum value measured on a road in the selected area.

The map data of the selected area in the city of Manchester (shown in Figure 3) is exported from the widely used open data source: OpenStreetMap. MongoDB is selected to be the database of our system as it is mainly based on the NoSQL technology which is normally considered faster than the traditional SQL technology.

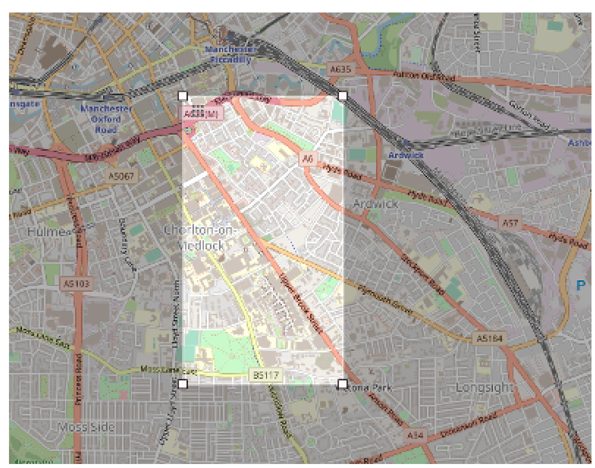


Fig. 3. The selected area in the city of Manchester, UK. (source: OpenStreetMap)

[1]https://flutter.dev/
[2]https://tfgm.com/

B. Users' perspective

When the end users plan a trip route using our system, the first step they need to do is to input the origin and destination locations from the given map. Here, the map is displayed using Google Maps api, however for the processing logic internally, the map data is from OpenStreetMap. The origin location of the users' trip is often their current location, but sometimes it could be anywhere according to their needs.

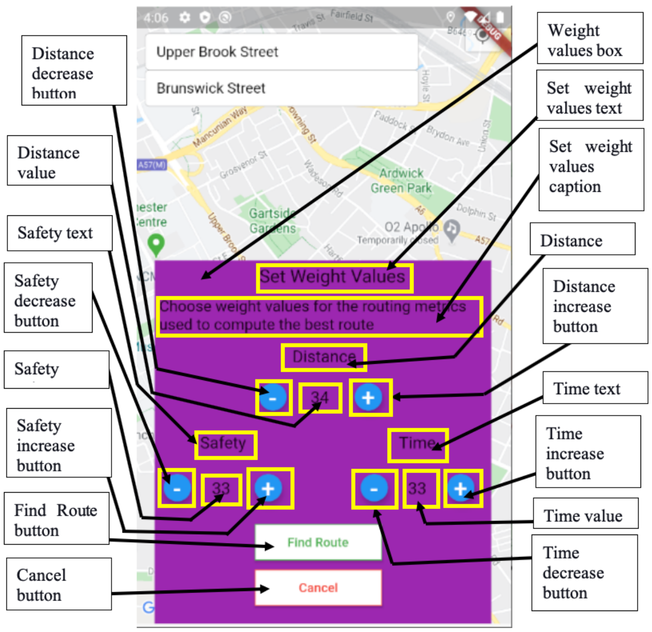


Fig. 4. User inputs the route preference by setting the weight value for each routing criterion

The second step for user input, which is also the main feature of our system, as shown in Figure 4, is to adjust route preferences. Specifically, users can adjust how important they think each of the three routing criteria is for their desired route. The number shown in Figure 4 indicates the percentage of each criterion as to 100%.

After the user hits the button *Find Route*, our system calculates the route accordingly and displays the results in route shapes, travel distance, and travel time, as shown in Figure 5. Although safety level is not shown, it is considered in the routing algorithm internally.

IV. Evaluation

This section describes the methodology used to evaluate the effectiveness of our proposed system and its corresponding results and analysis. In short, for our personalized routing system, this section will answer the two following questions:

1) Effectiveness: Is our customized routing algorithm reflects correctly various routing preferences?
2) Usability: Is our personalized routing application user-friendly enough?

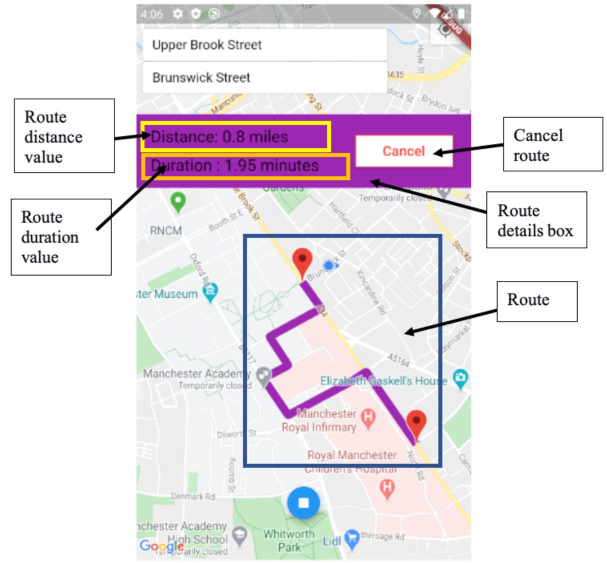


Fig. 5. Suggested route with its performance displayed to the end users

A. Effectiveness

To validate if our routing algorithm really reflects users' various routing preferences, we have tested our system under three origin/destination (O/D) pairs in the city of Manchester, UK. They are from *Upper Brook Street* to *Brunswick Street*, from *Upper Brook Street* to *Downing Street*, and from *Plymouth Grove* to *Downing Street*. For the sake of simplicity, we only present the last one of them. Similar conclusions can be drawn from other two O/D pairs.

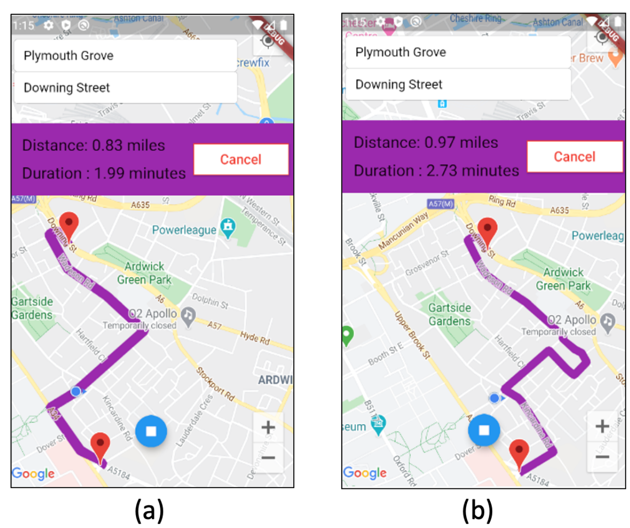


Fig. 6. Route calculated between Plymouth Grove and Downing Street (a: weight values set to 34 (distance), 33 (time) and 33 (safety); b: weight values set to 68 (distance), 32 (for both time and safety)

The route found is then displayed in Figure 6, where we have varied the three metrics weight values to assess their impact on the length and duration of the chosen route. As the routes shown in Figure 6 (a) and Figure 6 (b) are different, this connotes that the change in the metric weight values impact the routes computed and therefore, the routing algorithm works as expected.

Figures 7 and 8 indicate that the route with the shortest distance is not also the route with the shortest duration. This is because Figure 7 exhibits the route with the shortest distance is when the distance metric weight was set to 86 and safety to 14, while Figure 8 reveals the route with the shortest duration is when the time metric weight is set to 100. This is as expected because when the metric weight is set to 100 for time, the quickest route should be found. Therefore, the difference in the shortest and quickest routes computed as well as the different routes calculated indicate that the metric weights are being used correctly in the algorithm calculations. There was a 33% increase in the distance of the longest route compared to the shortest in terms of distance, and a 71% increase in the duration of the route, when the slowest route was compared to the duration of the quickest route calculated.

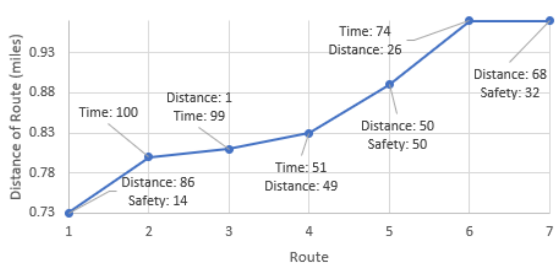


Fig. 7. Comparing the distance of the routes calculated and the metric weight values set for the route between Plymouth Grove and Downing Street

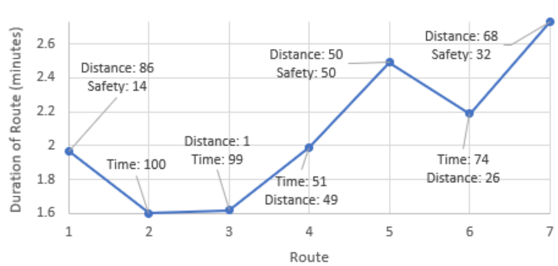


Fig. 8. Comparing the duration of the routes calculated and the metric weight values set for the route between Plymouth Grove and Downing Street

From the tests completed, it is evident that the algorithm is successfully computing routes and using the metric weights entered as expected. This is because the duration and distance of the routes changed when the metric weight entered changed for when the start and end point remained the same. On average, between the highest and lowest values calculated for routes, there was a 48% increase in the distance and 100% increase in the duration. Also, the tests confirmed that the routes computed were being output on the map as expected and would change if a different route had been found due to the metric weights entered.

B. Usability

The usability is very important for our personalized routing system. To assess the usability of the system, five testers (wide range in age and gender) have been selected to test the system and answer a questionnaire about their experience. Please note that due to COVID-19 lockdown measures we could not have larger number of testers. The questions included are based on the System Usability Scale (SUS) [15] with each of the ten questions (shown in Table I) the user being able to respond with one of these five answers- 'Strongly Agree', 'Agree', 'Neither Agree nor Disagree', 'Disagree' or 'Strongly Disagree' with the score value ranges from 5 (strongly agree) to 1 (strongly disagree). The final score of answers to all ten questions is calculated using the steps described as follows: note that a score of 68 is the average.

1) Add all the scores for the odd numbered questions (1, 3, 5, 7, 9) together and subtract 5 from this;
2) Add all the scores for the even numbered questions together (2, 4, 6, 8, 10) and subtract this number from 25;
3) Add the scores calculated in steps 1 and 2 together and multiply by 2.5;
4) Round the number calculated in step 3 to nearest whole number to get the final SUS score.

TABLE I
LIST OF QUESTIONS

Question 1	I would like to use the system frequently.
Question 2	The system was unnecessary complex.
Question 3	The system is easy to use.
Question 4	The support of a technical person was needed to use the system.
Question 5	The various functions of the system were well integrated.
Question 6	There was too much inconsistency in the system.
Question 7	People would learn to use the system very quickly.
Question 8	The system was very cumbersome to use.
Question 9	I am very confident using the system.
Question 10	I needed to learn a lot of things before I could get on with the system.

TABLE II
SUS SCORES FOR ALL FIVE TESTERS

	Tester 1	Tester 2	Tester 3	Tester 4	Tester 5
Question 1	4	3	2	4	1
Question 2	3	3	4	1	3
Question 3	4	4	3	4	2
Question 4	2	3	4	1	5
Question 5	4	4	4	4	4
Question 6	2	3	1	2	2
Question 7	3	4	3	3	3
Question 8	2	2	4	2	4
Question 9	4	3	2	5	1
Question 10	1	4	5	2	5
Score	**73**	**58**	**40**	**80**	**30**

From the SUS scores present in Table II, an average score of 56 has been calculated. As this is below the average of 68, this would indicate that the usability of our app could be improved. However, there was a vast range of scores between the testers, with the highest being 80 and the lowest being 40. This difference affects the average calculated and it is also likely the range of testers used effected it as well. The testers both below the age of 30 had SUS scores of above the average of 68, whereas the three testers over the age of 50 had SUS scores of below 68. This connotes that the system is potentially more suited to younger age groups, who in general use technology more than older age groups.

V. Conclusions and Future Work

In this paper, we have proposed, implemented, and tested a new personalized routing application that can provide route choice between two locations, according to each driver's preferences on travel distance, travel time, and the safety level. This latter metric is very important as the occurrence of a random incident on a road leads to non-recurrent traffic congestion that is hard to avoid. We also presented key validation results of this system based on the maps of the city of Manchester in England. As a future work, we plan to include more routing criteria, such as fuel consumption level, and carrying tests using maps of more cities with different road layouts, as well as improving the usability of our app.

References

[1] J. McGroarty, "Recurring and non-recurring congestion: Causes, impacts, and solutions," *Neihoff Urban Studio–W10, University of Cincinnati*, 2010.

[2] "Road traffic forecasts 2018 - moving britain ahead," Department for Transport, UK, Tech. Rep., July 2018.

[3] https://inrix.com/press-releases/scorecard-2018-uk/, [Online; accessed 25-June-2020].

[4] "Car ownership rates per local authority in england and wales," The RAC Foundation, Tech. Rep., December 2012.

[5] D. R. ALEKO and S. Djahel, "An efficient adaptive traffic light control system for urban road traffic congestion reduction in smart cities," *Information*, vol. 11, no. 2, 2020.

[6] I.-C. Chang, H.-T. Tai, F.-H. Yeh, D.-L. Hsieh, and S.-H. Chang, "A vanet-based a* route planning algorithm for travelling time- and energy-efficient gps navigation app," *International Journal of Distributed Sensor Networks*, vol. 9, no. 7, p. 794521, 2013.

[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec 1959.

[8] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[9] S. Wang, S. Djahel, Z. Zhang, and J. McManis, "Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2888–2899, 2016.

[10] M. Rezaei, H. Noori, M. Mohammadkhani Razlighi, and M. Nickray, "Refocus+: Multi-layers real-time intelligent route guidance system with congestion detection and avoidance," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2019.

[11] M. Farhan, "Traffic routing algorithm for road network," *Scientific Bulletin*, vol. 24, no. 2, pp. 131 – 138, 2019.

[12] J. Letchner, J. Krumm, and E. Horvitz, "Trip router with individualized preferences (trip): Incorporating personalization into route planning," in *Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence - Volume 2*. AAAI Press, 2006, p. 1795–1800.

[13] "Facts on young drivers," Department for Transport, UK, Tech. Rep., 2014.

[14] https://www.gmtu.gov.uk/gmaccidents/street_search.aspx, [Online; accessed 10-April-2020].

[15] J. Brooke, "Sus: a "quick and dirty'usability," *Usability evaluation in industry*, p. 189, 1996.