Towards a Framework for Intelligent Urban Traffic Routing

Matyáš Švadlenka and **Lukáš Chrpa**
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague
svadlmat@fel.cvut.cz, chrpaluk@cvut.cz

Abstract

Intelligent traffic routing is one of the key techniques that can be used to optimise traffic, especially in urban areas. Deliberative reasoning techniques such as Automated Planning have shown their potential since they can take a global and longer-term perspective on the traffic situations. Such techniques have to be embedded in a urban traffic control framework such that they can generate and assign routes to the vehicles on the fly while considering the current traffic situation in the area.

This paper presents an ongoing work on a framework that, in a nutshell, integrates an automated planning component, responsible for intelligent traffic routing, in the well known SUMO simulator in order to evaluate and study its impact in realistic traffic settings. In particular, the framework has to simplify the representation of the road network provided by SUMO, translate it into PDDL, a language for describing planning problems, and then interpret plans and fed them in form of vehicle routes into SUMO.

Introduction

Nowadays, traffic in urban areas becomes one of the major economical problems due to losses from traffic accidents and travel delays, especially during rush hours. For example, the cost of congestion in London has exceeded £5 billion in 2020 in lost time and fuel consumption (London 2022), and has become a major health threat (Chang et al. 2019). With continuing growth of global urbanisation, the problem with traffic congestion is expected to exacerbate. The introduction of highly innovative techniques such as *Connected Autonomous Vehicles* (CAVs) has the potential to revolutionise the field as we have more options for designing intelligent traffic control techniques to mitigate the problem (Vallati and Chrpa 2018; Rasheed et al. 2017; Arena and Pau 2019).

Intelligent traffic routing aims at planning routes for vehicles while minimising a given objective function such as travel time or fuel consumption. Modern SATNAV systems (e.g. Waze™) can utilise current information about traffic to generate routes that have more accurate estimation of the actual travel time. That is, such systems increase chance for drivers using them to avoid possible delays in heavy traffic or congested roads. Centralised approaches, on the other hand, generate routes for every CAV at once and thus take a global perspective over the controlled urban region. In particular, centralised traffic routing techniques can plan for all CAVs entering the controlled region or requiring a new route inside the region at once and optimise generated routes by a "global" objective function (e.g., average travel time).

Automated planning approaches had recently gained promising results in centralised traffic routing (Chrpa, Vallati, and Parkinson 2019; Vallati and Chrpa 2021a) as well as in traffic light control (Vallati et al. 2016; Pozanco, Fernández, and Borrajo 2021). That inspired us to develop a framework that integrates an automated planning component, responsible for intelligent traffic routing, in the well known SUMO simulator (Lopez et al. 2018) in order to evaluate and study its impact in realistic traffic settings. This paper presents an ongoing work on that framework that at this stage focuses on effective translation of the road network representation of SUMO into the PDDL language in which planning tasks can be described (Fox and Long 2003). The translation has to simplify the network by abstracting some its elements (e.g. roundabouts) and by precomputing parts of the road networks that can be used for routing in specific directions. As a planning domain model we use the one developed by Chrpa, Vallati, and Parkinson (2019). Our framework is evaluated on four scenarios concerning urban areas taken from OpenStreetMap™.

SUMO Simulator

Simulation of Urban MObility (SUMO) (Lopez et al. 2018) is an open-source microscopic urban traffic simulator that simulates traffic at the level of individual vehicles (and other objects). In a nutshell, SUMO operates over a road network, which can be taken from OpenStreetMap™, in which individual vehicles have their routes that they follow. Also, SUMO takes into consideration traffic rules (e.g. giving a way when merging from a side road) as well as traffic lights (if present on a junction). Hence, SUMO can realistically capture evolution of a traffic situation in a specified region and provide useful data about how traffic evolves measured by a number of quantities (e.g. average travel time, average travelled distance). In the context of centralised traffic routing, SUMO can provide a valuable feedback regarding the quality of vehicle routes (e.g. how average travel time decreases or increases with respect to default settings) and, for example, where traffic jams tend to evolve if traffic is heavier (such as in rush hours).

In general, SUMO supports intermodal simulation that besides traffic simulates behaviour of other entities such as pedestrians or public transport. In the current state of development of our framework, we resort only to traffic simulation without taking other modalities (e.g. pedestrians) into an account.

SUMO contains a number of useful components that allow users to operate with the simulator. We introduce those that are relevant to our framework.

- *osmWebWizard* can be used to capture regions on OpenStreetMap™; however, the resulting representation contains much more data that are not necessary for our purpose (e.g. geometry of the buildings).
- *netconvert* is an alternative to osmWebWizard that allows to convert maps from OpenStreetMap™, VISSUM and other sources into the SUMO representation (in XML). Netconvert is a command-line application that is simple to use by specifying only a type of the file together with the "map" file.
- *netedit* is an application with Graphical User Interface that can visualise and edit road networks (in SUMO representation). The tool also can be used to fix the issues in the road network that were introduced while importing it, or to update the network reflecting possible changes (e.g. road closure for maintenance).
- *Traffic Control Interface (TraCI)* is an interface that can connect SUMO with reasoning techniques (e.g. planning). TraCI gives an online access to the simulation to retrieve the current traffic situation and allows modifying behaviour of the vehicles or objects (e.g. by assigning a different route).

Centralised Traffic Routing Problem Formulation

In a nutshell, the problem of *centralised traffic routing* deals with finding a route for each individual vehicle that goes from its location of origin to its destination location while optimising a given "global" objective function (such as average travel time).

The road network is represented by a *directed graph* in which vertices represent *junctions* and edges *road links* (or segments) that connect the neighbouring junctions. Then, there is a set of *vehicles* such that each vehicle has its locations of *origin* and *destination*, which are, in our case, vertices of the graph, and the time in which it "appears" in the location of origin. The task is to find a path in the graph for each vehicle such that the path goes from vehicle's location of origin to its destination location such that it optimises the given objective function (e.g. average travel time). From the paths in the graph, we can straightforwardly get routes that are provided into SUMO. SUMO then simulates the traffic in which each vehicle follow its generated route in order to retrieve the actual value of the given objective function (e.g. average travel time).

Planning for Centralised Traffic Routing

Automated Planning is a prominent field of artificial intelligence that, in a nutshell, deals with the problem of finding a sequence of actions that modify the state of the environment to achieve a given goal (M. Ghallab and D. Nau and P. Traverso 2004). A planning task can be specified in the PDDL language (Fox and Long 2003) such that one specifies a *domain model*, consisting of description of the environment (by predicates, for example) and actions, and a *problem instance* consisting of an initial state and a goal. *Actions* are specified via preconditions, i.e., what has to hold to make the action applicable, and via effects, i.e., how the state is modified by applying the action.

In this paper, we rely on the model introduced by Chrpa, Vallati, and Parkinson (2019) that is based on a microscopic traffic representation, i.e., it represents each (relevant) individual vehicle. The road network is represented in the model in the same way as mentioned in the previous section, i.e., by a directed graph. Vehicles are navigated between graph vertices (junctions) by *drive* actions that move a vehicle from one junction (j_1) to another (j_2) if there is an edge going from the j_1 vertex to the j_2 vertex in the graph, i.e., the junctions are connected by a road link. The *drive* actions also consider traffic intensity on the road links such that a *drive* action increases the counter that counts the number of routed vehicles on a corresponding road link by one whenever the action is applied. Note that the model uses "predicate counters" instead of numeric fluents to represent the vehicle counters.

The model distinguishes three levels of traffic intensity – *light, medium* and *heavy* – that are derived from the capacity of the given road segment (i.e., the maximum theoretical number of vehicles that can physically fit onto the road link). According to the traffic intensity level on a given road link determined by the number of vehicles on it, one of four variants of the *drive* action is applied, i.e., *drive-light*, *drive-medium*, *drive-heavy* and *drive-congested*. Note that the *drive-congested* action allows the use of road links where the number of vehicles exceeds their capacity. However, in cases of heavy traffic, it might not be always possible to route traffic around "bottleneck" road links. In order to minimise traffic intensity on road links, the cost of *drive* actions depends on the traffic intensity such that the cost is higher for *drive* actions concerning a higher traffic intensity level (e.g. *drive-heavy* is more expensive than *drive-medium* which is more expensive than *drive-light*).

The model also introduces an *allowed* predicate that represents whether a road link can be used by a vehicle with a certain destination location. The *allowed* predicate is added into the precondition of the *drive* actions. It should be noted that the *allowed* predicates forbids routing of vehicles to road links that do not lead to the destination locations of these vehicles. That prunes some unpromising alternatives in the search space and improves efficiency of planners.

The task is to "move" all the vehicles from their location of origin to their destination locations while minimising the total cost of used *drive* actions in the plan. It means that the plans aim at minimising traffic intensity levels for all road links in the network. Although a plan contains *drive* actions for all vehicles, the routes for individual vehicles can be straightforwardly extracted from the plan by "projecting" to a sequence of *drive* actions for a particular individual vehicle (one by one). For details about the model the interested reader is referred to (Chrpa, Vallati, and Parkinson 2019).

Architecture of the Framework

Our framework is designed to connect the SUMO simulator with a planning component that is responsible for generating routes for vehicles in order to optimise traffic flows. The architecture of our framework is depicted in Figure 1.

Converting Road Network Maps to SUMO

The left hand side of the architecture is responsible for generating road network representation from a selected urban area on OpenStreetMap™ (technically, other sources of maps can be used as well if supported by *netconvert*). We have used *osmfilter*, which is a tool provided by OpenStreetMap™, to preprocess the data by filtering out unnecessary information (e.g. buildings). Then, we leveraged the *netconvert* component of SUMO that can process the preprocessed data capturing the road network in the area and output it into a xml file that SUMO uses. This step can be done offline and needs to be done only once for a given region unless the road network in that region changes (then a new xml file representing the road network has to be generated).

It should be noted that the road network (in an xml file) can be visualised and edited in the *netedit* component. Besides the graphical visualisation of the network (in the SUMO format), it might be important to sometimes fix issues (e.g. incorrect number of lanes) that might arise during the automated acquisition of the road network data from OpenStreetMap™. Also, *netedit* can be used to amend the road network to account for planned road closures or to consider some minor changes in the road network (e.g. changes in the right of way).

Simplifying Road Networks

The *GraphLauncher* component of our framework is responsible for simplifying the SUMO representation of the road network such that the planning component can more effectively reason with it. Although it is possible to convert the road network in SUMO representation directly into a planning task specification (in PDDL), it is useful to simplify the road network in a preprocessing step, so a planning engine needs to reason only with promising alternatives while routing traffic.

One way how the road network can be simplified is to abstract certain elements of it. Junctions with one incoming and one outgoing road segments can be abstracted out and the road segments are merged into one. These junctions might occur in the SUMO representation due to, for instance, graphical reasons or some glitches in the OpenStreetMap™ model. Also, more complex junctions such as roundabouts consist of, in the SUMO representation, a number of (individual) junctions connected by road segments (e.g. parts of a roundabout). We abstract such complex junctions into a single one. The above two simplifications usually simplify the networks by about 10% (measured the the number of junctions and road segments).

To further simplify the network we investigated how reasonable routes between given origin and destination locations can look like. At this stage of the development we considered routes that are at most c times longer than the optimal (shortest) route (between a given origin and destination location). For this purpose, we have developed a variant of the well known A* algorithm (Hart, Nilsson, and Raphael 1968) that after finding the shortest route continues the search until either the current node value is greater than c times the shortest route length, or the number of found routes is higher than K. The output of that variant of A* is a set of routes (from a given location of origin to a given destination location) that forms a subgraph of the graph representing the road network. An example depicting a map of a part of Dejvice in Prague, Czech Republic, is in Figure 2 left. The result of our variant of A* for each traffic flow, determined by a given location of origin and a destination location (illustrated by arrows of a different colour in Figure 2) is depicted in Figure 3.

Then, the subgraphs generated for each origin/destination pair are merged into a graph representing a simplified road network. An example of the result of the simplification can be seen in Figure 2 right. Also, we can set the *allowed* predicates in the planning model (see the above section) such that for each flow we consider only road links in its corresponding subgraph.

Both pre-processing steps can be done once as long as the topology of the road network or considered traffic flows does not change. Hence, this step can be done offline and does not introduce any overhead during planning (or traffic routing).

Generating Traffic Scenarios

The *ScenarioLauncher* component of our framework is responsible for generating traffic scenarios. In particular, it takes into account traffic flows determined by locations of origins and destinations (usually where a given flow enters the region and where it leaves the region) and by traffic intensity measured by the number of vehicles per minute. At this stage of the development we consider *uniform* and *increasing* flows while taking some (small) fluctuations into account (e.g. for a uniform flow the number of vehicles is not exactly the same for every minute). Increasing vehicle flows simulate situations in which the traffic intensity increases in a given time span (e.g. beginning of a rush hour).

Each traffic scenario considers a defined time span (e.g. 30 seconds) in which a number of vehicles is expected to enter the region. From the defined flows, we can determine the number of vehicles for each direction and then for each vehicle we specify its initial location (where the corresponding flow starts) and its goal location (where the corresponding flow ends).

Planning Component

The right hand side of our architecture contains the planning component that is responsible for generating routes for the vehicles. The planning component considers the domain model of Chrpa, Vallati, and Parkinson (2019), men-

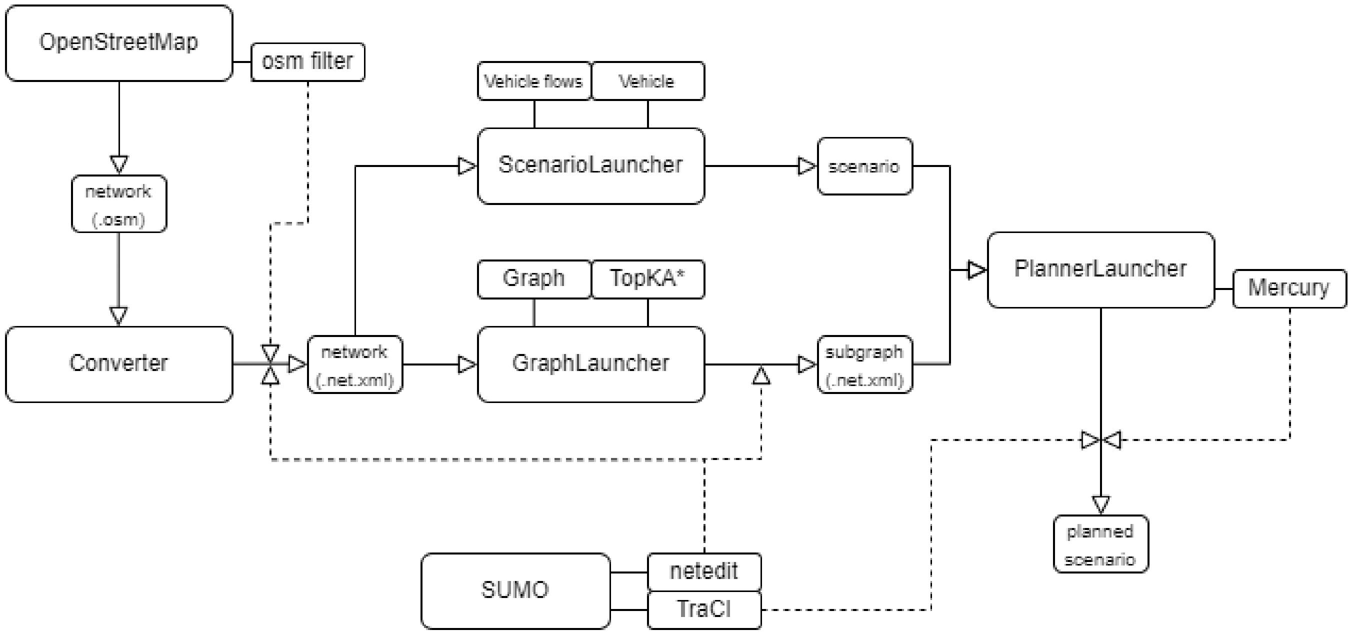

Figure 1: Architecture of our framework

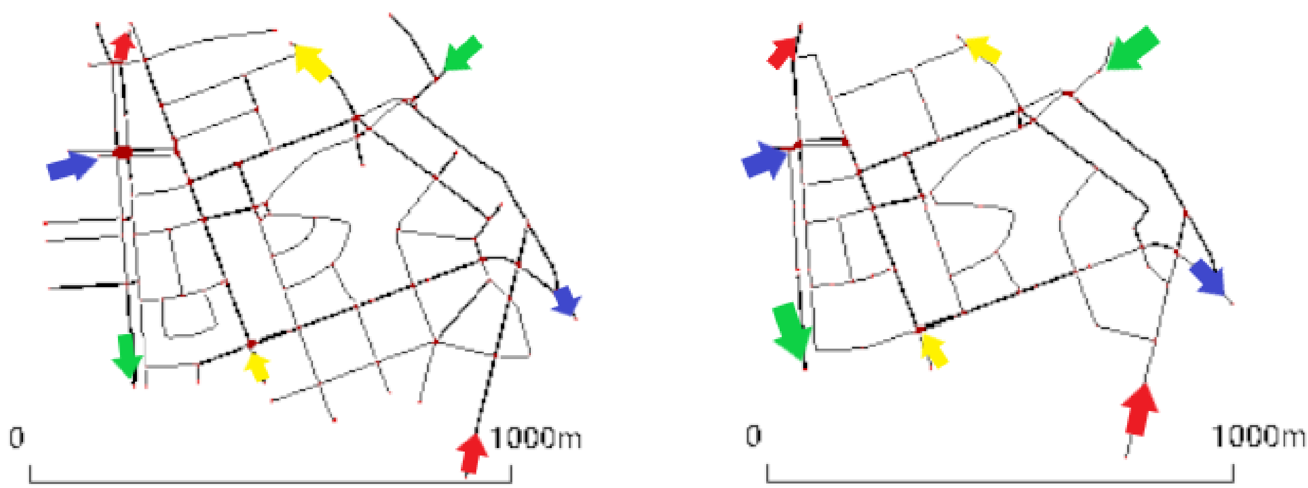

Figure 2: An example of an urban road network with arrows of different colours determining the considered origin and destination locations (left) and a simplified variant by our variant of the A* algorithm (right).

tioned in the previous section. Problem instances, generated by *ScenarioLauncher*, which is responsible for vehicle information, and *GraphLauncher*, which is responsible for road network information, represent planning episodes that consider vehicles that need routing (e.g. when they enter the region) in a given time span (e.g. 30 seconds). For plan generation, we used the Mercury planner that is a domain-independent planning engine that incorporates Red-black heuristics (Domshlak, Hoffmann, and Katz 2015) as it is efficient in the given domain (Chrpa, Vallati, and Parkinson 2019). Generated plans are then translated into routes of the vehicles that are passed to the SUMO simulator, which is depicted in the bottom side of our architecture, via the TraCI interface.

Experiments

Our experimental analysis aims at evaluating possible impact of centralised routing techniques on traffic in urban areas in a rush hour. We considered four maps of metropolitan urban areas, namely Dejvice (Praha), South Bronx (New York), Clerkenwell (London) and Clovelly (Sydney), that range from about 90 junctions and 180 road segments to more than 440 junctions and 1000 road segments per a map. For each map, we considered a one hour simulation that is

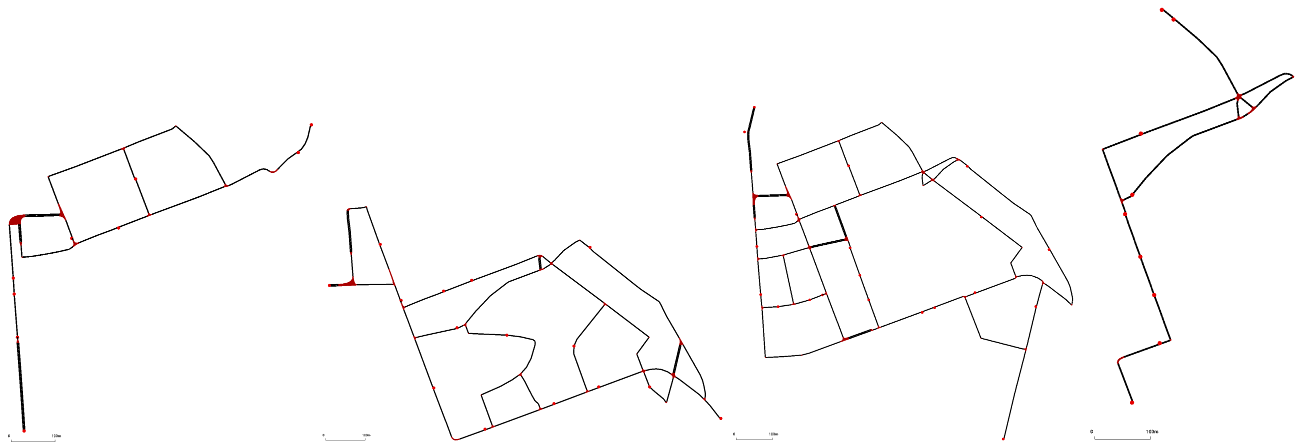

Figure 3: Subgraphs of the road network generated by our variant of A* for particular flows – green, blue, red, yellow (from left to right).

Map	Simplified size (%)	Plans (%)	Distance (m)	Speed (m/s)	Travel time (s)	Waiting time (s)
Dejvice (Praha)	71	100	1282.83	2.68	1140.28	873.46
			1329.01	4.12	989.26	767.51
South Bronx (New York)	71	74	2280.52	1.09	4077.99	3447.53
			2469.07	1.9	2384.96	1887.96
Clerkenwell (London)	61	55	2088.74	2.12	1549.7	1076.7
			2221.15	2.1	1718.83	1220.76
Clovelly (Sydney)	34	25	2593.68	3.02	1610.0	1215.98
			2671.83	3.17	1573.53	1173.32

Table 1: Results of the simulation on the four considered scenarios. Values for Distance, Speed, Travel and Waiting time are averages per a vehicle. First line refers to a naive routing (shortest route) while the second line refers to centralised routing via planning.

split into 120 planning episodes where one episode routes vehicles coming to the network in a 30 second time span. We consider 3 different flows (4 for Dejvice map) – pairs of origin and destination locations – that are randomly assigned with uniform or increasing traffic with the number of vehicles per hour ranging from 2700 to 3700 (per map). For each map, we generated 10 scenarios with different flows. The parameters for our variant of A*, c and K, were set to 1.3 (1.4 for Dejvice map) and 3000, respectively. Planning time was limited to 25 seconds (so we can get the routes for each 30-second episode on time). If the planner (Mercury) failed to generate routes in some case, then the shortest routes were assigned to the vehicles for that planning episode. Experiments were run on Intel i7-8700 (3.2 GHz) and 4GB of RAM[1].

The results in Table 1 show that centralised routing via planning has a potential to improve road traffic in the South Bronx scenario by about 40% (measured by average travel time) where nearly $3/4$ of planning episodes were successfully solved. In the Dejvice scenario, the improvement of the average travel time was about 14% with all planning episodes being solved in the time limit. In other scenarios the results are not that great, which is caused by a smaller number of solved planning episodes and some peculiarities of the road network that the planning model does not properly capture. It can be seen that our simplification techniques can reduce the size of the road network representation that the planning engine has to consider by $30-65\%$.

Manhattan-like topology of the South Bronx road network allows for more effective distribution of traffic and hence cetralised routing techniques achieved very good results. In Clerkenwell, however, using shorter routes even with (slightly) higher traffic intensity is better than trying to reroute some traffic to longer routes. Another peculiarities that might not be captured well by the planning model are situations in which vehicles from a side road have to merge (or cross) a main road on an uncontrolled junction. If the traffic on the main road is intense enough, then vehicles on the side road get stuck for a long time (albeit in reality in such situations drivers on the main road allow the vehicles on the side road to merge, SUMO does not capture such a behaviour).

[1]Source code of the framework and benchmark data can be found at `https://github.com/Matyxus/FLAIRS`

Discussion and Conclusion

Centralised traffic routing techniques in the light of new revolutionary technologies such as CAVs are getting more importance due to their ability to look at the problem of traffic routing from a global perspective which might be important to prevent traffic congestion in some area by smart routing of the vehicles. Recently, automated planning techniques have shown some promise in centralised traffic routing (Chrpa, Vallati, and Parkinson 2019; Vallati and Chrpa 2021b). Such an advancement motivated our work that aims at incorporating centralised routing techniques into the SUMO simulator that provides a realistic simulation of urban traffic (Lopez et al. 2018).

In this paper, we report an ongoing work on the framework that at this stage of the development allows to "connect" SUMO with a planning component responsible for centralised routing. That involved acquiring data about the road network from OpenStreetMap™ by components of SUMO (e.g. *netconvert*) into an xml format. The xml representation of the road network has to be simplified before being translated into PDDL and on top of that an information about vehicles needing routing has to be provided within the PDDL problem file as well. That the PDDL problem file alongside the PDDL domain model (taken from (Chrpa, Vallati, and Parkinson 2019)) is fed into a planning engine (e.g. Mercury (Domshlak, Hoffmann, and Katz 2015)) and plans are translated into an xml file describing vehicle routes which is an input to SUMO that simulates the routes. Since the SUMO representation might be too complex (large) to be straightforwardly compiled to PDDL and used by planners, we have designed and developed a variant of the A* algorithm that generates a subnetwork in which vehicles navigating in a given direction are very likely to stay. Then the planner might need to reason only on parts of the road network relevant to the direction of vehicles that need to be routed. Our preliminary experiments have shown some promise of the use of planning techniques in centralised traffic routing (especially, in the South Bronx scenario).

The preliminary results also identify some opportunities for improvement. For instance, simplification of road networks has to be stronger in order to accommodate larger networks. One possibility is to generate several diverse routes for each direction (instead of a subnetwork) that the vehicles can navigate, where we have already achieved promising results (Švadlenka, Chrpa, and Vallati 2023). The routes should be diverse enough to mitigate the risk of routing traffic via the same bottlenecks (junction or road segments that are prone to traffic jams). Also, the model should better capture some aspects that can (strongly) influence traffic such as merging from a side road onto a main road in an uncontrolled junctions. Last but not least, different types of vehicles (e.g. trucks, buses) should be also considered in the routing techniques as they have different demands and influence traffic flows in a different way.

Acknowledgements

This research was funded by the Czech Science Foundation (project no. 23-05575S) and the OP VVV project no. EF15_003/0000470 "Robotics 4 Industry 4.0".

References

Arena, F., and Pau, G. 2019. An overview of vehicular communications. *Future Internet* 11(2):27.

Chang, K.-H.; Hsu, P.-Y.; Lin, C.-J.; Lin, C.-L.; Juo, S.-H. H.; and Liang, C.-L. 2019. Traffic-related air pollutants increase the risk for age-related macular degeneration. *Journal of Investigative Medicine* 67(7):1076–1081.

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

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

Fox, M., and Long, D. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.* 20:61–124.

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

London, G. A. 2022. Cost of congestion in capital revealed as car use remains high. `https://www.london.gov.uk/press-releases/mayoral/cost-of-congestion-in-capital-revealed`. Accessed: 2023-01-15.

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

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

Pozanco, A.; Fernández, S.; and Borrajo, D. 2021. Online modelling and planning for urban traffic control. *Expert Systems*.

Rasheed, A.; Gillani, S.; Ajmal, S.; and Qayyum, A. 2017. Vehicular ad hoc network (vanet): A survey, challenges, and applications. In *Vehicular Ad-Hoc Networks for Smart Cities*. Springer. 39–51.

Vallati, M., and Chrpa, L. 2018. A principled analysis of the interrelation between vehicular communication and reasoning capabilities of autonomous vehicles. In *Proceedings of ITSC*, 3761–3766.

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

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

Vallati, M.; Magazzeni, D.; De Schutter, B.; Chrpa, L.; and McCluskey, T. L. 2016. Efficient macroscopic urban traffic models for reducing congestion: a PDDL+ planning approach. In *Proceedings of AAAI*.

Švadlenka, M.; Chrpa, L.; and Vallati, M. 2023. Improving the scalability of automated planning-based vehicle routing via smart routes identification. In *8th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS*, to appear.