Vallati, Mauro, Magazzeni, Daniele, De Schutter, Bart, Chrpa, Lukáš and McCluskey, T.L.

Efficient Macroscopic Urban Traffic Models for Reducing Congestion: a PDDL+ Planning Approach

Original Citation

Vallati, Mauro, Magazzeni, Daniele, De Schutter, Bart, Chrpa, Lukáš and McCluskey, T.L. (2016) Efficient Macroscopic Urban Traffic Models for Reducing Congestion: a PDDL+ Planning Approach. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16). AAAI Press, Phoenix, Arizona USA, pp. 3188-3194.

Efficient Macroscopic Urban Traffic Models for Reducing Congestion: a PDDL+ Planning Approach

Mauro Vallati
School of Computing and Engineering
University of Huddersfield, UK

Daniele Magazzeni
Department of Informatics
King's College London, UK

Bart De Schutter
Delft Center for Systems and Control
Delft University of Technology, Netherlands

Lukáš Chrpa and **Thomas L. McCluskey**
School of Computing and Engineering
University of Huddersfield, UK

Abstract

The global growth in urbanisation increases the demand for services including road transport infrastructure, presenting challenges in terms of mobility. In this scenario, optimising the exploitation of urban road networks is a pivotal challenge. Existing urban traffic control approaches, based on complex mathematical models, can effectively deal with planned-ahead events, but are not able to cope with unexpected situations –such as roads blocked due to car accidents or weather-related events– because of their huge computational requirements. Therefore, such unexpected situations are mainly dealt with manually, or by exploiting pre-computed policies.

Our goal is to show the feasibility of using mixed discrete-continuous planning to deal with unexpected circumstances in urban traffic control. We present a PDDL+ formulation of urban traffic control, where continuous processes are used to model flows of cars, and show how planning can be used to efficiently reduce congestion of specified roads by controlling traffic light green phases. We present simulation results on two networks (one of them considers Manchester city centre) that demonstrate the effectiveness of the approach, compared with fixed-time and reactive techniques.

Introduction

In the 21st Century the world's population is expected to increase –from 5.9bn in 2013 to 9.6bn by 2100– and to become more urbanised. This huge growth increases the demand for housing, as well as associated utilities and services including transport infrastructure, vehicle parking and public transport.

Current urban traffic control (UTC) techniques help to minimise delay within day to day traffic flows, by providing strategies for traffic light phases, and are very effective in planned or typical conditions. UTC methods follow two main directions: model-based predictive approaches (Papageorgiou et al. 2007) and reactive approaches. The former are based on complex mathematical models, amounting to the solution of thousands of equations. Conversely, reactive approaches –such as SCOOT (Bretherton 1989) or SCATS (Lowrie 1982)– are based on simple models of traffic within a small cluster of traffic lights, and thus are faster and can adapt to changes of traffic flows in their local area. Both approaches are not designed to work adequately in the face of unplanned exceptional events, such as when roads have been blocked due to car accidents or weather-related events. Model-based predictive approaches are computationally expensive and usually slow to converge, thus requiring up to several hours for providing a strategy to be applied. Conversely, reactive models are extremely quick, but they rely on pre-computed knowledge and very simplified traffic models; therefore, the generated strategies are not so effective.

Whereas there has been a long record of the use of AI techniques in road transportation (Various 2007; Miles and Walker 2006), there has been little application of AI Planning and Scheduling techniques to UTC. With an adequate description of the world and in particular of valid control actions, however, we conjecture that centralised automated planning techniques can deal with exceptional events and multi-objective optimisation by generating an effective plan for traffic light changes. Recently, the benefits of such planning-based approaches for supporting traffic control have been argued. For instance, Jimoh et al. (2013) introduced the idea of using automated planning in UTC as a planning aid to be used in exceptional circumstances, e.g. in situations where roads within a network of roads become blocked due to some unanticipated incident. However, they assume that each vehicle can be directed by the planner –thus limiting scalability–, and will follow instructions given. Shah et al. (2013) provided a model for dealing with road traffic accident management. A scheduling approach (called SURTRAC) has been proposed by Xie, Smith, and Barlow (2012). They focused on the exploitation of decentralised scheduling techniques to synchronise a group of traffic light clusters. Each intersection is controlled by a scheduling agent that communicates with neighbours to predict the future traffic demand, and to minimise predicted vehicles waiting time at the traffic signal. The interested reader is referred to a recent survey of planning and scheduling approaches applied to traffic management (Cenamor et al. 2014).

Transport studies often use microscopic models to simulate and validate off-line traffic control techniques (Treiber and Kesting 2013). This entails modeling at the individual vehicle level, where each vehicle is a single element. A similar microscopic representation was used in a planning approach to route vehicles to respect air quality limitations in urban networks (Chrpa et al. 2015), utilising a basic STRIPS representation (Fikes and Nilsson 1972). The exploitation of planning approaches on microscopic models has a number of drawbacks, however, the main one being that it has limited scalability. Also, if centralised planning is used to control individual cars, then this assumes that the position of each vehicle is known by urban traffic authorities, and they can communicate to and control each vehicle's route.

In order to overcome the aforementioned drawbacks we utilise a *macroscopic simulation model* –that models traffic at the flow level rather than at the single vehicles level– (Treiber and Kesting 2013), and encode it using a more expressive language, PDDL+ (Fox and Long 2006). PDDL+ has been used in a number of real-world planning applications, see for example (Fox, Long, and Magazzeni 2012; Della Penna et al. 2010), because it enables the system to reason with an explicit model of continuous processes and temporal constraints. The aim of modern urban traffic controls, and leading research in this area (e.g. Xie, Smith, and Barlow), is to minimise the waiting time of cars at traffic signals in day to day conditions. The aim of our system is more regional and strategic: to deal promptly with congested networks caused by exceptional events, by identifying strategies (i.e. temporal plans specifying changes to traffic light green phases through the network) for reducing the congestion of critical roads, and hence restoring the desired state of the network.

This paper's main contribution is to introduce a planning system that in comparison with traditional urban traffic control strategies within our experiments, deals much more efficiently and effectively with region-wide congestion, where traffic volumes are in excess of 10,000 vehicles during peak hour. To do this, we extend the state-of-the-art PDDL+ planning engine UPMurphi (Della Penna et al. 2009) modularly with a novel UTC-targeted heuristic. A further contribution is in the engineering of the domain model: we introduce a token-based approach which is expressive enough to represent a wide range of traffic signal junction types.

Background

In this section we briefly introduce urban traffic control models and automated planning in mixed discrete-continuous domains.

Urban Traffic Control Models

Initially, fixed-time controls and then reactive controls were used in urban traffic control. In the former, actions are predefined according to some sort of historical information. In the latter, which includes widely used systems like SCOOT and SCATS, proposed after sensors were introduced, strategies are reactive to the given input from sensors and can control a set of connected intersections. More recently, model-based predictive traffic control approaches have been proposed. They rely on complex mathematical models that can describe the traffic dynamic mechanics of a given traffic network. Well-known examples include the model proposed by Dotoli, Fanti, and Meloni (2006) and van den Berg et al. (2004). Traffic models are currently exploited in traffic control for predicting the future traffic states, allowing to implement model-based traffic control strategies and optimisation (Lin et al. 2013).

From a general perspective, there exist three main classes of traffic models: microscopic, macroscopic and mesoscopic models (Hoogendoorn and Bovy 2001). Microscopic models are usually very detailed, and describe every vehicle individually; they are mostly used for modelling small portions of larger networks. Macroscopic models are able to handle larger areas, at the price of a higher level of abstraction: the average behaviour of groups of vehicles (or flows) is considered. Mesoscopic models combine properties of the aforementioned models.

In this work, in order to overcome the limits of previously introduced STRIPS-based planning approaches and to effectively handle unexpected traffic conditions, we focus on macroscopic models. Particularly, for modelling urban traffic in PDDL+ we take inspiration from the well-known "Simplified Model" (S-model) (Lin et al. 2012).

PDDL+ Planning

PDDL+ (Fox and Long 2006) is an extension of the standard planning domain modelling language, PDDL, to model mixed discrete-continuous domains. In addition to instantaneous and durative actions, PDDL+ introduces *continuous processes* and *exogenous events*, that are triggered by changes in the environment. Processes are used to model continuous change, and therefore are well suited in this context to model *flows* of vehicles. An example of process is shown in Figure 2, where the process *FlowGreen* is used to model the flow of vehicles in each intersection when the traffic light is green. Note that the continuous effects

```
(increase (queue ?r2) (* #t (flow ?r1 ?r2 ?i)))
(decrease (queue ?r1) (* #t (flow ?r1 ?r2 ?i)))
```

are used to model transition of vehicles from one road to another at the intersection at a specified flow rate.

Figure 2 also shows an example of event, *MaxGreenReached* which is used to switch tokens at intersections, and it is triggered *as soon as* the green time threshold is reached. The use of PDDL+ proved to be very suitable for modelling the traffic control scenario, as described in the next section.

PDDL+ Formulation

In this section we introduce the model that is the basis of the PDDL+ representation of urban traffic control problem.

A region of the *road network* can be represented by a directed graph, where edges stand for *road sections* and vertices stand for either *intersections, entry or exit points*. Intuitively, vehicles enter the network in entry points, and leave the network from exit points. Each road section has a given maximum *capacity*, i.e. the maximum number of vehicles it can serve, and *congestion* threshold, i.e., the upper bound of the number of vehicles that allows fluent traffic flow. The current number of vehicles of a road section is denoted as a *queue*. Intersections are controlled by traffic lights. Traffic in intersections is distributed by *flow rates* that are defined

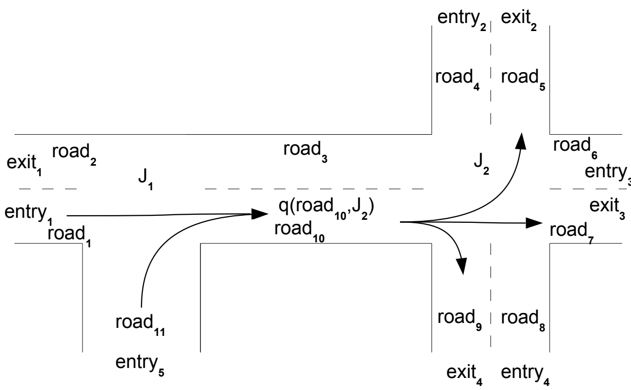

Figure 1: A road network with two intersections (J_1, J_2) and eleven road sections. Arrows indicate how traffic can flow into and out from the road_{10} section.

between each couple of road sections. Given two road sections r_x, r_y, and an intersection i such that r_x is an incoming road section to the intersection i and r_y is an outgoing road section from i. Flow rates stand for the number of vehicles that leave r_x, pass through i and enter r_y per time unit. For the sake of simplicity, we assume that vehicles going in the same direction move into the correct lane, thus not blocking other vehicles going in the different directions.

Figure 1 shows an example of a road network with two intersections (J_1, J_2) and eleven road sections. Arrows show the flows of the incoming and outgoing traffic for the road_{10} section. Vehicles can leave the road_{10} section in three different directions (road_5, road_7 and road_9). If the flow rate between road_{10} and road_7 (through the intersection J_2) is of 2 vehicles per time unit, and 1 vehicles per time unit for the other directions, and the traffic lights are set to green for all the directions, then in one time unit the number of vehicles in the road_{10} section decreases by 4, while the number of vehicles in the road_7 section increases by 2 and in the road_5 and road_9 sections by 1 each.

For controlling traffic lights in intersections we designed a *token*-based approach. Each intersection has an associated token. According to its value, one (or more) traffic lights are allowed to turn green, while the others must be red. Clearly, traffic lights that are turned green by the same token value must not be in conflicting directions. It is worth noting that the designed token-based approach, in conjunction with the use of flows, is very expressive, and allows to describe a wide range of traffic light configurations –from traffic lights controlling specific lanes only, to traffic lights controlling different road sections at the same time– regardless of the number of traffic lights of the intersection. For each intersection, the *minimum* and *maximum* time between a change of value of the token is specified. This range controls, implicitly, the green phase minimum and maximum length for traffic lights of a given intersection.

Intersections are regulated using the following PDDL+ constructs:

- An action *SwitchTrafficSignal(i,t)* is used by the planner for changing token t in intersection i if the minimum green time of t has been reached. This action is the "tool" allowing the planner to affect the traffic flows.
- An event *MaxGreenReached(i,t)* is triggered when token t in intersection i reaches the maximum green time. The event changes token t (in the same way as the *switchTrafficSignal* action does).
- A process *KeepGreen(r,i,t)* is used for "keeping" the traffic light of road section r on intersection i set to green, and measuring the time the green light is on. This process is activated when the token t is given to the aforementioned traffic light, and automatically stops when the green time has reached the maximum allowed value, or the token has been passed by using the *SwitchTrafficSignal* action.
- A process *FlowGreen(r1,r2,i,t)* is activated when the *KeepGreen(r1,i,t)* process is active. It is used for moving vehicles from road *r1* to road *r2* through intersection i at the given flow rate. If there is no vehicle on *r1*, or *r2* is full (i.e., the number of the vehicles is the same as the capacity of *r2*), the *FlowGreen* process is stopped.

```
(:action SwitchTrafficSignal
    :parameters (?i - intersection)
    :precondition
        (>= (greenTime ?i) (minGreenTime ?i))
    :effect (and
        (increase (token ?i) (tokenAdd ?i))
        (assign (greenTime ?i) 0)))

(:process FlowGreen
 :parameters (?r1 ?r2 - road ?i - intersection)
 :precondition (and
       (= (token ?i) (tokenvalue ?r1 ?i))
       (>= (queue ?r1) (flow ?r1 ?r2 ?i))
       (<= (queue ?r2) (- (max_queue ?r2) (flow ?r1 ?r2 ?i))
       (<= (greentime ?i) (maxgreentime ?i)))
:effect (and
       (increase (queue ?r2) (* #t (flow ?r1 ?r2 ?i)))
       (decrease (queue ?r1) (* #t (flow ?r1 ?r2 ?i)))))

(:event MaxGreenReached
 :parameters (?i - intersection)
 :precondition (and
       (>= (greentime ?i) (maxgreentime ?i))
       (< (token ?i) (maxtoken ?i)))
 :effect (and
       (assign (greentime ?i) 0)
       (increase (token ?i) (tokenAdd ?i)))
```

Figure 2: Part of PDDL+ encoding of the UTC domain.

Figure 2 shows the PDDL+ encoding of the *SwitchTrafficSignal* action, the *MaxGreenReached* event and the *FlowGreen* process.

Each entry point of the road network has a corresponding buffer. Vehicles that are going to enter the network are firstly added to the corresponding buffer. As soon as the road section connected to the entry point is not full, an event *releaseCar(r,b)* is triggered for moving vehicles from buffer b to road section r. Situations where vehicles are exiting the road network are handled by considering road sections that lead to exit points to have infinite capacity.

A planning problem is specified by a road network (including road capacities, tokens, minimum and maximum green times, etc.), which captures the static part of the problem, and by the queue length of each road section, initial token values and numbers of vehicles and frequency of their appearance in entry points, which captures the dynamic part of the problem. Timed Initial Literals (Fox and Long 2003) are used to represent situations when vehicles are ready to enter the network later. For example, 35 vehicles are ready to enter the network at the entry point (buffer) $entry1$ at time 5. Then, in PDDL+ we represent it as `(at 5 (= (cars-ready entry1) 35))`.

Given a traffic planning problem, the goal is specified in terms of road sections that are required to be not congested as soon as possible. For instance, given the example shown in Figure 1, it can be required to reduce the congestion of road section $road10$; therefore, the goal would be described as `(< (queue road10) (congested road10))`.

It should be noted that with regards to the typical optimisation target used by traffic light controllers, i.e. minimise average delay of vehicles in the network, the proposed approach provides a wider range of possibilities. For instance, it allows to prioritise flows from specific roads, e.g. known network bottlenecks, or to maximise traffic flows in some directions. Also, a goal in which queues of all the network road sections have to be reduced, corresponds to minimise the average delay.

Forward Heuristic Search for UTC

The UPMurphi planning system has been used for solving urban traffic control problems encoded using the PDDL+ formulation described in the previous section. UPMurphi is a forward search planner that deals with continuous processes using the *Discretise and Validate* approach, where the continuous model is initially discretised, then solved, and finally the found solution is validated against the original continuous model. If the solution is not valid, the discretisation is refined and the process iterates. Among other features, UPMurphi allows one to easily add constraints on the applicability of the actions (in order to prune the state space at the design phase), and to plug in new heuristics, that might be best tailored for a given class of problems. We exploited these two features and designed a specialised forward heuristic search that is well suited for the UTC domain. It can be efficiently used with any UTC problem, i.e., to handle different traffic scenarios and different urban networks.

In this model, the state explosion is caused by the *SwitchTrafficSignal(i,t)* action, that modifies token t at intersection i, and so changes vehicle flows by stopping the green phase of the currently active traffic light(s). The problem specifications already limit the applicability of this action to when token t has not been modified since a *minimumGreenTime* amount of time. Furthermore, for each road section r_j controlled by the traffic lights that will be turned red as the effect of the action, we added the following precondition(s) ($q(r_j)$ and $cap(r_j)$ stand for queue and capacity of r_j respectively):

$$q(r_j) < cap(r_j) \cdot \alpha$$

The purpose of these constraints is to consider stopping flow(s) coming from roads sections only when the corresponding queues are not too long. Similar constraints are considered by commonly used reactive approaches, that take into account the number of vehicles queued at the considered intersection in order to minimise delays. α is used to limit the size of the state space for the planner. Higher values for α give more flexibility to the planner, and possibly lead to find better solutions, at the cost of longer runtimes. We empirically found a good value to be $\alpha = 0.2$, as no significant improvements on the found solutions were observed with greater values.

The heuristic we devised is automatically extracted from the problem description, and it is based on relaxing the constraints that vehicles can leave a road only when the corresponding traffic signal is green. Formally, we have:

$$f(s) = \sum_{r_i \in G} (q_s(r_i)/leave(r_i))$$

where $r_i \in G$ are the road sections specified in the planning task goal (for which we want to reduce the flow under a given threshold), $q_s(r_i)$ is the current queue length on road section r_i and $leave(r_i)$ represents the total flow of vehicles that can leave road section r_i (abstracting from the status of the traffic signals).

Experimental Evaluation

The aim of the experimental evaluation is to test whether the proposed planning-based approach can efficiently and effectively provide plans for minimising the time required to reduce the saturation of critical roads in urban networks. As previously mentioned, the proposed PDDL+ models are designed for planning at a macroscopic level, i.e. the planner deals with flows of vehicles rather than with single vehicles.

In order to assess the usefulness of the proposed planning-based approach, we compared its performance with two different techniques: the well-known *fixed-time* strategy and an *isolated traffic-responsive* strategy (hereinafter, reactive) (Papageorgiou et al. 2003). In the former, that is commonly used as baseline for testing urban traffic control approaches, the green time length of each traffic light is set once and does not change over time. In other words, regardless of the state of the network, green time lengths are fixed. In the reactive approach, each intersection is controlled separately. Given an intersection J_i and an active traffic light controlling the traffic flow from the incoming road section r_m, this technique switches the active traffic light if the $minimal$ green time has been reached and if there is a queue on another incoming road section r_n that is longer than a given threshold. In essence, the reactive approach tries to distribute green time between traffic lights of the same intersection, favouring those that mostly need it in order to quickly shorten their queue. We did not consider *coordinated* reactive approaches, since they rely on a large amount of knowledge extracted from historical data (Papageorgiou et al. 2003). This allows them to effectively deal with typical traffic situations, but that can possibly reduce their ability to cope with unprecedented conditions. It should be noted that we did not compare against a model-based predictive approaches because, as stated in the Introduction, although they are able to globally consider a network, they are computationally expensive, and therefore not suitable to be used in real-time to cope with unexpected events.

UPMurphi –enhanced with the previously described heuristic– has been used for generating plans. The plan-

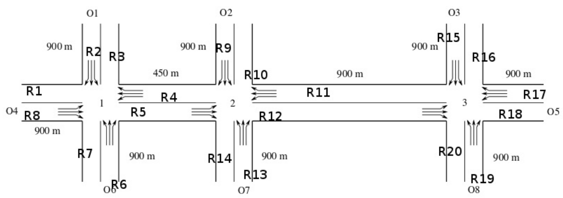

Figure 3: The A scenario, taken from (Lin et al. 2014). This network includes three intersections, eight entry and eight exit points (O1–O8), and twenty road sections (R1–R20).

ner has been run on a system equipped with 2.5 Ghz Intel Core 2 Quad Processors, 4 GB of RAM and Linux operating system. The plans derived by using fixed-time and reactive strategies have been generated by exploiting some specifically developed python scripts: their execution CPU-time is negligible.

The generated plans have firstly been validated using the well-known VAL tool (Howey, Long, and Fox 2004). This was done in order to check their correctness with regards to the designed PDDL+ model, and also to test the presence of flaws in the model. After the first validation, plans have been tested using the traffic simulation tool SUMO (Krajzewicz et al. 2012) (ver 0.23). The exploitation of traffic simulators allows to verify the effectiveness of both plans and models, by actually simulating their execution using realistic physics.

Scenarios

The experimental evaluation focuses on two urban road networks, shown in Figures 3 and 4. The first network (Figure 3) includes three intersections and eight entry/exit points; a similar network has been already used for evaluating a number of model predictive controls for urban traffic, see e.g. (Lin et al. 2014). Each road section has three lanes, corresponding to three different directions vehicles can take. Given a road, one third of the traffic follows each direction. In this scenario we simulated cases in which the main flows of vehicles have to navigate the network from East to West. We considered approximately 600 vehicles over the simulation period. Most of them are already in the network at the initial state; approximately 150 new vehicles progressively enter the network from East and North entry points, as soon as the maximum capacity of roads section allows that. The goal is to unsaturate $R4$, which is the bottleneck of the network due to its length. Hereinafter we will refer to this scenario as scenario A.

The second network is shown in Figure 4. It represents a large section of the Manchester (UK) urban area. Specifically, it simulates what happened in August 2015, when a section of the ring road had to be closed because of a large hole due to heavy rainfall. [1] As a consequence, the traffic that usually flows through the ring road has to pass through part of the city centre, which is already saturated by the usual passing-through traffic. This network has been selected in order to check the scalability of the proposed approach on

[1] www.bbc.co.uk/news/uk-england-manchester-33929490

Figure 4: The considered scenario B. It represents a section of the Manchester urban area. Modelled roads are shown in blue. Part of the ring road (crossed out) is blocked. Incoming traffic flows enter the network from entry points (green) and have to reach exit points (yellow).

large networks that involve a significant number of vehicles. In our experiments we populated this scenario with 11,000 vehicles. 3,000 of them are already in the network, and represent the usual city centre traffic out of peak hours. 8,000 vehicles access the network from the ring road entry points and can leave the network through the two exit points located in the North of the map. The goal is to reduce as quickly as possible the queues on roads connected to entry points, to a few hundreds of cars. In the rest of this section, this scenario will be referred to as scenario B.

It should be noted that our experimental analysis is focused on testing the resilience of the system, by using saturated networks as initial states, i.e. when in every intersection the vehicle queues on (some) roads cannot be dissolved completely at the end of the following green phase. Such situations are the most complex to deal with, and a poor control strategy will quickly lead to very long queues, filled roads sections, and blocked intersections, with a significant impact on traveller delay. Furthermore, scenarios in which networks are not saturated are not so interesting as they can be handled by human traffic experts or by reactive approaches.

Results

In our models, one time step corresponds to approximately five real-world seconds. In scenario A, green time lengths range between 4 and 20 time steps (20 to 100 seconds). In scenario B we increased the range between 5 and 40 time steps, for amplifying the impact of traffic light controllers.

In order to obtain a good overview of the performance of fixed-time and reactive approaches, we considered different configurations. For fixed-time, we consider green phase times ranging from the minimum to the maximum available lengths. The reactive approach strongly depends on the threshold value used for reacting to a "long" queue: such threshold has been expressed as percentage of the road that

Scenario	PDDL+	FixedT	Reactive
A	25	78 – 82	55 – 78
B	3842	5983 – 6281	4650 – 6286
B.1	310	2167 – 2774	1905 – 2766
B.2	686	3631 – 3787	1515 – 3728
B.3	190	1171 – 2017	124 – 1968

Table 1: Quality, in terms of time steps required to achieve a state in which the goal condition is solved, of plans generated by the proposed PDDL+-based approach, a fixed-time technique (FixedT) and a purely reactive approach (Reactive), in the considered scenarios. For the fixed-time (reactive) approach results are shown in terms of minimum–maximum plan quality, according to the different green phases (thresholds) used. One time step corresponds to approximately 5 real-world seconds.

is full. A value of 100% (0%) indicates that the approach reacts only when the road is completely full (empty). In our experiments we run the reactive approach by considering threshold values in the range 0.1%-100.0%, thus testing a wide range of different reactivity levels.

The upper part of Table 1 shows the results of this experimental analysis on scenarios A and B. Results are shown in terms of time steps (maximum and minimum) required by the three approaches for reaching a state in which the goal is satisfied. Remarkably, the proposed approach is able to effectively control traffic light phases in order to minimise the time required to achieve goals. Unsurprisingly, the reactive approach is able to provide better quality as well as a wider range of solutions than the fixed-time approach. Nonetheless, provided plans are still significantly longer than those obtained by exploiting the PDDL+ approach.

In order to investigate how the performance of the considered approaches are affected by different levels of network saturation, we generated three different configurations of the scenario B: B.1, in which incoming traffic flows are reduced by half; B.2, where initially no vehicles are in the network; and B.3, that mixes B.1 and B.2. Unsurprisingly, results shown in Table 1 indicate that in case of very low traffic congestion, reactive approaches can be very effective. On the other hand, the proposed PDDL+ approach allows to generate better quality plans in both B.1 and B.2 scenarios, where the network is not congested yet, but poor traffic light strategies can quickly lead to the saturation of some critical intersections.

The results shown in Table 1 have been validated by using SUMO. The validation confirms that plans provided by the PDDL+ approach allow to reach goals set for scenarios A and B in a significantly shorter time than plans generated by using the other approaches. In a nutshell, SUMO-validated plans tend to take around 2-3 times longer to execute than predicted for all the tested approaches. This is because some elements (e.g., vehicle acceleration and brake time) are not considered in the models.

In urban traffic control applications it is of pivotal importance that effective plans are provided quickly, in order to tackle unexpected issues as soon as possible. The proposed approach is able to solve scenario A in less than 2 CPU-time seconds; scenario B requires at most 20 CPU-time seconds, while B.1–B.3 are solved in less than 5 CPU-time seconds.

Discussion

The experiments demonstrate the extent to which our approach is able to efficiently control traffic lights phases in order to cope with unexpected traffic conditions, and hence demonstrates its potential for increasing the resilience of network management. Even though such phase changes are the main way that UTC can effect traffic flows, it should be noted that the proposed PDDL+ model can be extended in order to enlarge the set of potential UTC interventions modelled. For instance, an extended model can allow the planner to modify the order in which traffic lights are activated around a junction; or it could incorporate a model (utilising historical data) of the effects of using variable-message signs on traffic flows.

One of the main challenges in urban traffic control is to efficiently deal with saturated traffic and/or full networks. As shown in our experimental analysis, fixed-time and reactive approaches can quickly provide plans for underused networks, but they can not effectively handle saturated conditions. For this reason, the proposed approach has been designed for focusing on saturated networks. This also allows to simplify the model for increasing efficiency; for instance in saturated conditions vehicles are mostly moving extremely slowly on average or even stationary, thus very accurate speed models are not needed.

In its current form, the proposed PDDL+ approach operates as an open loop controller: the planner is used to predict and then control the evolution of traffic flows in the network. Nevertheless, given the low runtime, it can be integrated as a component of a closed loop control system by including monitoring. If the state of the network significantly differs from the planner's prediction, then the speed of the planner enables a re-planning approach to be adopted.

The performed experimental analysis demonstrated that the proposed approach can handle up to 11,000 vehicles: this is comparable to one-hour traffic flow of the rush hours traffic of a medium-sized European city. In order to efficiently increase the size of traffic flows involved, a continuous planning approach (Coddington 2002) can be exploited. This direction will be investigated in our future works, in collaboration with our industrial partners.

Finally, it is worth noting that the proposed PDDL+ approach does not require any additional knowledge about traffic dynamics in the considered network. It is capable of generating effective plans with unexpected traffic conditions by considering the structure of the network and the current traffic condition. This extends its applicability to urban areas where sensors have been only recently introduced, and where other approaches would show low performance. Nonetheless, whether available, historical data knowledge can be easily encoded.

Conclusion

In this paper we proposed a PDDL+ encoding of the urban traffic control problem, where traffic green phases can be controlled by a planner for reducing network congestion, and we designed a forward heuristic search in order to improve the performance of existing state-of-the-art planners. The performed experimental analysis shows that the proposed approach can effectively cope with unexpected traffic conditions. For the future, we propose to further validate our solution in collaboration with our industrial partners. We are also interested in extending the PDDL+ model for considering other traffic control actions, such as variable-message signs for route guidance or variable speed limits.

References

Bretherton, R. 1989. Scoot urban traffic control system: Philosophy and evaluation. In *Proceedings of the IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation*, 237–239.

Cenamor, I.; Chrpa, L.; Jimoh, F.; McCluskey, T. L.; and Vallati, M. 2014. Planning & scheduling applications in urban traffic management. In *Proceedings of the Annual Workshop of UK Planning & Scheduling Special Interest Group*.

Chrpa, L.; Magazzeni, D.; McCabe, K.; McCluskey, T.; and Vallati, M. 2015. Automated planning for urban traffic control: Strategic vehicle routing to respect air quality limitations. In *Italian Workshop on Planning and Scheduling*.

Coddington, A. 2002. A continuous planning framework with durative actions. In *Proceedings of the IEEE International Symposium on Temporal Representation and Reasoning*, 108–114.

Della Penna, G.; Magazzeni, D.; Mercorio, F.; and Intrigila, B. 2009. UPMurphi: A tool for universal planning on PDDL+ problems. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 106–113.

Della Penna, G.; Intrigila, B.; Magazzeni, D.; and Mercorio, F. 2010. A PDDL+ benchmark problem: The batch chemical plant. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 222–225.

Dotoli, M.; Fanti, M. P.; and Meloni, C. 2006. A signal timing plan formulation for urban traffic control. *Control Engineering Practice* 14(11):1297–1311.

Fikes, R. E., and Nilsson, N. J. 1972. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2(3):189–208.

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

Fox, M., and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *J. Artif. Intell. Res.(JAIR)* 27:235–297.

Fox, M.; Long, D.; and Magazzeni, D. 2012. Plan-based policies for efficient multiple battery load management. *J. Artif. Intell. Res. (JAIR)* 44:335–382.

Hoogendoorn, S. P., and Bovy, P. H. 2001. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 215(4):283–303.

Howey, R.; Long, D.; and Fox, M. 2004. VAL: automatic plan validation, continuous effects and mixed initiative planning using PDDL. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, 294–301.

Jimoh, F.; Chrpa, L.; McCluskey, T.; and Shah, M. M. S. 2013. Towards application of automated planning in urban traffic control. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems*. 985–990.

Krajzewicz, D.; Erdmann, J.; Behrisch, M.; and Bieker, L. 2012. Recent development and applications of SUMO–simulation of urban mobility. *International Journal On Advances in Systems and Measurements* 5(3&4):128–138.

Lin, S.; De Schutter, B.; Xi, Y.; and Hellendoorn, J. 2012. Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C* 24:122–140.

Lin, S.; De Schutter, B.; Xi, Y.; and Hellendoorn, H. 2013. Integrated urban traffic control for the reduction of travel delays and emissions. *IEEE Transactions on Intelligent Transportation Systems* 14(4):1609–1619.

Lin, S.; De Schutter, B.; Hegyi, A.; Xi, Y.; and Hellendoorn, H. 2014. On a spatiotemporally discrete urban traffic model. *Intelligent Transport Systems, IET* 8(3):219–231.

Lowrie, P. 1982. The sydney coordinated adaptive traffic system-principles, methodology, algorithms. In *Proceedings of the International Conference on Road Traffic Signalling*, number 207.

Miles, J. C., and Walker, A. J. 2006. The potential application of artificial intelligence in transport. *Journal of Intelligent Transport Systems* 153(3):183–198.

Papageorgiou, M.; Diakaki, C.; Dinopoulou, V.; Kotsialos, A.; and Wang, Y. 2003. Review of road traffic control strategies. *Proceedings of the IEEE* 91(12):2043–2067.

Papageorgiou, M.; Ben-Akiva, M.; Bottom, J.; Bovy, P.; Hoogendoorn, S.; Hounsell, N.; Kotsialos, A.; and McDonald, M. 2007. *Handbooks in Operations Research and Management Science: Volume 14, Transportation*. Elsevier Science & Technology.

Shah, M. M.; Chrpa, L.; Kitchin, D.; McCluskey, T. L.; and Vallati, M. 2013. Exploring knowledge engineering strategies in designing and modelling a road traffic accident management domain. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2373–2379.

Treiber, M., and Kesting, A. 2013. Traffic flow dynamics. *Springer*.

van den Berg, M.; De Schutter, B.; Hegyi, A.; and Hellendoorn, J. 2004. Model predictive control for mixed urban and freeway networks. In *Proceedings of the Annual Meeting of the Transportation Research Board*, volume 19.

Various. 2007. Artificial intelligence in transportation. *Transportation Research Circular E-C113, Transport Research Board*.

Xie, X.-F.; Smith, S.; and Barlow, G. 2012. Schedule-driven coordination for real-time trafc network control. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 323–331.