# A Hybrid Automated Planning Approach for Urban Real-time Routing of Connected Vehicles

Mauro Vallati[1] and Enrico Scala[2] and Lukáš Chrpa[3]

***Abstract*— One of the pivotal challenges presented to urban road traffic controllers is the effective utilisation of transport infrastructure, as a result of growing urbanisation, the finite network capacity, and of the increasing number of road vehicles. The arrival of connected autonomous vehicles (CAVs) represents a unique opportunity for a fundamental change in urban mobility, and urban traffic control should take an active role in integrating CAVs into the mobility ecosystem in order to maximise benefits. A practical way to exploit CAVs for urban traffic control is to build an intelligent control mechanism that can distribute traffic in the controlled region. In this context, automated planning, a well-studied Artificial Intelligence topic, can provide techniques to dynamically generate plans to distribute traffic –thus maximising the network utilisation and reduce congestion. In this paper, we present an approach based on hybrid discrete continuous planning, and we demonstrate its impact using real-world historical data of a large UK town.**

## I. INTRODUCTION

Over half of the world's population now lives in cities and global urbanisation continues at a steady pace. In the UK alone, the cost of congestion has reached nearly £8 billion in 2018 in lost time and fuel consumption, and has become a major health threat that goes beyond the cardiac and respiratory systems [1].

The arrival of Connected Autonomous Vehicles (CAVs) presents a unique opportunity for a fundamental change in urban mobility and urban traffic control. CAVs can communicate with other vehicles and with the infrastructure, via dedicated protocols, to take better informed decisions. Considering that the general importance and improved capabilities attainable via different types of communication have been well argued [2], and a number of protocols and technologies to implement such communications has been presented [3], here we focus on how urban traffic control can exploit communication with vehicles to improve the use of the controlled network. In particular, the advent of connected vehicles can pave the way to intelligent and sophisticated approaches for dynamic traffic assignment. Dynamic traffic assignment aims at routing traffic in a road network by estimating the time-varying traffic flows on each of the links within a road network, with the overall objective of avoiding congestion and minimising delays via a better use of the capacity of the network [4]. This is a different approach from the currently exploited navigation systems, that can propose alternative routes to users, but may suggest the same route to several vehicles, causing congestion on the alternative route.

Automated planning [5] is a field of Artificial Intelligence that has been shown to be capable of solving challenging problems in a wide range of real-world applications. In particular, it has been successfully applied to urban traffic management [6], [7], [8]. One of the main advantages of automated planning is that the description of tasks to deal with and the logic of the domain are specified in a readable and compact declarative language, usually PDDL (Planning Domain Definition Language). This standardised language leads to the development of a wide range of planning engines, which can be used off-the-shelf to solve challenging tasks from any application domain. In one of its most advanced derivations, automated planning is capable of reasoning with hybrid discrete-continuous systems, and the corresponding tasks are represented using the hybrid PDDL+ formalism [9]. This formalism is particularly well positioned to deal with dynamic traffic assignment tasks, as it provides a way to encode the likely evolution of the traffic of a road network, and can support the reasoning side in deciding the best route to assign to vehicles approaching the controlled network.

In this paper, we introduce a microsimulation PDDL+ knowledge model for real-time routing of connected vehicles, inspired by the well-known S-model [10]. We describe how the propagation of traffic within the modelled network can be efficiently performed, and we present an optimisation function that can be used to allow existing state-of-the-art planning engines to generate routes that maximise the distribution of traffic in the network. We then demonstrate its usefulness in a simulation driven by real-world traffic data of Milton Keynes – a large town of the United Kingdom.

## II. AUTOMATED PLANNING

Automated planning deals with the problem of finding a plan (a sequence of actions) that transforms the environment from an initial state to some desired goal state [5].

A planning task comprises a domain model and a problem model, where the former defines operators while the latter consists of objects, initial state and goal condition. The domain model and the problem model are referred to as the *knowledge models*, as they encode all the knowledge needed by an automated reasoner to solve the corresponding task.

The simplest form of planning is classical planning. A classical *planning task* is defined as a quadruple $(F, A, I, G)$, where $F$ is the set of *fluents* such that each fluent $f \in F$ has its own domain $D(f)$, $A$ is the set of *actions*, $I$ is a complete fluent assignment over $F$ representing the *initial*

[1]Mauro Vallati is with the School of Computing and Engineering, University of Huddersfield, Huddersfield, United Kingdom. m.vallati@hud.ac.uk

[2]Enrico Scala is with the University of Brescia, Brescia, Italy.

[3]Lukáš Chrpa is with the Czech Technical University in Prague, Prague, Czech Republic.

*state* and $G$ is a set of constraints over $F$ representing the *goal*. A *state* is a complete fluent assignment over $F$ and we denote $S$ as the set of all states over $F$. We denote $s[f]$ the value of the fluent $f$ in the state $s$.

A *classical action* $a \in A$ is a pair $a = (pre(a), eff(a))$, where $pre(a)$ is a set of constraints over $F$ representing the *precondition* of $a$ and $eff(a)$ is a partial assignment over $F$ representing the *effects* of $a$. We say that an action $a$ is *applicable* in a state $s$ if and only if $s \models pre(a)$. We say that a state $s'$ is the *result* of application of $a$ in $s$ (if possible) if and only if for each fluent $f$ mentioned in $eff(a)$ it is the case that $s'[f] = eff(a)[f]$ and for each remaining fluent $f'$ it is the case that $s'[f'] = s[f']$.

A classical *plan* is a sequence of actions whose consecutive application in the initial state $I$ results in a state in which $G$ is satisfied.

A hybrid PDDL+ planning task extends a classical planning task along two dimensions [9]. First, the set of fluents also involves numeric fluents; Second, there is an explicit representation for the evolution of the environment given in terms of so called processes and events. More formally: a PDDL+ *planning task* is the tuple $(X, F, A, P, E, I, G)$ where: $X$ is the set of numeric *fluents*, each having its domain expressed over the set of rational numbers $Q$; $F$ is as for classical planning tasks; $A$ is the set of numeric actions. Each such action is as classical action with the addition that i) $pre(a)$ can also contain inequalities over variables from $X$, e.g., $x + y \leq 10$ ii) $eff(a)$ can also contain numeric assignments of the form $x = \xi$ where $x \in X$ and $\xi$ is a numeric expressions over variables from $X$; $P$ is the set of processes. As an action, a process $p$ is a pair $p = (pre(p), eff(p))$ with the difference being that $eff(p)$ only involves numeric assignments. Each numeric assignment $x = \xi$ is to be interpreted as specifying that $\xi$ is the time derivative of $x$; $E$ is the set of events. Events are syntactically equivalent to actions; $I$ is as for a classical planning task, with the difference that it also assigns to each variable in $X$ a numeric value from $Q$; $G$ is syntactically equivalent to a precondition of a numeric action.

The application of an action in PDDL+ further modifies all numeric variables with the values specified in the action effects evaluated in the state in which the action is applied. Applicability of actions straightforwardly extends from that defined for the classical case. Indeed, we further requires that also all numeric constraints are satisfied. The difference between actions, and processes and events is that, while actions are under the control of the agent, processes and events are to be considered must transitions. They are applied as soon as their preconditions are satisfied.

A PDDL+ plan is a sequence of time-stamped actions plus a time horizon. A plan is said to be valid *iff* all actions are applicable at their time and the goal is satisfied at the prescribed horizon. Different from classical planning tasks, in PDDL+ the state trajectory evolves continuously through time as for the prescription imposed by the processes and events. In particular, between any two actions where there is some passage of time, all active processes (those having their preconditions satisfied) describe continuous changes, while events automatically trigger instantaneous transitions whenever their preconditions are satisfied. More details about the syntax and semantics of PDDL+ can be found in [9].

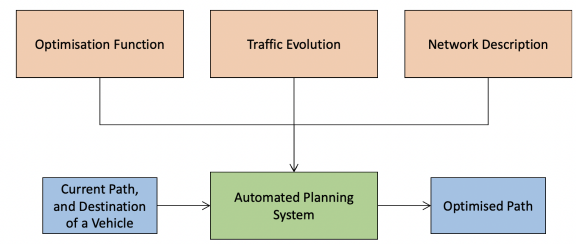


Fig. 1. The input, output, and main components of the planning-based controller.

## III. PLANNING-BASED REAL-TIME ROUTING

In this work we design a planning-based approach for dynamic traffic assignment of connected vehicles. We assume that, when a connected vehicle is approaching the controlled urban region, there is in place an infrastructure that allows the vehicle to communicate its destination, and its current path, to the traffic control system. On the basis of this information, the planning-based controller is then reasoning upon the available data in order to generate a personalised route for the vehicle, which must be communicated back to the vehicle in a matter of seconds. Figure 1 provides an overview of the controller, and highlights the main data that the planning system needs in order to perform its task. In the remainder of this section, we detail each of the components, and describe the required input and output.

### A. Description of the Network

Inspired by previous work using PDDL+ for urban traffic control [8], the controlled network is represented as a set of unidirectional *links*. Each link is characterised by its *length*, and by the fact that it is *connected* to a subsequent link. More formally, two links `A` and `B` are connected, formalised by means of a dedicated Boolean fluent `(connected A B)`, if vehicles reaching the end of link `A` can move to link `B` via a junction. It should be noted that the connected relation is unidirectional, so if link `A` is connected to link `B`, it is not possible for vehicles to go the other way around. We do not explicitly represent junctions in our knowledge model.

Further, each link has also associated an *occupancy* value, that is used to indicate the number of passenger cars that are navigating the specific link at a given time $t$. On the basis of the occupancy, and of the length, we can also characterise the congestion level of a link. From the length of a link, and considering the number of lanes, it is possible to estimate the maximum number of vehicles that can be on a link at the same time. This is a theoretical physical maximum, calculated by considering the average length of a passenger car (5 metres, according to the SUMO simulator [11]) and the minimum average distance between vehicles (3 metres,

```
(= (length LinkX) 800)  ;;value in metres
(= (occupancy LinkX) 0)
(= (density-medium LinkX) 40)
(= (density-heavy LinkX) 70)
(= (timeNeeded-light LinkX) 64)  ;;value in seconds
(= (timeNeeded-medium LinkX) 116)
(= (timeNeeded-heavy LinkX) 195)
(connected LinkX LinkY)
...
```

Fig. 2. An excerpt of the PDDL+ code used to encode the relevant information for a link (LinkX) of the controlled region.

for SUMO). This information allows us to define the density level of a single link, in terms of the number of vehicles in a link with regards to its physical maximum capacity. We designed a three-level system, corresponding to a link density being *light*, *medium*, or *heavy*. The level of density of a link is then discriminated via a threshold value, specified in terms of the occupancy of the link via predicates *density-medium* and *density-heavy*.

Our knowledge model relies on the fact that there is a relation between density and velocity [12], and for the sake of reducing the computational complexity, we consider that relation on a link by link basis. We therefore introduce a *timeNeeded-light*, *timeNeeded-medium*, and *timeNeeded-heavy* value for each link; this encodes the expected time that a vehicle would spend to navigate the link. These times are function of the length of the link, and of the expected average speed for the corresponding level of density.

Figure 2 provides an overview of the PDDL+ excerpt that encodes the information for a specific link.

### B. Evolution of Traffic

This component is in charge of the propagation of traffic within the modelled network. It considers the propagation at a microscopic level, and relies on data collected from vehicles that are already navigating the controlled region, and in particular their current location, final destination, and the assigned route. Such information is encoded in PDDL+ using two dedicated predicates: *at* that is used to indicate the current location (link) where the vehicle is, and *next*, that is used to describe the sequence of links that composes the assigned path. For example, the set of predicates `(at Vehicle1 LinkX) (next Vehicle1 LinkX LinkY) (next Vehicle1 LinkY LinkZ)`, would encode in the knowledge model the fact that `Vehicle1` is initially at link `LinkX`, and is reaching `LinkZ` via `LinkY`. Longer routes can be represented by adding more *next* predicates to the knowledge model.

On the basis of vehicles' data, and considering the network description, traffic propagation is done by leveraging one of the main advantages of the selected hybrid planning framework: the ability to compactly represent dynamic changes and to efficiently simulate them. More specifically, the evolution of traffic is modelled using the event constructs provided by the PDDL+ language. An event is triggered when a vehicle reaches the end of a link, and calculates the time needed to navigate the next link by considering

```
(:event progress-simulated-medium
   :parameters (?c - vehicle ?r - link ?r2 - link)
   :precondition
       (and (= (time) (timeSim ?c))
       (< (occupancy ?r2) (density-heavy  ?r2))
       (>= (occupancy ?r2) (density-medium ?r2))
       (at ?c ?r)
       (connected ?r ?r2)
       (next ?c ?r ?r2))
   :effect
       (and (increase (occupancy ?r2) 1.0 )
   (decrease (occupancy ?r) 1.0 )
   (not (at ?c ?r))
   (at ?c ?r2)
       (increase (timeSim ?c)
                (timeNeeded-medium ?r2))))
```

Fig. 3. An example event that is used to model the movement of a vehicle that is already navigating the network.

its density level. This time is then stored into a numeric predicate *timeSim* that is specific for each vehicle. There is a different event for each level of density. An example of an event is shown in Figure 3. For a considered vehicle (`?c`), the event is triggered when the vehicle reaches the end of a link (`?r`) according to the estimated `timeSim`. Considering the density level of the next link that is on the route of the vehicle, the position of the vehicle is updated (`at` predicate) and the occupancy levels of the corresponding links. Finally, the estimated time is updated for the next link.

Using this event-based approach, it is possible to model the evolution of traffic in the controlled region, and to propagate vehicles by taking into account the density levels of the links of the network.

### C. Optimisation Function

When distributing traffic in a controlled region, there can always be a tension between the best route for the single vehicle, and the best route for the overall network. A vehicle is usually interested in reaching the destination as soon as possible, while the network controller perspective is to balance the traffic in the region, and optimise aspects such as congestion, air quality, utilisation, etc.

In our work, a route $L$ that initiates at time $t_0$ has a cost calculated according to Equation 1. For each link $l$ included in the route a $penalty$ value is calculated on the basis of the density of $l$ at the expected time of arrival of the vehicle $v$, $t_p$. The higher the density level of $l$, the higher the penalty value.

$$cost(v, L, t_0) = \sum_{l \in L} penalty(l, t_p) \qquad (1)$$

In other words, the $cost$ of a route for a vehicle $v$ is a function of the density levels of the links included in the sequence, at the time when they will be traversed by the considered vehicle. In our implementation, we gave a penalty of 0 for links that have a light density level, 10 for medium, and 100 for heavy. The values can be optimised according to the preferences of the traffic controller, and to the characteristics of the region.

```
(:process move-vehicle
   :parameters (?c - vehicle)
   :precondition (and
       (> (modelTime ?c) 0.0))
   :effect (and
       (decrease (modelTime ?c) (* #t 1 ) )))
```

Fig. 4. The process in charge of modelling the time needed by a vehicle to navigate a link.

Given the *cost* notion defined above, the planning system is optimising, for all the vehicles $V$ entering the controlled region at time step $t_i$, the following objective:

$$\forall v \in V, minimise(cost(v, L, t_i)) \quad (2)$$

Notably, for the sake of this analysis, and in order to minimise the burden on vehicles and limit the computational complexity on the controller side, we assume that vehicles can be rerouted only when they are on the boundaries of the controlled region. In other words, the controller can only consider the option of re-routing vehicles that are on links directly connected to an entry point.

### D. Route Generation

For a new vehicle $v$ that is entering the controlled region, and taking into account the information provided by the three components described above, the planning system generates an optimised route, that is then communicated back to the vehicle. More formally, the planning system receives as input a knowledge model expressed in PDDL+ that includes all data gathered from the three components, and the origin and destination of the new vehicle. The origin position of the vehicle is described using the *at* predicate, and the same predicate is used in the goal description to indicate the desired destination. Via a dedicated set of operators, the planning engine can then identify a sequence of links that allows the vehicle to reach its destination. The operators have a different associated cost, according to the expected density level of the corresponding link. The time needed by a vehicle to traverse a link is measured using a dedicated PDDL+ process *move-vehicle*, shown in Figure 4, that updates the *modelTime* of the vehicle.

Routes are to be generated in a matter of seconds, as they are tailored for each vehicle and produced in real-time. Since optimally-good routes, according to the objective function described in Section III-C, are computationally expensive to generate, we implemented a constraints-based approach. The planning system is initially required to generate a path with an associated *cost* of 0. In other words, a path that includes only link that are deemed to have a light density level when the vehicle is expected to reach them. If that is not possible, the allowed cost is increased by 50 units; if, again, no solution can be found, the restriction on costs is lifted. This approach allows to quickly explore paths with low associated penalties, and only as a last resort links heavily congested are considered.

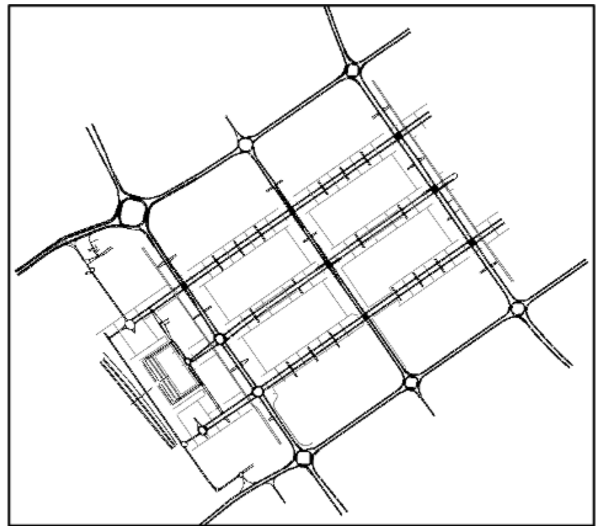

Fig. 5. The modelled central Milton Keynes urban area. Avebury, Midsummer, and Silbury Boulevard are the three parallel road links traversing the area SW to NE.

## IV. EMPIRICAL EVALUATION

To assess the usefulness of the proposed centralised architecture, here we consider a SUMO [11] microsimulation model of Milton Keynes centre. The network is shown in Figure 5. Milton Keynes is a town of the United Kingdom, located about 80 kilometres north-west of London. Milton Keynes has a population of circa $230,000$. The model covers an area of approximately $2.9$ square kilometres, includes more than $25$ junctions and more than $50$ links, and the total length of the links is of approximately $46$ kilometres.

The model simulates the morning rush hour, and has been built by considering historical traffic data collected between 8am and 9am on non-holiday weekdays. Data has been provided by the Milton Keynes Council, and gathered by sensors distributed in the region between December 2015 and December 2016. Traffic signal control information has been provided again by the Council. The model has been calibrated and validated.

During the modelled period, $1,900$ vehicles enter the controlled region either to navigate through it, or for reaching one of the parking slots in the area. No vehicle is initially in the area, but they are all injected over the simulation time. All vehicles have a pre-defined destination and an initially assigned route. The largest flow of traffic comes from the west entry points, via the large North Grafton roundabout (the largest roundabout of the map). Many residential areas are connected with the centre of Milton Keynes through those entry points.

The framework presented in the previous section has been implemented in Python, and uses the TraCI interface[1] to interact with the SUMO simulation environment, in order to get the current network status, communicate with approaching vehicles, and inform vehicles of re-routing. For every couple of origin-destination, described by the traffic

[1] https://sumo.dlr.de/docs/TraCI.html

TABLE I

PERFORMANCE ACHIEVED BY THE PROPOSED PLANNING-BASED SYSTEM (CONTROLLED) ON THE CONSIDERED URBAN AREA. DEFAULT INDICATES THAT NO ROUTING IS IN OPERATION.

| | Considered Metrics | |
|---|---|---|
| | Default | Controlled |
| Departed vehicles [#] | 1669 | 1884 |
| Arrived vehicles [#] | 801 | 1464 |
| Avg, speed [m/s] | 0.58 | 2.19 |
| Avg, trip length [m] | 2297.35 | 2098.05 |
| Avg, trip duration [s] | 788.86 | 511.84 |
| Avg, trip time loss [s] | 622.12 | 359.19 |

flows of the model, traffic experts have identified reasonable links to be considered for routing. They did not provide complete paths, but only the set of links that they would consider for distributing traffic for the considered origin-destination couple. All the relevant links have been included in the topology, provided as part of the knowledge model to the planning engine. While alternatives can, in principle, be automatically calculated (see, for instance, [13]), relying on human expertise can allow to exploit some insights that are based on knowledge that is not captured by the symbolic model of the network.

For each link, density levels have been defined as follows. Light density corresponds to occupancy being less than $40\%$ of link's physical maximum capacity; a heavily congested link has occupancy above $60\%$ of its capacity; medium level sits in between. With regards to the expected average velocity of vehicles on links, according to the density levels, we considered that under light density conditions vehicles can travel at $45km/h$ (freeflow); medium density reduces the average velocity to $25km/h$, and heavy density drops velocity to $10km/h$. Those velocities have been used to calculate the numerical values of the *timeNeeded-** predicates described in Section III-A.

The simulation is run for 1 hour and then stopped. For each set of experiments, the simulation is run five times and results are averaged, to account for non-determinism. To generate the routes, we used the state-of-the-art planning engine ENHSP [14], [15], a well-known Java-based planning engine that includes a wide range of domain-independent search techniques and heuristics for solving PDDL+ planning instances. In our experimental analysis, it was set to use $A^*$ as search technique, and the well-known additive delete-relaxation heuristic $h_{add}$ as guiding heuristic [16]. The experiments were run on an Intel i7-4750HQ CPU, 8 GB of RAM and Linux OS.

### A. Results

The simulation results are summarised in terms of the following SUMO-calculated performance indices:

- Number of departed (arrived) vehicles. Indicates the number of vehicles that entered the region (reached destination) during the simulation. A vehicle can enter the region if the entry link has enough space to accommodate it, otherwise it is assumed to queue outside.

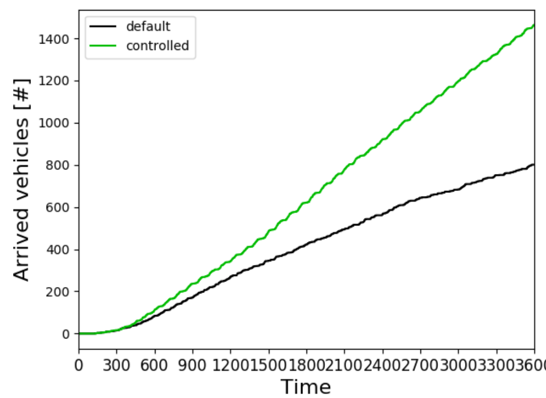


Fig. 6. The number of vehicles that reached their destination with (green) and without (black) the use of the proposed planning system.

- Average speed (m/s) of the vehicles.
- Average trip length (m) and duration (s). Length and duration reports the average measurement of the trips of vehicles to reach their destination from the entry point.
- Average time loss (s). This value indicates the time that has been lost due to vehicles queuing, or travelling at a very low speed.

In a first set of experiments, we consider the ideal scenario: all the vehicles that are navigating through the network are CAVs, and follow the re-routing instructions provided by the automated planning system. Results are presented in Table I. In the Default settings, vehicles enter the network and follow their pre-calculated path to the destination, that does not take into account the network conditions. The presented results suggest that the proposed planning-based approach can effectively route traffic in the modelled region. In particular, the planning-based approach allows the largest number of vehicles to enter the region and to reach their predefined destination.

Figure 6 shows the number of vehicles arriving at destination, over time. As it is apparent, the planning-based controller is capable of effectively exploiting the available road links of the network for distributing traffic. In this way, congestion is avoided, and the traffic is flowing quickly throughout the network. The analysis of the results indicate also that the use of the framework allows vehicles to reach their destination earlier, and can significantly reduce waiting times, i.e. time wasted in queuing. In terms of affected origin-destination routes, we observed that reductions in travel time tend to be evenly distributed; all vehicles are therefore benefitting from the improvement.

To assess the importance of the penetration rate, we also considered experiments where an increasing percentage of vehicles are not communicating with the planning-based controller, and their routes cannot therefore be modified. We considered four penetration rate values: $25\%$, $50\%$, $75\%$, and $100\%$. Interestingly, we observed that the proposed approach is capable of delivering performance close to those shown in Table I even with a penetration rate of $50\%$. With only half of the vehicles following the given instructions, the overall performance of the controlled region are not significantly negatively affected. This is a remarkable result, and it suggests that the proposed controller is capable of adapting its routing generation strategy to such circumstances. Even with a penetration rate of $25\%$, the use of the proposed approach is beneficial when compared to the considered default. Over the simulation time, more than $1,100$ vehicles can reach their destination with an average speed of $1.04m/s$ and an average trip time loss of $441.1$ seconds.

The presented results suggest that significant improvements can be achieved in the near future using planning-based techniques, as soon as communication between centralised urban traffic controllers and vehicles can be established.

## V. RELATED WORK

Automated planning has been exploited to address a range of problems within the urban traffic control field. The generation of traffic light control strategies has been tackled by using PDDL+ [7], [8]: traffic has been modelled at a mesoscopic level, and a forward heuristic search has been implemented to generate the strategies. On a similar note, classical planning [17], [18] has been used, in cooperation with SUMO, to generate traffic signal control strategies using macroscopic models: traffic is modelled using qualitative levels of congestion, and the impact of changing traffic light strategies is modelled accordingly. Also scheduling-based approaches [19] have been designed to deal with the problem of controlling traffic lights: in SURTRAC each junction is controlled by an agent that can exchange information with neighbours to coordinate the traffic light strategies.

The generation of routes for vehicles in a road network has been considered by the perspective of routing emergency vehicles [20], or by considering small urban areas and model the movement of vehicles to respect air quality limitations [21]. In both cases, microscopic representations are used, and the scalability of the approaches is limited.

Finally, Lu *et al.* [6] introduce an approach based on automated planning for producing predictable management strategies of UTC.

## VI. CONCLUSIONS

The advent of CAVs is paving the way to smart and sophisticated approaches for traffic control and management. Inspired by the successful applications of automated planning to urban traffic control problems, in this paper we introduced an approach that leverages the expressive power of hybrid PDDL+ planning for performing dynamic traffic assignment, by calculating in real-time the routes to be followed by vehicles approaching a controlled urban region. The PDDL+ formalism provides the ideal ground for simulating the likely evolution of traffic conditions in the network, while planning the routes to be followed by incoming vehicles. An experimental analysis, performed on real-world data collected for the Milton Keynes area, demonstrated the benefit of the proposed approach.

Future work will focus on the application of the proposed technique to different urban regions, and will consider the path to deployment of this framework. While standardisation efforts are well underway for the basic communications layer, roll-out will require coordination of both public traffic authorities and vehicle manufacturers.

## REFERENCES


[1] K.-H. Chang, P.-Y. Hsu, C.-J. Lin, C.-L. Lin, S.-H. H. Juo, and C.-L. Liang, "Traffic-related air pollutants increase the risk for age-related macular degeneration," *Journal of Investigative Medicine*, vol. 67, no. 7, pp. 1076–1081, 2019.

[2] 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.

[3] F. Arena and G. Pau, "An overview of vehicular communications," *Future Internet*, vol. 11, no. 2, p. 27, 2019.

[4] Y. Jiang, S. Wong, H. Ho, P. Zhang, R. Liu, and A. Sumalee, "A dynamic traffic assignment model for a continuum transportation system," *Transportation Research Part B: Methodological*, vol. 45, no. 2, pp. 343–363, 2011.

[5] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[6] X. Lu, N. Zhang, C. Tian, B. Yu, and Z. Duan, "A knowledge-based temporal planning approach for urban traffic control," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] G. Antoniou, S. Batsakis, J. Davies, A. Duke, T. L. McCluskey, E. Peytchev, I. Tachmazidis, and M. Vallati, "Enabling the use of a planning agent for urban traffic management via enriched and integrated urban data," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 284 – 297, 2019.

[8] M. Vallati, D. Magazzeni, B. D. Schutter, L. Chrpa, and T. L. McCluskey, "Efficient macroscopic urban traffic models for reducing congestion: A PDDL+ planning approach," in *Proceedings of AAAI*, 2016, pp. 3188–3194.

[9] M. Fox and D. Long, "Modelling mixed discrete-continuous domains for planning," *J. Artif. Intell. Res.*, vol. 27, pp. 235–297, 2006.

[10] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, 2012.

[11] 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.

[12] O. Derbel, T. Péter, B. Mourllion, and M. Basset, "Generalized velocity–density model based on microscopic traffic simulation," *Transport*, vol. 33, no. 2, pp. 489–501, 2018.

[13] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "Alternative routes in road networks," *Journal of Experimental Algorithmics (JEA)*, vol. 18, 2013.

[14] E. Scala, P. Haslum, S. Thiébaux, and M. Ramírez, "Interval-based relaxation for general numeric planning," in *Proceedings of ECAI*, 2016, pp. 655–663.

[15] E. Scala, P. Haslum, and S. Thiébaux, "Heuristics for numeric planning via subgoaling," in *Proceedings of IJCAI*, 2016, pp. 3228–3234.

[16] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence*, vol. 129, no. 1-2, pp. 5–33, 2001.

[17] A. Pozanco, S. Fernández, and D. Borrajo, "On-line modelling and planning for urban traffic control," *Expert Systems*.

[18] M. Gulić, R. Olivares, and D. Borrajo, "Using automated planning for traffic signals control," *PROMET-Traffic&Transportation*, vol. 28, no. 4, pp. 383–391, 2016.

[19] X.-F. Xie, S. Smith, and G. Barlow, "Schedule-driven coordination for real-time traffic network control," in *Proceedings of ICAPS*, 2012.

[20] L. Chrpa and M. Vallati, "On the exploitation of automated planning for efficient decision making in road traffic accident management," in *55th IEEE Conference on Decision and Control, CDC 2016, Las Vegas, NV, USA, December 12-14, 2016*, 2016, pp. 6607–6612.

[21] L. Chrpa, D. Magazzeni, K. McCabe, T. L. McCluskey, and M. Vallati, "Automated planning for urban traffic control: Strategic vehicle routing to respect air quality limitations," *Intelligenza Artificiale*, vol. 10, no. 2, pp. 113–128, 2016.