Improving the Scalability of Automated Planning-based Vehicle Routing via Smart Routes Identification

Matyáš Švadlenka
Faculty of Electrical Engineering
Czech Technical University in Prague
Prague, Czech Republic
svadlmat@fel.cvut.cz

Lukáš Chrpa
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague
Prague, Czech Republic
chrpaluk@cvut.cz

Mauro Vallati
School of Computing and Engineering
University of Huddersfield
Huddersfield, United Kingdom
m.vallati@hud.ac.uk

***Abstract*—Due to growing urbanisation, traffic infrastructures have to accommodate increasing demands of traffic volume. One promising way for supporting a better exploitation of traffic networks is vehicle routing, that can distribute traffic from congested links to under utilised ones. Automated Planning techniques, a research field of Artificial Intelligence, have demonstrated to be a suitable approach for performing effective centralised traffic distribution. However, a main weakness of this class of approaches is the limited scalability to large and complex networks.**

In this paper, we aim to improve the scalability of automated planning techniques for urban traffic distribution by introducing an approach for the identification of routes to be considered. The proposed technique can significantly improve the planning capabilities by simplifying the complexity of the urban network to be considered, as demonstrated by our extensive experimental analysis performed on realistic traffic data on part of the New York and Sydney urban areas.

***Index Terms*—Traffic Routing, Artificial Intelligence, Automated Planning, Urban Traffic Control**

I. Introduction

Given the rapid rise in the number of people living in urban areas, that is expected to continue in the next decade, there is an increasing pressure on urban traffic networks. Currently, traffic control in urban areas is mostly performed via the optimisation of traffic signals, with the overall objective of minimising delays. Example of techniques commonly deployed in urban regions include SCOOT [1], MOVA [2], and SCATS [3]; they are adaptive approaches that can react to changing traffic conditions by rapidly adjust the behaviour of a set of connected junctions. With the advent of connected vehicles [4], and the growing interest in the smart city concept [5], traffic routing is gaining traction as an additional approach to support urban traffic controllers [6].

The key idea of traffic routing is to deal with unbalanced network usage. It is often the case that during rush hours most main roads are congested, at least in one direction, while many other surrounding roads are largely underused. This is usually the case also for large scale events, like sport matches or concerts. Such under-use of the network is caused by the fact that traffic is navigated via the same route between given way-points, and is often resulting from similar behaviour and habits of vehicle drivers or SATNAV systems. Considering the available network, it is generally the case that the exploitation of alternative routes can lead to a better distribution of vehicles and a better use of the capacity of the network. This can provide a suitable trade-off between overall network performance and an increased travelled distance for some vehicles.

Considering the Artificial Intelligence field, Automated Planning [7] has shown to be an effective tool for centralised vehicle routing to distribute traffic in an urban road network. The results presented in the recent works [8]–[10] have shown that planning-based approaches can considerably reduce the travel time of the vehicles passing through the controlled urban areas, hence improving the overall performance of the network. However, the main drawback of approaches based on planning, as well as of centralised approaches in general, is their lack of scalability that, in consequence, allows their effective use to small scale road networks. On the other hand, centralised approaches can leverage on an encompassing overview of the network, allowing better informed decisions.

In this paper, we propose an approach to address the scalability issue of planning-based traffic routing by efficiently identifying suitable routes to be considered for the routing of a vehicle. The idea follows the observation that automated

Lukáš Chrpa was funded by the OP VVV project no. EF15_003/0000470 “Robotics 4 Industry 4.0”. Mauro Vallati is supported by the UKRI Future Leaders Fellowship [grant number MR/T041196/1].

planning solvers are not effective for repetitive route finding for multiple vehicles, and can therefore benefit from the exploitation of a set of pre-identified high quality routes that vehicles can take. To guarantee the quality of the routes and, at the same time, ensure their diversity and suitability to distribute traffic, the proposed approach leverages on a variant of the well-known A* search algorithm [11] to generate routes with bounded suboptimality. Then, the DBSCAN [12] clustering algorithm is used to divide routes into clusters based on their similarity, which is measured by the Jaccard index. The final set of routes that are considered by the automated planning system includes one route from each cluster. Our experimental analysis, which considers two different road networks – one from New York and one from Sydney, evaluates the effectiveness and efficiency of the proposed method. The results clearly show that our approach can improve the scalability of the centralised planning approach by an order of magnitude, while maintaining the high quality of solutions in terms of minimising travel times and waiting times.

II. Automated Planning

Automated Planning is a prominent field of artificial intelligence that, in a nutshell, deals with the problem of finding a sequence of actions transforming the state of the environment from a given initial state to some goal state [7]. Differently from data-driven approaches, in automated planning a knowledge model is required. A knowledge model is a symbolic formal model representing the application domain, and a description of the actions that can be performed, of the object, initial states and goals to be achieved.

In the context of centralised intelligent vehicle routing, Automated Planning provides an effective decision-making tool as has been shown in recent works [8]–[10]. In this paper, we rely on the model introduced by Chrpa et al. [8] that is based on a microscopic traffic representation, i.e., represents each (relevant) individual vehicle. The model represents a road network as a directed graph, in which nodes represent junctions and edges road links. Each road link has its capacity, i.e., the maximum theoretical number of vehicles that can fit into the link (taking into account a minimum space between vehicles), and two thresholds that divides traffic intensity into three categories – *light, medium* and *heavy*. To draw a parallel between the categories and the well-known Level of Service, light intensity level corresponds to grades A and B, medium intensity level to C and D and heavy intensity level to E and F. The actions included in the model, that are under the control of the planning system and embody the idea of traffic routing, represent the movements of a vehicle between two junctions connected by a road link. The movement of vehicles on links with higher traffic intensity level is penalised by a corresponding higher cost of the action. The goal of the planning problem is to get all the vehicles from their initial position to their destinations while minimising total action cost, hence minimising traffic intensity levels for every road link. To achieve the goal, the system has to calculate, for each vehicle and for each junction in the network, the best direction to take: this particular aspect is wasting resources and reducing scalability, as the same reasoning has to be repeated multiple times. For details about the model the interested reader is referred to [8].

Algorithm 1: Our variant of A* for best k routes

```
Data: Starting junction id start, target junction id
      target, suboptimality bound c, max number of
      routes K
Result: The set of found routes
Routes ← ∅
queue, shortest_path ← A*(start, target)
if shortest_path is undefined then
    return ∅
end
Routes ← Routes ∪ {shortest_path}
limit ← c × length(shortest_path)
while queue not empty do
    priority, route ← pop(queue)
    if priority > limit or |Routes| == K then
        break
    else if destination(route) == target then
        Routes ← Routes ∪ { route}
        continue
    else
        foreach unvisited neighbour of route do
            route′ ← add(route, neighbour)
            priority ← length(route′) +
              distance(neighbour, target)
            push(queue, (priority, route′))
        end
    end
end
return Routes
```

While the approach introduced by Chrpa et al. [8] demonstrated promising results, it does not scale on large and complex urban road networks due to the highlighted need to repeat the same reasoning over time. In this paper we tackle such scalability issue by reducing the burden on the planning process by identifying, in a pre-processing step, suitable alternative routes to be considered for each origin-destination pair, hence reducing the computational cost of the task.

III. Smart Routes Identification

To smartly identify routes that can be considered for planning-based traffic distribution, we designed a three-step approach. First, k routes between each origin and destination are generated. Second, routes are clustered and, third, the most representative route from each cluster is selected and provided as input to the planning system. In the remainder of this section each step is detailed.

Algorithm 2: Selecting most appropriate routes

Data: Set of routes R, Minimum distance ϵ, Minimum cluster size q

Result: A set of diverse routes

```
forall r_i, r_j ∈ R do
    diversity[i][j] ← 1 − J(links(r_i), (r_j))
end
Clusters ← DBSCAN(ε, q, diversity)
selected ← ∅
forall R^c ∈ Clusters do
    selected.insert(Select(R^c))
end
return selected
```

A. Initial Routes Generation

The first step of the proposed approach aims at generating a candidate set of routes to be considered for each origin and destination pair. Traditionally, this has been addressed by using top-k path planning techniques. Examples of such techniques includes the well-known Yen's algorithm [13] or the K* algorithm [14]. These algorithms require to explicitly provide the value of k, i.e. the number of routes required. However, it might not always be possible to estimate a good value of k for a given road network and the locations of origin and destination.

We propose a variant of the well known A* algorithm, depicted in Algorithm 1, that does not require to specify the value of k a priori. Instead, it requires to specify the maximum number K of routes that can be generated, and a threshold value c of suboptimality of the provided routes for each origin and destination of the considered road network. The modified algorithm generates a set of routes R such that $|R| \leq K$ and, for each $r \in R$, it is the case that $length(r) \leq c * length(r^*)$ (where r^* is a route with minimum length). As a heuristic we use Euclidean distance from the current state to the destination location. Such a heuristic is admissible and hence guarantees optimality of the (first) solution generated by A*.

More in detail, the A* algorithm has been modified so that, instead of terminating after the optimal route from an origin to the considered destination (r^*) is found, it keeps expanding the nodes from the open queue and storing routes in R. The algorithm terminates when the value of the expanded node is greater that $c * length(r^*)$, hence no more routes that satisfy the suboptimality bound exist, or $|R| = K$ – hence the maximum number of allowed routes have been generated.

The underlying idea of the proposed A* modification is to identify a potentially large number of routes that sits within a provided bound from the optimal path, hence avoid routes that are unlikely to be used by vehicles navigating to the considered destination and that would probably be rejected by drivers or vehicle users.

B. Selecting Diverse Routes

Given the set of routes R generated by the introduced A* algorithm, it is usually the case that many routes share a large number of links – hence reducing their diversity while remaining within the c bound. A typical example can be drawn from grid-like networks, where 2 routes are different only because in one case the vehicle goes around a block instead of going straight. This high level of similarity can be a problem, since it may reduce the ability of the system to effectively distribute traffic.

To measure similarity between routes, we use the Jaccard index that measures similarity of two sets as a ratio between the number of elements the sets have in common and the number of the elements the union of the sets has. In particular, for sets A and B, the Jaccard index is defined as follows:

$$J(A,\ B) = \frac{|A\ \cap\ B|}{|A\ \cup\ B|}$$

It shall be noted that, for the Jaccard index, routes are considered not as sequences of links, but as sets. Therefore, it is not relevant when a given link is traversed, but only the fact that it is part of the route. The Jaccard index ranges between $[0, 1]$, where 0 represents the case in which A and B are fully disjoint while 1 represents the case in which A and B include exactly the same links. Jaccard distance is the complement of Jaccard index and is computed as $1 - J(A, B)$.

The Jaccard distance between routes is then considered to perform the clustering of generated routes. Algorithm 2 describes the procedure for dividing a set of routes into clusters and selecting the most appropriate route from each of the clusters. Clusters are generated by the DBSCAN algorithm [12] according to the Jaccard diversity, ϵ that refers to the minimum distance between routes from different clusters and q that refers to the minimum number of routes per cluster. Then, the most representative route from each cluster is selected. To identify the most representative route, we designed three alternative approaches: Shortest Route, Maximum Similarity Route, and Maximum Dissimilarity Route.

The Shortest Route is a straightforward way for selecting a route from a cluster, defined as follows.

$$\text{Select}(R^C) = \arg\min_{r \in R^C}(length(r)) \tag{1}$$

The Maximum Similarity Route stems from the idea of selecting as representative of a cluster the one that has the most in common with other routes from the same cluster.

$$\text{Select}(R^C) = \arg\max_{r \in R^C}(\sum_{r' \in R^C} J(r, r')) \tag{2}$$

Lastly, the Maximum Dissimilarity Route metric captures the idea of selecting the route from a cluster that differs the most from routes from other clusters.

$$\text{Select}(R^C) = \arg\max_{r \in R^C}(\sum_{r' \in \bigcup(\text{Clusters} \setminus \{R^C\})} (1 - J(r, r'))) \tag{3}$$

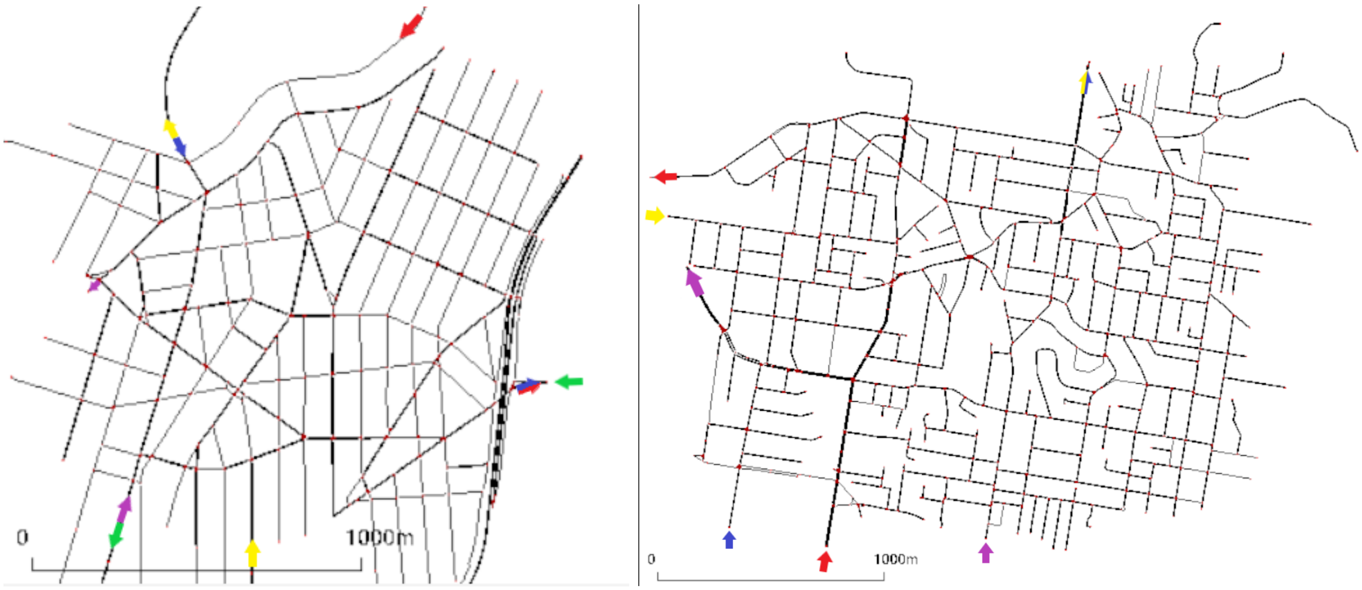

Fig. 1. Considered scenarios – a part of New York (left) and a part of Sydney (right). Arrows of different colours highlight corresponding entry and exit points we have considered.

TABLE I
SIZE OF THE SIMPLIFIED ROAD NETWORKS, IN TERMS OF THE NUMBER OF JUNCTIONS AND THE NUMBER OF ROAD LINKS

	New York		Sydney	
	Junctions	Links	Junctions	Links
Def	212	455	441	1061
KR	153	341	218	442
SR	130	236	184	267
MSR	145	283	194	300
MDR	138	249	208	335

C. Merging Subgraphs of Routes of Different Directions

A controlled urban traffic network usually includes multiple origin-destination couples. For each couple, a set of routes is identified by using the approaches described above. It is worth noting that the set of routes represents a subgraph of the original road network. In order to allow the planning system to perform traffic distribution, all the subgraphs are merged together into a more general traffic network graph, that is then provided to the planning system. The planning approach responsible of distributing traffic is then allowed to reason only on the provided merged graph, and ignores links or routes that are not part of it.

IV. EXPERIMENTAL RESULTS

The aim of this experimental analysis is to assess the impact of the proposed techniques for generating routes on planning-based vehicle routing in urban areas. On this regards, we are also interested in comparing the introduced metrics to select the most representative route from each cluster (Alg. 2), i.e., Shortest Route (SR) per cluster (see eq. (1)), Maximum Similarity Route (MSR) per cluster (see eq. (2)) and Maximum Dissimilarity Route (MDR) per cluster (see eq. (3)). As baselines, we consider the vanilla best k route approach (KR) generated by our modified A* algorithm with no subsequent filtering (Alg. 1), and the default approach (no simplification of the road network) (Def) used by Chrpa et al. [8] For reference, we also include a naive approach that always considers the shortest routes.

A. Scenarios

To test the ability of the proposed approach to generate effective routes for different urban networks, in this experimental analysis we consider two scenarios: one is an area of New York (NY), located between Grand Concourse and Sheridan Boulevard (Figure 1 left), and the other is a region of the Sydney metropolitan area, located southeast from Centennial Park (Figure 1 right). For the NY scenario we consider 5 couples of origin and destinations, while for the Sydney scenario 4 of such couples are used (illustrated by coloured arrows). The networks have been extracted by using OpenStreetMaps, and are simulated in SUMO [15]. The simulation is run for 1 hour, and traffic intensity ranging between 760 and $1,208$ vehicles per hour per origin-destination is used. The considered traffic intensity is realistic with regards to the considered regions.

B. Settings

Following the concept of Chrpa et al. [8], the 1 hour simulation time window has been divided into 30 second time windows, each corresponding to an individual planning episode. In total, 120 planning episodes for the 1-hour traffic simulation are generated. In particular, data about incoming vehicles (i.e., their entry and destination points) arriving into the network in a considered 30 second time window are fed to a planning engine – we used the Mercury planner [16] that performs well in such domains [8] – and found plans (routes)

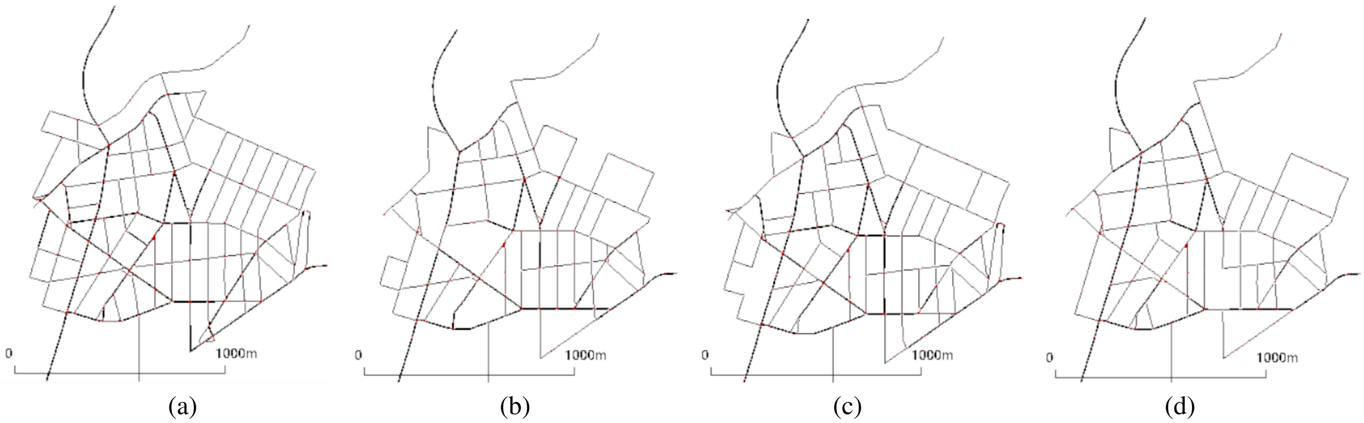

Fig. 2. Simplified road networks of the New York scenario generated by merging the subgraphs obtained by KR (a), MSR (b), MDR (C) and SR (d).

TABLE II
RESULTS OF THE SIMULATION. WE MEASURE WHAT PERCENTAGE OF PLANS WERE GENERATED (OUT OF 120 PLANNING EPISODES), AVERAGE DISTANCE TRAVELLED, AVERAGE SPEED, AVERAGE TRIP DURATION AND AVERAGE WAITING TIME.

	New York					Sydney				
	Planned (%)	Dist (m)	Speed (m/s)	Duration (s)	Waiting (s)	Planned (%)	Dist (m)	Speed (m/s)	Duration (s)	Waiting (s)
Naive	-	2247.54	0.97	4143.74	3391.99	-	2703.49	2.03	3188.7	2730.34
Def	0	2247.54	0.97	4143.74	3391.99	0	2703.49	2.03	3188.7	2730.34
KR	0	2247.54	0.97	4143.74	3391.99	0	2703.49	2.03	3188.7	2730.34
SR	96	2562.18	2.19	2768.87	2072.15	38	2739.1	2.58	2380.25	1940.02
MSR	82	2527.99	1.4	4037.35	3330.49	4	2716.17	2.14	3166.72	2718.38
MDR	45	2366.94	1.48	3755	3101.09	0	2703.49	2.03	3188.7	2730.34

are distributed to vehicles. A maximum planning time is set to 25 seconds (so there is some time to process data before the 30 second time window expires). If no plan is generated within the limit, then the shortest routes are assigned to each vehicle by default. The planning component is connected to the SUMO simulator via the TraCI interface.

For our variant of the A* algorithm (see Algorithm 1) we set the constants as $c = 1.3$ and $K = 10,000$, i.e., we limit the routes to be within a 1.3 bound from the optimal one, and we generate at most $10,000$ routes for each origin destination. For the DBSCAN algorithm we used to cluster routes (see Algorithm 2), we set the constants as $\epsilon = 0.26$ and $q = 4$, i.e., a minimum number of routes in a cluster was 4 and the minimum distance between routes to be considered as similar was 0.26.

The experiments were run on a cluster equipped with AMD EPYC 7543 CPU, and memory usage has been limited to 4GB RAM.

C. Results

From the perspective of road network simplification, the results, summarised in Table I, clearly show that the proposed techniques can considerably simplify the road networks. Although pre-selecting potentially useful roads (that are not much longer than the optimal ones) by our variant of A* (KR) show an improvement, considering a smartly selected set of diverse roads (SR, MSR, MDR) can simplify the road networks even more. This is also illustrated in Figure 2 on the New York scenario, where it is shown what part of the original network (see Figure 1 (a)) is considered for planning after our methods identified representative routes between all origin destinations. It is worth highlighting that the SR method is the one that leads to the most significant simplification of the networks. This is particularly apparent in the Sydney scenarios, where the number of considered road links dropped to about 25%.

Table II summarises the results of the simulation in both scenarios for all our methods. It can be observed that the percentage of successfully solved planning episodes (i.e., the plans were generated within the time limit) usually grows with the decreasing number of junctions and road links the planning engine has to consider while during plan generation. A notable exception are MSR and MDR in the New York scenario, where MDR was less successful in planning than MSR despite producing a smaller road network. Of course, size of the network is not the only indicator that determines how "hard" a given planning problem is for the planning engine. The topology of the road network might also play its role in the planning process.

Other considered metrics, i.e., Average distance, Average speed, Average travel time (or trip duration) and the average waiting time are obtained from SUMO. Speaking about the average travel time, the most successful method is SR by a considerable margin against the other methods. On the other hand, the travelled distance was slightly higher than for the other methods. Since SR was also the most successful method in terms of solved planning episodes, it demonstrates efficiency of the underlying planning approach as it optimise routes for travel time while considering the traffic in the network and such routes might be (a bit) longer but less congested.

D. Discussion

In comparison to related work in using automated planning for intelligent vehicle routing [8]–[10] that considered road networks of sizes about 20 junctions and 50 road links, the introduced approach allows centralised planning-based traffic routing systems to tackle networks that are larger by an order of magnitude. Straightforward application of the planning approach on the road network (as in the related work) have not produced any plan (in both scenarios). Also, pre-filtering routes by the KR method has shown not to be effective enough to deal with such large road networks. Refining the road networks by considering only a subset of diverse enough routes has shown to be promising. In particular, the SR method that select the shortest route from each cluster was the most successful one. Although SR might not have found very diverse routes, it had shown that the clustering itself (by DBSCAN) identifies routes that are diverse enough (in different clusters).

A significant advantage of the proposed approach is that routes can be generated and selected in advance, hence minimising the time needed for online route generation. Further, the approach can straightforwardly take into account planned roadworks or more general traffic network modifications.

V. Conclusion

Intelligent vehicle routing is a growing research area that can reduce the mounting pressure on traffic networks by supporting a better utilisation of the existing infrastructure. Recent works have shown that techniques based on Automated Planning are capable of effective distribution of traffic within urban networks, with significant benefits in terms of reduced travel times and waiting times for the navigating vehicles [8]–[10]. The traditional drawback of these techniques, that leverages on a centralised perspective, is their lack of scalability to large and complex traffic networks. In this paper we addressed the scalability issue by simplifying road networks via smart routes identification. The proposed approach allows the generation of routes that have bounded suboptimality with regards to shortest paths, so the travelled distance (and consequently travel time) will not considerably increase for some of the vehicles. To compute such routes, we proposed a variant of the A* algorithm, and introduced a way for clustering routes and selecting the most representative ones, with the aim of avoiding undesired high levels of similarity.

The performed experimental analysis, that considered two scenarios – a part of New York and a part of Sydney – demonstrated that the proposed methods significantly improve the performance of planning-based traffic distribution by improving scalability while maintaining the positive impact on network performance. Quality-wise, the results have also shown that with increasing success rate of solving planning episodes, the average travel and waiting times dropped considerably.

In future, we plan to focus on explicit identification of bottlenecks in road networks, i.e., road links or junctions with heavy traffic from multiple directions. We believe that information about bottleneck can narrow the set of alternative routes even more. Also, we would like to investigate whether and how the route pre-computing techniques can be used in more reactive fashion that might be important to quickly deal with unexpected circumstances (e.g. traffic accidents).

References

[1] R. Bretherton, "Scoot urban traffic control system: Philosophy and evaluation," in *Proceedings of the 6th IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation*, 1989.

[2] R. Vincent and J. Pierce, "Self-optimising signal control for isolated intersections," in *Crowthorne: Transport and Road Research Laboratory Research Report*, no. 170, 1988.

[3] P. Lowrie, "The sydney coordinated adaptive traffic system-principles, methodology, algorithms," in *Proceedings of the International Conference on Road Traffic Signalling*, no. 207, 1982.

[4] M. Vallati and L. Chrpa, "A principled analysis of the interrelation between vehicular communication and reasoning capabilities of autonomous vehicles," in *Proceedings of ITSC*, 2018, pp. 3761–3766.

[5] A. Kirimtat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, "Future trends and current state of smart city concepts: A survey," *IEEE access*, vol. 8, pp. 86 448–86 467, 2020.

[6] A. M. Falek, A. Gallais, C. Pelsser, S. Julien, and F. Theoleyre, "To re-route, or not to re-route: Impact of real-time re-routing in urban road networks," *Journal of Intelligent Transportation Systems*, vol. 26, no. 2, pp. 198–212, 2022.

[7] M. Ghallab and D. Nau and P. Traverso, *Automated Planning Theory and Practice*. Elsevier Science, 2004.

[8] L. Chrpa, M. Vallati, and S. Parkinson, "Exploiting automated planning for efficient centralized vehicle routing and mitigating congestion in urban road networks," in *Proceedings of SAC*, 2019.

[9] M. Vallati and L. Chrpa, "Effective real-time urban traffic routing: An automated planning approach," in *7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS*, 2021, pp. 1–6.

[10] M. Vallati, E. Scala, and L. Chrpa, "A hybrid automated planning approach for urban real-time routing of connected vehicles," in *24th IEEE International Intelligent Transportation Systems Conference, ITSC*. IEEE, 2021, pp. 3821–3826.

[11] 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*, vol. 4, no. 2, pp. 100–107, 1968.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[13] J. Y. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quarterly of Applied Mathematics*, vol. 27, pp. 526–530, 1970.

[14] H. Aljazzar and S. Leue, "K∗: A heuristic search algorithm for finding the k shortest paths," *Artificial Intelligence*, vol. 175, no. 18, pp. 2129–2154, 2011.

[15] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *Proceedings of ITSC*, 2018.

[16] C. Domshlak, J. Hoffmann, and M. Katz, "Red-black planning: A new systematic approach to partial delete relaxation," *Artif. Intell.*, vol. 221, pp. 73–114, 2015.