Streaming Route Assignment for Connected Autonomous Vehicles (Systems Paper)

Sadegh Motallebi, Hairuo Xie, Egemen Tanin, Jianzhong Qi, and Kotagiri Ramamohanarao
s.motallebi@student.unimelb.edu.au,xieh,etanin,jianzhong.qi,kotagiri@unimelb.edu.au
University of Melbourne
Australia

Abstract

In the coming era of connected autonomous vehicles, data-driven traffic optimization will reach its full potential. By collecting highly detailed real-time traffic data from sensors and vehicles, a traffic management system will have the full view of the entire road network, allowing it to plan traffic in a virtual world that replicates the real road network. This will bring significant innovations to transport-domain applications. We prototype a traffic management system that can perform traffic optimization with connected autonomous vehicles. We propose two route assignment algorithms that aim to reduce traffic delays by reducing intersecting routes. The proposed algorithms and two state-of-the-art route assignment algorithms are implemented in the prototype system. We evaluate the algorithms with both synthetic and real road networks. The experimental results show that the proposed algorithms outperform competitors in terms of the travel times of the routes.

CCS Concepts

• **Information systems → Geographic information systems**.

Keywords

Traffic Management Systems, Streaming Traffic Data, Route Assignment, Autonomous Vehicles

ACM Reference Format:

Sadegh Motallebi, Hairuo Xie, Egemen Tanin, Jianzhong Qi, and Kotagiri Ramamohanarao. 2019. Streaming Route Assignment for Connected Autonomous Vehicles (Systems Paper). In *27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19), November 5–8, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3347146.3359062

1 Introduction

It is estimated that a majority of the vehicles will be *connected autonomous vehicles (CAVs)* in the future [10]. The current generation of CAVs are designed to imitate the driving behaviour of human drivers. The vehicles work as independent units just like human-driven vehicles. In our view, the potential of CAVs will be better realized when all the vehicles are coordinated for maximizing system-wide traffic efficiency [6]. A highly coordinated traffic system may soon become a reality given the rapid development of CAVs [8]. In such a system, a CAV constantly reports traffic information to a *traffic management system (TMS)*. The information includes not only the status of the vehicle itself, such as its position and speed, but also the details about its surrounding environment, such as a road accident near the vehicle. By collecting traffic data from CAVs, the TMS can get a full view of the entire road network, allowing it to perform system-wide traffic planning in a virtual world. For example, when a TMS predicts traffic congestions caused by an unusual surge of traffic demand, it can plan detours for vehicles such that they can avoid the predicted congestions without causing new congestions in other areas. This work, the first of its kind, focuses on traffic optimization through route allocation based on the aforementioned traffic management scheme.

We propose two route assignment algorithms to mitigate traffic congestions for metropolitan areas, where traffic congestions are at their worst [9]. The algorithms tackle a key contributor to traffic congestions: the concentration of intersecting routes at road junctions [2, 11]. Intersection of routes can lead to long waiting times of the vehicles at junctions. Although innovative junction designs can help to mitigate congestions caused by intersecting routes [1], they require significant costs to implement and may lack the flexibility to adapt to sudden changes in traffic patterns. It may also be infeasible to alter the junctions due to spatial constraints or heritage conservation. Figure 1 illustrates the effects of an ideal route assignment algorithm. The left sub-figure shows three sub-optimal routes with two route-intersections. The right sub-figure shows the altered routes, where there is no intersecting route and the total travel time of all the vehicles can be reduced.

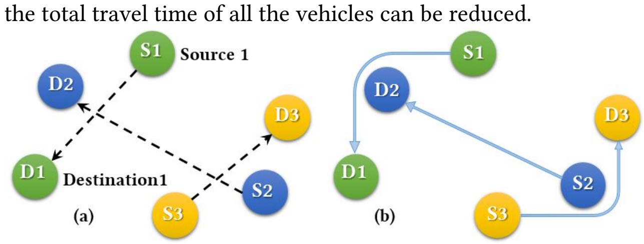

Figure 1: (a) Intersecting routes between three pairs of source-destination, S1-D1, S2-D2 and S3-D3. (b) Routes between the same source-destination pairs without intersections. Routes are shown on Euclidean space for the sake of simplicities.

Due to the NP-hardness of traffic optimization problems, our algorithms use heuristics for efficient computation. Both of them are based on the A* algorithm [3], which considers a heuristic function in addition to the minimal travel cost during the search for shortest paths. The algorithms differ in the scope of traffic information used for searching the routes. The first algorithm, *Local Detour Algorithm (LDA)*, uses traffic information that is confined to specific road links or road junctions, which are explicitly explored during route allocation. The second algorithm, *Multiple Intersection Reduction*

SIGSPATIAL '19, November 5–8, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-6909-1/19/11.
https://doi.org/10.1145/3347146.3359062

Algorithm (MIRA), enlarges the scope of traffic information by using a global view of traffic conditions. A heatmap is constructed based on the travel times on road links. With this heatmap, new routes can bypass entire city blocks that are affected by intersecting routes.

To the best of our knowledge, the most comparable algorithm is Self-Adaptive Interactive Navigation Tool (SAINT) [4]. For a source-destination pair, SAINT creates a number of candidate routes to minimize the increase of congestion based on the current traffic conditions. The algorithm tries to assign different candidate routes to different vehicles with the same source-destination pair. Our experiments compare the proposed algorithms with SAINT using microscopic traffic simulations. The results show that MIRA is the best approach for mitigating traffic congestions.

2 Streaming Route Assignment Algorithms

We propose two route assignment algorithms that work with streaming traffic data. The algorithms can be used in a traffic management system that receives navigation requests from users and assigns routes to the users. We assume that the users will follow the allocated routes. Our algorithms can work with three types of streaming data. The first is the source and the destination of vehicle trips. A vehicle submits the information when it is ready to start a trip. Based on the source-destination pairs, the algorithms generate routes. A newly-generated route will be stored as it may affect the generation of routes in the future. The second type of streaming data is the travel times on individual road links. When a vehicle passes through a road link, its travel time spent on the link is reported to the system. The average travel times on road links are updated periodically so they can show the most recent traffic flow conditions. The third type of streaming data is the updated routes. When a vehicle reaches the end of a road link, the vehicle's route is shortened as the link is removed from the route. The traffic management system considers the shortened route for generating routes in the future because there can be less intersecting routes and lower traffic load due to the change of the route.

Both algorithms use a **reservation graph** that consists of a set of vertices and a set of edges. The graph is the same as the road network graph except that it keeps a **reservation count** at each edge of the road network. The count keeps the number of routes passing through the edge. When a new route is created, all the reservation counts on the route increase by one. The count on an edge decreases by one when a vehicle has passed through the edge. LDA uses the graph to check the existence of intersecting routes. MIRA uses the graph to predict the traffic load at road links. We conjecture that knowing exact time plays no role when counting intersecting routes and predicting traffic load at road links based on our experience.

2.1 Local Detour Algorithm (LDA)

The intuition behind LDA is that the travel time of a vehicle can be reduced if the vehicle spends less time waiting for conflicting traffic at road junctions. LDA is based on the A* algorithm [3]. In order to detour junctions with intersecting routes, LDA computes a delay-value as the heuristic function. The delay-value is defined in Equation 1, where α is an *intersection factor* and $MATT$ is the maximum average travel time on conflicting edges.

$$Delay = \alpha MATT \tag{1}$$

To compute the $MATT$ of an edge $e_{m,n}$, which starts from vertex m and ends in vertex n, LDA first uses the reservation graph to find other edges that end in n but conflict with $e_{m,n}$. The algorithm then compares the average travel times on the conflicting edges, which have a positive reservation count. $MATT$ is set to the maximum average travel time.

A longer average travel time on a road link generally implies that the traffic load at the link is higher. With an increase of the traffic load on conflicting edges at a road junction, the number of conflicting routes increases. As a result, a vehicle tends to spend a longer time waiting at the road junction. This is the reason that the delay-value is proportional to the maximum average travel time $MATT$. By decreasing the intersection factor, the delay-value is decreased, which means the travel cost between two adjacent vertices in the road network graph is lower. Consequently, the search can be expanded to more vertices. However, there can be a higher chance that a returned route intersects with existing routes. On the contrary, increasing the intersection factor can help to avoid more intersecting routes. Based on our early tests, we set the value of LDA's α parameter to 0.5 in the experiments.

2.2 Multiple Intersection Reduction Algorithm (MIRA)

The intuition behind MIRA is that the travel time of a route can be reduced more effectively by avoiding entire city blocks that are affected by intersecting routes. Existing route assignment algorithms, such as SAINT and other diversification-based approaches, do not attempt to make long detours at the city level. They focus on not assigning traffic to the same route. The routes given by these algorithms generally go in one direction, from source to destination, with minor deviations from the shortest paths. MIRA, on the other hand, enables a higher level of flexibility of choosing travel directions on the route. This is achieved based on two data structures. The first is a heatmap that shows the normalized average travel times on road links in different city blocks, each of which covers a rectangular area of a city. As the concentration of vehicles (and the intersections of their routes) in a city block generally leads to a relatively high average travel time in the block, the difference between adjacent heatmap values shows the direction of traffic flows at the global level. The heatmap is periodically updated based on the latest traffic information sent from CAVs. The second data structure is the reservation graph, which helps to show the direction of traffic at individual road junctions. The reservation counts in the graph are updated in the same way as in LDA. Similar to LDA, MIRA is also based on A* algorithm but the heuristic function for an edge is defined as the product of two values. One is the value of the heatmap cell that covers the edge. Another is the reservation count at the edge. This not only helps vehicles to avoid a large number of intersecting routes altogether by detouring around city blocks affected by those routes, but also helps vehicles to avoid individual road junctions that are affected by intersecting routes within a block.

The heatmap is constructed by mapping the whole road network space into a grid. For each cell of the grid, the average travel time on road links is computed. As we focus on mitigating congestions in metropolitan areas, we assume that the road links in different grid cells are homogeneous, which means the road links have similar length, capacity and speed limit. The average travel time of each cell is normalized by dividing the average travel time of the cell by the total value of all the cells. The normalized values are filled into the heatmap.

2.3 Super-MIRA (sMIRA)

There can be many variations of MIRA based on the heatmap. We describe one variation, called super-MIRA (sMIRA). Compared to MIRA, the global traffic flow directions play a more important role with sMIRA. The algorithm increments the reservation count of an edge on a newly allocated route by the edge's corresponding heatmap value, rather than a fixed value, 1, as in MIRA. The heatmap value is deducted from the reservation count once the vehicle with the route passes through the edge. Our experiments show that sMIRA can work even better than MIRA in a situation, where a majority of the vehicles concentrate to the centre of an area while other vehicles move across the area.

3 Experiments

Our main experiments compare the proposed algorithms, LDA and MIRA, against two baseline algorithms in terms of the quality of the allocated routes. The first baseline algorithm is called *First-In-First-Assigned Fastest (FIFA-Fastest)*. This algorithm computes routes based on the current travel times using Dijkstra's algorithm. The second baseline algorithm is SAINT [4], which is described in Section 1. We also compares sMIRA with MIRA.

We build a prototype traffic management system that consists of two components, a route allocator and a traffic simulator. The route allocator receives navigation queries from the simulator and computes routes based on the queries. The routes are given to the traffic simulator, which simulates vehicles based on the routes and outputs the travel times of the vehicles. The simulator used in our experiments is SMARTS [7]. The simulator performs realistic simulations based on microscopic traffic models. It also simulates adaptive traffic lights as in the real world, which adjust traffic signal timing in real time based on dynamic traffic flow conditions. SMARTS has been used intensively in our research work, such as a route allocation algorithm based on traffic diversification [5].

We measure the performance of the algorithms using two metrics, *Travel Time Ratio at Individual level (TTRI)* and *Travel Time Ratio at System level (TTRS)*, where $BTT(v_i)$ is the best theoretical travel time for a vehicle v_i and $TT(v_i)$ is the actual travel time of the vehicle (Equation 2). The best theoretical travel time is calculated under the assumption that the road network is empty and the vehicle never encounters a red light at any junction. Therefore it is always equal to or smaller than the actual travel time. Consequently, TTRI is equal to or higher than 1. Different to TTRI, TTRS is based on the total travel time of all the vehicles. TTRS is equal to or higher than 1.

$$TTRI = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\frac{TT(v_i)}{BTT(v_i)} \quad \text{and} \quad TTRS = \frac{\sum_{i=1}^{|\mathcal{V}|} TT(v_i)}{\sum_{i=1}^{|\mathcal{V}|} BTT(v_i)} \tag{2}$$

We note that there is no meaningful TTRI and TTRS if there are gridlocks in the simulated traffic system. A gridlock appears once an area is so severely congested that no vehicle can make further movement. We observe that gridlocks will eventually appear when the number of vehicles is beyond a certain value, no matter which algorithm is used for routing. We call the maximum number of vehicles that exist in a network without a gridlock the **gridlock threshold**. Different algorithms can have different gridlock thresholds. A higher gridlock threshold is better as the traffic network can function with a larger number of vehicles. Our results do not show TTRI and TTRS when the number of vehicles is beyond the gridlock threshold.

3.1 Experimental Settings

We evaluate the algorithms based on a synthetic road network and a real road network. The synthetic network is based on a grid plan with two sets of streets. Each set has 12 streets that are parallel to each other. A street in one set runs at right angle to a street in another set. Traffic lights are installed at all road junctions. Adjacent junctions are connected by a two-way road segment that is 400 metres long. The speed limit for all the road segments is set to $40km/h$, which is the common speed limit for many central business areas across Australia. The default granularity of the heatmap for MIRA is 3×3 as the area can be fully divided at this granularity. For this road network, we evaluate the effects of two parameters on all algorithms. For each parameter, we run simulations with different values of the parameter while keeping the other parameter at its default value. The first parameter is the **number of vehicles**. This parameter can have a significant impact on travel times. As the number of vehicles increases, the number of intersecting routes increases, leading to a higher chance of traffic congestions. A good route assignment algorithm can suggest routes that lead to satisfactory travel times even when there are a large number of vehicles. We vary the value of this parameter between 1000 and 10000. The default value is 6000 because we cannot run SAINT with 7000 or more vehicles. The second parameter is the **spatial distribution of source and destination**, which is either uniform distribution or Gaussian distribution. If a source or a destination is generated with the uniform distribution, we randomly pick a point from the road network as the source or the destination. With Gaussian distribution, the sources and the destinations are more likely to be located around the centre of the road network area. Gaussian distribution is used as the default value as it is more realistic than the uniform distribution for the central area of a city.

The real road network covers a $30km\times30km$ area centred at the CBD of Melbourne. The speed limit of road segments, the number of lanes on road links and the direction of road links are extracted from OpenStreetMap (https://www.openstreetmap.org). This network has 20400 vertices and 25600 edges. To minimize the impact of variance in road links, this road network only contains freeways and arterial roads in the area. For this network, we vary the number of vehicles between 10000 and 50000. We use Gaussian distribution for generating sources and destinations. The granularity of MIRA's heatmap is set to a relatively low value, 3×3, because the density of roads is not high as we only use arterial roads. We do not include SAINT in this experiment because the algorithm cannot return routes in a manageable time based on this network.

3.2 Results

Our result with the synthetic network shows that MIRA outperforms other algorithms except when the number of vehicles is very low (Figure 2). MIRA is the best algorithm for avoiding gridlocks.

The gridlock threshold of MIRA is 9600 while the gridlock threshold of the second-best algorithm, SAINT, is 6800. As shown in Figure 3, all algorithms perform well when the sources and destinations follow a uniform distribution but MIRA has a small advantage over other algorithms in this situation. FIFA-Fastest and LDA perform poorly with Gaussian distribution as they lead to gridlocks when the number of vehicles are at the default value, 6000. SAINT and MIRA are the only two algorithms that still work in this situation. MIRA achieves a lower TTRI (3.29) than SAINT (3.92). The result also shows that TTRI increases significantly for both algorithms when the distribution changes from uniform to Gaussian. This is understandable as there is a higher probability of route intersections when the sources and destinations become clustered. Overall, MIRA achieves the best performance with both distributions.

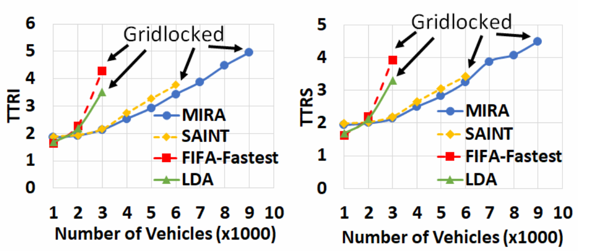

Figure 2: TTRI and TTRS achieved with the synthetic road network. Lower values are better.

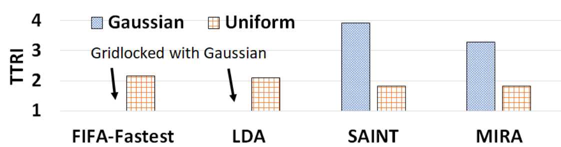

Figure 3: TTRI achieved with the synthetic road network using two source-destination distributions. Lower values are better.

For the real network, MIRA achieves the best performance among all the algorithms (Figure 4). The gridlock threshold of both LDA and FIFA-Fastest is lower than that of MIRA by 10000. This shows that the global view of traffic conditions used by MIRA helps to reduce intersecting routes significantly. The gap of TTRI between MIRA and other two algorithms becomes higher when there are more vehicles in the network. For example, the gap between MIRA and LDA increases from 0.136 to 1.467 when the number of vehicles increases from 10000 to 30000. We observe a similar trend in TTRS. Compared to small road networks, such as the synthetic network shown earlier, large road networks allow vehicles to make relatively longer detours in exchange for better travel times. MIRA can take this advantage by using the heatmap while other algorithms cannot.

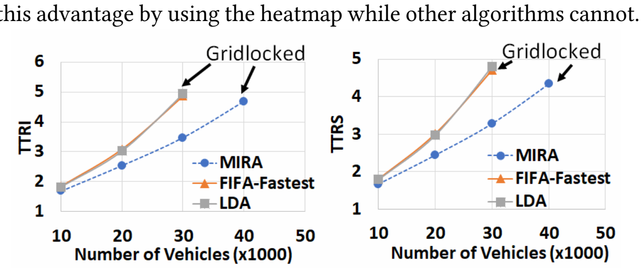

Figure 4: TTRI and TTRS achieved with the real road network. Lower values are better.

We compare MIRA and its variation, sMIRA, in a more complex traffic scenario with the synthetic network, where the source-destination pair of 80% of the vehicles follows Gaussian distribution while the remaining vehicles start and end their trips at random locations on the border of the area. The heatmap resolution is set to 6×6 as it works best for both algorithms in this scenario. Our result shows that sMIRA's gridlock threshold is higher than that of MIRA by 2000. sMIRA also achieves a lower TTRI and a lower TTRS than MIRA.

We also run a test, where a portion of the routes are computed with FIFA-Fastest while the remaining routes are given by MIRA or SAINT. MIRA performs better than SAINT as it achieves a lower TTRI when the ratio of FIFA-Fastest routes is between 0% and 20%. SAINT leads to a gridlock when the ratio is higher than 20% but MIRA can still work until the ratio reaches 40%.

Finally, we run an experiment to evaluate the impact of query batch size on the TTRI achieved by MIRA. A larger batch size means the system needs to wait for more queries until a batch is fully filled. All the queries in a batch are processed together. We vary the batch size between 1 and 50. Our result shows that the impact of the batch size on the quality of the routes is negligible. That means the stream window size can be set to a small value, e.g., a few seconds, which ensures a fast response time of an application.

4 Conclusions

Our work shows that MIRA is a ready-deployable algorithm for system-wide traffic optimization by minimizing intersecting routes. Compared to a state-of-the-art algorithm, SAINT, MIRA can handle more vehicles without causing gridlocks. The travel times of vehicles are lower with MIRA than with other algorithms. It is also easy to create variations of MIRA for specific traffic scenarios. We hope more traffic optimization solutions can be inspired by this work for the era of connected autonomous vehicles.

References

[1] J. Autey, T. Sayed, and M. El Esawey. Operational performance comparison of four unconventional intersection designs using micro-simulation. *Journal of Advanced Transportation*, 47(5):536–552, 2013.

[2] I. Dadić, G. Kos, K. Poić, and E. Gašparac. Measures for reducing traffic congestion in cities. *PROMET-Traffic & Transportation*, 11(1):15–19, 1999.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[4] J. Jeong, H. Jeong, E. Lee, T. Oh, and D. H. C. Du. SAINT: Self-Adaptive Interactive Navigation Tool for cloud-based vehicular traffic optimization. *IEEE Transactions on Vehicular Technology*, 65(6):4053–4067, 2016.

[5] U. T. Nguyen, S. Karunasekera, L. Kulik, E. Tanin, R. Zhang, H. Zhang, H. Xie, and K. Ramamohanarao. A randomized path routing algorithm for decentralized route allocation in transportation networks. In *SIGSPATIAL*, pages 15–20, 2015.

[6] K. Ramamohanarao, J. Qi, E. Tanin, and S. Motallebi. From how to where: Traffic optimization in the era of automated vehicles. In *SIGSPATIAL*, pages 10:1–10:4, 2017.

[7] K. Ramamohanarao, H. Xie, L. Kulik, S. Karunasekera, E. Tanin, R. Zhang, and E. B. Khunayn. SMARTS: Scalable Microscopic Adaptive Road Traffic Simulator. *ACM Transactions on Intelligent Systems and Technology*, 8(2):26:1–26:22, 2016.

[8] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2016.

[9] M. Terrill, H. Batrouney, S. Etherington, and H. Parsonage. *Stuck in traffic? Road congestion in Sydney and Melbourne.* Number 2017-10. Grattan Institute, 2017.

[10] Transport Systems Catapult. Market forecast for connected and autonomous vehicles. https://ts.catapult.org.uk/intelligent-mobility/im-resources/publications/, 2017.

[11] C. Wright, D. Jarrett, and G. Appa. Spatial aspects of traffic circulation: II. Routing patterns that exactly minimise path crossings. *Transportation Research Part B: Methodological*, 29(1):33–46, 1995.