An Innovative Heuristic for Planning-Based Urban Traffic Control

Santiago Franco(✉), Alan Lindsay, Mauro Vallati, and Thomas Lee McCluskey

School of Computing and Engineering, University of Huddersfield, Huddersfield, UK
{s.franco,a.lindsay,m.vallati,t.l.mccluskey}@hud.ac.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 network is a pivotal challenge, particularly in the case of unexpected situations. In order to tackle this challenge, approaches based on mixed discrete-continuous planning have been recently proposed and although their feasibility has been demonstrated, there is a lack of informative heuristics for this class of applications. Therefore, existing approaches tend to provide low-quality solutions, leading to a limited impact of generated plans on the actual urban infrastructure.

In this work, we introduce the Time-Based heuristic: a highly informative heuristic for PDDL+ planning-based urban traffic control. The heuristic, which has an admissible and an inadmissible variant, has been evaluated considering scenarios that use real-world data.

Keywords: Urban traffic control · Automated planning
Heuristic for planning

1 Introduction

It is expected that, during the 21st century, there will be a huge growth in urbanisation. In 2014, 54% of the global population were living in urban areas, and this is projected to rise to 66% by 2050. This increase in urbanisation, coupled with the socio-economic motivation for increasing mobility, is going to push the transport infrastructure well beyond its current capacity. In response, more stringent and intelligent control mechanisms are required to better monitor, exploit, and react to unforeseen conditions.

Urban Traffic Control (UTC) is normally the responsibility of local authorities whose aims include reducing congestion, improving journey times, increasing the reliability of the road network, safety regulation compliance and traffic pollution limitation. Conventional UTC techniques are widely deployed in urban areas, and help to minimise delay within day to day traffic flows, by providing strategies for traffic light phases. This is the case of traffic-responsive systems like SCOOT [8] and SCATS [1], fixed time light strategies optimised using historical data, or model-based predictive controllers [5]. These approaches work

Y. Shi et al. (Eds.): ICCS 2018, LNCS 10860, pp. 181–193, 2018.
https://doi.org/10.1007/978-3-319-93698-7_14

reasonably well in normal or expected conditions, but are not designed to work adequately in the face of unexpected or unplanned events. In these cases Transport Operators may struggle to find a strategy tailored to solve the unexpected situation. Creating such strategies is a manual task that may take several days or weeks, and is therefore infeasible to be done in real-time.

Recently, in order to overcome the aforementioned issues of conventional UTC techniques, the application of AI Planning to help in the management of road traffic has been investigated. Works like [4,9] have shown the feasibility of applying planning to deal with unexpected circumstances in urban traffic control, by optimising the length of traffic signal phases in the controlled region in order to achieve some specified high level goals. These works also highlighted that the representation of vehicles through the urban network needs to be performed at a macroscopic level – i.e., no explicit representation of each single vehicle – to cope with large volumes of traffic. The main choice is then between using PDDL and discretising the traffic density (using a sequence of density descriptors) on road sections, as done in [4], or use a numeric representation of traffic density and explicit continuous flow processes by encoding using the PDDL+ [3] language, as done in [9] and, more recently, in [6]. An advantage in using PDDL+ resides in its accuracy, i.e. the representation contains exact counts of vehicles, and models continuous change of vehicle numbers on road sections according to traffic light phases. A very significant drawback is that the few available domain-independent heuristics fail because of the high complexity of mixed discrete/continuous planning, and the size of region-wide urban networks.

In this paper, we introduce the innovative Time-Based heuristic, a domain-specific heuristic designed for improving the performance of a PDDL+ planning-based urban traffic solution. The Time-Based heuristic considers each road section which is present in the planning task goal in isolation, and performs an analysis of the expected input/output traffic flows in order to estimate the distance from a goal state. During the design, emphasis has been given to reducing the computational complexity by pre-calculating –in a pre-processing phase– most of the information needed. The Time-Based heuristic has two variants: an admissible one, which can be fruitfully exploited also by optimal PDDL+ planning engines, and an inadmissible version, which instead focuses on maximising the informativeness of the heuristic value at the cost of the general admissibility. The experimental analysis, that considers a region of the Manchester (UK) city centre and different challenging scenarios using real-world data, demonstrates the beneficial impact of the Time-Based heuristic on the performance of the state-of-the-art PDDL+ planning engine UPMurphi [2], and shows that the Time-Based heuristic outperforms the state of the art of domain-specific heuristics in terms of quality of generated plans.

2 PDDL+ Model for Region-Wide Traffic Control

PDDL+ [3] is an extension of the standard planning domain modelling language, PDDL, to model mixed discrete-continuous domains. In addition to instanta-

neous and durative actions, PDDL+ introduces *continuous processes* and *exogenous events*, that are triggered by changes in the environment. In 2016, Vallati et al. [9] proposed the first PDDL+ model for region-wide traffic control. The model was subsequently re-engineered by McCluskey and Vallati [6], in order to improve scalability and to provide a more accurate representation of the involved constraints. In the remainder of this paper, we will refer to the 2017 model.

A region of the urban *road network* is represented by a directed graph, where edges stand for *road sections* and vertices stand for *intersections*. One vertex is used for representing the *outside* of the modelled region. Intuitively, vehicles enter (leave) the network from road sections connected with the outside. Each road section has a given maximum *occupancy*, i.e. the maximum number of vehicles that can be, at the same time, in the road, and the current number of vehicles of a road section, which is denoted as the current occupancy.

Traffic in intersections is distributed by *flow rates* that are defined between each pair of road links. Given two road sections r_x, r_y, an intersection i, and a traffic signal phase p such that r_x is an incoming road section to the intersection i, r_y is an outgoing road section from i, and the flow is active (i.e., has green light) during phase p. Flow rates stand for the number of vehicles – expressed in terms of *Passenger Car Unit* (PCU) – that can leave r_x, pass through i and enter r_y per time unit.

A road section connected with the outside area can either have incoming or outgoing flows of vehicles. In the first case, vehicles from the outside region are entering the modelled area through the section, otherwise the road section is used by vehicles that are leaving the modelled area. Each road section connected with the outside has a corresponding *entering* (*leaving*) rate, that indicates the maximum flows of vehicles, in either direction, that can be served by the section.

Intersections are described in terms of a sequence of traffic signal phases. Specifically, intersections *contain* signal phases, which are connected using a *next* predicate. According to the active traffic light phase, one (or more) flow rates are activated, corresponding to the traffic lights that are turned green. For each phase, the *minimum* and *maximum* phase length is specified. Within this range, the planner can decide whether to stop the phase currently active, or not. Between two subsequent signal phases, an *intergreen* interval is specified. Intergreens are (usually) short periods of time designed to allow vehicles that are stacked in the middle of the junction to leave, and pedestrian crossing time, before the next phase is started. Intergreens have a fixed length, which cannot be modified by the planner (or by traffic controllers).

Processes are used for modelling the continuous flow of vehicles through a junction, and for measuring the time phases and intergreens are kept on. Limits and boundaries are controlled by specifically designed PDDL+ events. The planner can influence the behaviour of the network, and actually perform traffic control, by using the *switchPhase(p, i)* action, shown in Fig. 1. This action can be used for stopping the currently active phase p in intersection i, if the intersection i is controllable, and minimum phase time of p (increased by the corresponding process) has been reached.

```
(:action switchPhase
    :parameters (?p - phase ?i - intersection)
    :precondition (and
        (controllable ?i)
        (activePhase ?p)
        (contains ?i ?p)
        (> (phaseTime ?i) (minPhaseTime ?p) ))
    :effect (and
        (trigger ?i) ))
```

Fig. 1. The PDDL+ model of the only action under the control of the planning engine: *switchPhase*, used for stopping the currently active phase $?p$ in intersection $?i$.

Given a traffic planning problem, traffic operators are concerned about the degree of saturation of road sections –in other words, the closeness of the current number of vehicles travelling along a section to its capacity. The degree of saturation determines, for example, whether or not traffic can flow at the maximum allowed speed limit –if it is too high, this results in "stop-start" conditions. Hence, the goal of operator interventions during an exceptional or emergency event would be to de-saturate the surrounding roads in an efficient manner. This immediately translates to goals specified in terms of required occupancy of road sections (since capacities are well known), e.g., road section, r_x, should have an occupancy of less than 50 PCU.

2.1 Existing Heuristics for Planning-Based UTC

A domain-specific heuristic for discrete-continuous planning-based UTC, the queue-based heuristic, was introduced in [9]. Such a heuristic is based on relaxing the constraints that vehicles can leave a road only when the corresponding traffic signal is green. More formally:

$$h(s) = \sum_{r_i \in G} (O_c(r_i)/leave(r_i))$$

where $r_i \in G$ are the road sections specified in the planning task goal, $O_c(r_x)$ is the current occupancy of road section r_x, and $leave(r_x)$ represents the total flow of vehicles that can leave road section r_x, obtained by summing all the outgoing flows over all the traffic signal phases (abstracting from the status of the traffic signals).

The queue-based heuristic is obtained by summing the heuristic value of each road section in the goal. It is not admissible because it does not consider the possibility that two (or more) road sections can have outgoing flows of vehicles active at the same time.

3 The Time-Based Heuristic

The proposed heuristic is designed to be used by a forward search planning engine, that deals with continuous processes via discretisation.

The Time-Based heuristic considers each road section r_i specified in the planning task goal in isolation. For each r_i, the Algorithm 1 is invoked for assessing the heuristic distance h_{r_i}, expressed in terms of number of discretised time steps, from a state in which the goal is satisfied. Computed heuristic values are then combined as follows:

$$h(s) = \max_{r_i \in G} (h_{r_i})$$

where $r_i \in G$ are the road sections specified in the planning task goal, and h_{r_x} is the heuristic distance from a goal state for the r_x road section, computed in isolation.

In order to compute the heuristic value of a goal road section r_i efficiently, a pre-processing step is needed. In the pre-processing step, the sequence of phases for maximising the outgoing traffic flows from r_i, called P^o, is calculated as follows. We consider the sequence $P = \langle p1, \ldots, pm \rangle$ of traffic signal phases of the intersection x, that receives the outgoing traffic flows of road section r_i. Each phase pn carries information about the minimum and maximum green time, and the maximum outflow traffic that the phase enables from r_i. The initial P^o is the sequence where all the traffic signal phases of the intersection are set to the minimum green time length. Then, the length of the phase(s) with the highest outgoing traffic flow from the road section in object r_i is maximised, according to the maximum allowed value specified in the model. After that, iteratively:

- (i) calculate the average outgoing flow from r_i, called a_{ri}, of P^o.
- (ii) considering the phases that are not already maximised $pl, \ldots, py$: the phase pn with the highest outgoing flow from r_i is selected;
- (iii) the green time length of pn is maximised if its outgoing flow from r_i, per time-step, is higher than the average a_{ri}.

The cycle terminates when the length of all the traffic light phases have been maximised in P^o, or there is no phase within P^o with an outgoing flow higher than the current a_{ri}. This leaves us with the final value of P^o.

In a nutshell, the underlying idea is to optimise the sequence of phases following a "common sense" solution that would have been applied by human controllers. This is done by applying the described hill-climbing approach, that divides phases into "good" and "bad". Good phases get the maximum possible green time, as they provide a significant outgoing flow from the road section in consideration; bad phases instead are minimised, in order to reduce the time spent between good phases.

Intergreen intervals are taken into account in P^o and considered during the computation of the heuristic value, in Algorithm 1. They were not mentioned in the explanation above, for the sake of readability. Although the average outgoing flow is maximal for the considered road section, r_i, the pre-computation step can overestimate the time needed for reaching the goal for r_i. This can be corrected by considering alternatives to P^o in the final sequence of phases, which we will describe below.

Algorithm 1 shows how the heuristic value of a road section of the planning task goal r_i is computed. The core of the procedure is the while loop (lines 3–16)

Algorithm 1. The procedure for assessing the admissible version of the Time-Based Heuristic for a road section r_i which is listed in the planning task goal. Input of the procedure are: p_c^o, current active traffic light phase for the outgoing flow from r_i; O_c, current occupancy of the road section; O_g goal required occupancy for the road section; P^o, optimised sequence of phases for maximising the outgoing flows; and Δ, the discretisation step.

Input: p_c^o,O_c,O_g,P^o,Δ
Output: h
1: $h = 0$
2: j = position(p_c^o, P^o) ▷ Initial phase set for Outgoing flows
3: **while** $O_c > O_g$ **do**
4: **if** phase_at(j) not maximised in P^o
5: f = potential_flow_before_maximised(phase_at(j),P^o)
6: **if** $(O_c - O_g) \leq f$
7: $\langle h', O_c' \rangle$ = try_optimise(P^o,O_c,O_g,h,j)
8: **if** $O_c' == O_g$
9: **return** h'
10: **end if**
11: **end if**
12: **end if**
13: $O_c = O_c -$ flow(P^o,j,Δ)
14: $j = j + \Delta$
15: $h = h + \Delta$
16: **end while**
17: **return** h

where, considering the optimised sequence of phases P^o, the occupancy of the section r_i is updated for each discretisation step. The general case is described in lines 13–15. Lines 4–12 are designed to tackle the last steps of the heuristic evaluation, where the use of the optimised sequence of phases may not lead to the best possible solution, thus making the heuristic value inadmissible.

Let us use an example for explaining under which circumstances this may happen. We assume that the considered intersection, from which vehicles can leave the road section r_i, has four traffic signal phases: $\langle p1, p2, p3, p4 \rangle$. In this example we ignore intergreens for readability, but the same reasoning would have applied in the presence of intergreens. $p1$ has an outgoing flow from r_i of 5 PCUs per time step, $p2$ has an outgoing flow of 1 PCU per time step, while no vehicles can leave r_i when phases $p3$ or $p4$ are active. For the sake of simplicity, we can assume that each phase has a minimum length of 1 time step and a maximum length of 5 time steps. The optimised sequence of phases calculated during pre-processing would be $P^o = \langle p1(0\text{–}4), p2(5), p3(6), p4(7) \rangle$: $p1$ is active for 5 time steps (0–4) and each of the other phases is active for one time step. This cycle then repeats. Let us now assume that, during the heuristic evaluation, the current occupancy O_c of the considered road section is of 2 PCUs, the goal is to have the road section completely empty, and phase $p1$ has just terminated. By using the optimised sequence of phases, the goal would be 4 time steps away:

one PCU leaves the road section during $p2$, then $p3$ and $p4$ are active for one time step each (but no vehicles leave r_s), and finally the remaining PCU leaves the road section in the first time step of $p1$. However, by extending the length of $p2$, the goal could have been reached in 2 time steps, instead of 4.

Generalising from the described example, the use of P^o may prevent the shortest heuristic distance from the goal being found in the cases in which there is a sequence of bad phases, and the remaining number of PCUs in the road section can be cleared by extending the length of one (or more) of them, before the start of the subsequent good phase(s). Lines 4–12 of Algorithm 1 are dedicated to handle these cases.

3.1 Admissibility

In order to demonstrate the admissibility of the Time-Based heuristic, we have to focus on the three aspects which are involved in the computation of the heuristic distance from the goal of a given r_i: current occupancy, outflows, and inflows. Each of them must not lead to an overestimation of the distance from the closest state in which the goal is satisfied. The admissibility of the Time-Based heuristic is always guaranteed because:

- the current occupancy is provided as input to Algorithm 1, and is then updated according to the outflows and inflows as follows;
- inflows are relaxed: it is assumed that no incoming flows of vehicles are activated for the considered road section r_i;
- vehicles can always leave r_i if an appropriate traffic light phase is active, regardless of the congestion of the subsequent road sections;
- the use of the optimised phase sequence P^o, in conjunction with the control previously described, can provide an accurate estimation of the distance from the goal, but it does not overestimate the distance.

Finally, the heuristic evaluation of a state is done by considering only the maximum heuristic value among the heuristic values of road sections included in the planning task goal. In this way, any possible overestimation due to the combination of heuristic values is avoided.

3.2 An Inadmissible Variant of the Time-Based Heuristic

Relaxing the problem by assuming incoming flows to the road section r_i are zero is important in guaranteeing the admissibility of the heuristic. As the road section is considered in isolation r_i, with no information about the surrounding network, it may be the case that some expected traffic flows are not "available", for instance because a road section is empty. However, assuming incoming traffic flows always exist can usually lead to a more accurate evaluation of the distance from the goal compared to ignoring them completely. For this reason, we devised an inadmissible version of the Time-Based heuristic, that is presented in Algorithm 2. Beside P^o, in the pre-processing step of the inadmissible heuristic it is

Algorithm 2. The procedure for calculating the inadmissible version of the Time-Based Heuristic for a road section r_i which is listed in the planning task goal. Input of the procedure are: p_c^o, current active traffic light phase for the outgoing flow from r_i; p_c^i, current active traffic light phase for the upstream intersection; O_c, current occupancy of the road section; O_g goal required occupancy for the road section; P^o, optimised sequence of phases for maximising the outgoing flows; P^i, optimised sequence of phases for minimising the incoming flows to r_i; S, is the list of road sections receiving traffic flows from r_i, and their current occupancies; and Δ, the discretisation step.

Input: p_c^o,p_c^i,O_c,O_g,P^o,P^i,S^o,Δ
Output: h
1: $h = 0$
2: i = position(p_c^i, P^i) ▷ Initial phase set for Incoming flows
3: j = position(p_c^o, P^o) ▷ Initial phase set for Outgoing flows
4: **while** $O_c > O_g$ **do**
5: $O_c = O_c -$ flow(P^o,j,Δ) + flow(P^i,i,Δ)
6: $h = h + \Delta$+potential_delay(S)
7: $i = i + \Delta$
8: $j = j + \Delta$
9: **end while**
10: **return** h

also required to compute P^i, which is an optimised sequence of phases for the intersection that has incoming flows to r_i. As the goal is to de-congest as soon as possible r_i, P^i is optimised in order to *minimise* the incoming flow to r_i, following the dual approach of the one previously described. Phases reducing the average incoming traffic flow are given the maximum green time, while others are given the minimum green time. This optimisation encodes the domain knowledge of a human expert that, for reducing the congestion on a given road section, minimises the incoming traffic to that section by reducing the corresponding green times.

The core of the procedure for computing the inadmissible heuristic resides in lines 5 and 6, where the occupancy of road section r_i is updated according to the expected incoming and outgoing flows in the considered time step, and the heuristic distance from the goal is updated. The calculation in line 5 of Algorithm 2 is reminiscent of the kind of conservation equation that a model predictive control approach would entail, en route to deriving the solution for a matrix of intersections for the region [5]. The *potential_delay* method deals with a very important aspect of traffic flows. Outgoing vehicles from r_i are either leaving the controlled region, or entering subsequent road sections. If the receiving sections are full or heavily congested, then some delay in the flow of vehicles has to be taken into account. In our implementation, the potential delay is assessed by computing the queue-based heuristic of each road section r_j that receives traffic flows from r_i. The queue-based heuristic is then multiplied by the ratio of traffic of r_i that r_j receives over a cycle of the optimised P^o

traffic signal phases. For each iteration of the loop (lines 4–9 of Algorithm 2), the current occupancy of receiving road sections is updated by considering the outflows from r_i. Taking into account the potential delay can greatly improve the accuracy of the heuristic evaluation but has two main drawbacks: since it relies on the queue-based heuristic, the admissibility can not be guaranteed, and –due to the additional calculations– the complexity is increased.

In Algorithm 1, lines 4–12 a forward search was made in attempt to find alternatives to P^o. This can allow the goal to be discovered early and is therefore necessary for admissibility. However, as admissibility cannot be guaranteed in Algorithm 2, this part has been omitted in the interest of performance.

4 Experimental Evaluation

In this section we evaluate the performance of the introduced heuristics. For our experimental evaluation, we consider the urban network presented in [6]. The modelled region is shown in Fig. 2, and represents an area of the Manchester (UK) urban network. This urban network allows to design scenarios which are the most challenging currently available for PDDL+-based urban traffic control, and that are based on real data. The region is considered already congested with the typical morning peak hour traffic, that is derived from historical data. The region includes 15 junctions and 34 road links: 7 junctions are controllable junctions (in red) and the 8 outer junctions are not modelled as controllable, but act as a boundary to the region. Each controllable junction has between 2 and 7 traffic light phases. For this experimental analysis we considered three scenarios, which have been crafted by traffic experts from Manchester; they provided the required data and validated the strategies generated by the planning approach. **Scenario A** simulates an extreme vehicle build upon a road section entering into the controlled region. The scenario focuses on clearing the road section as soon as possible. It is formalised by assuming the road section connecting intersection 1202 (Fig. 2) and the southernmost entry point of the region contains at the initial state an unexpectedly large number of vehicles (in this case, 300), and the goal state is to reduce the number to less than 10. The focus of **Scenario B** is to clear congestion from 3 road links leading into the junctions 1867, 1349 and 1202 shown in Fig. 2, where an extra 600 vehicles are entering as a result of a disturbance in another region. Finally, **Scenario C** simulates cases where a large number of vehicles have to leave a specific area of the controlled region in a short time horizon, like in the case of sport or cultural events where vehicles are rapidly emerging from car parks. For this scenario we considered an extra 200 vehicles on the road section heading from intersection 1349 to intersection 1867. In our models, one time step corresponds to approximately five real-world seconds.

All results were achieved by running the considered systems on a machine equipped with i7-4750HQ CPU, 16 GBs of memory, running Ubuntu 16.10 OS. A 10 CPU-time minutes cut-off time limit was enforced.

The proposed heuristics have been plugged in the UPMurphi [2] planning framework, compiled with g++ version 4.9. for a 32 bit architecture. Hereinafter,

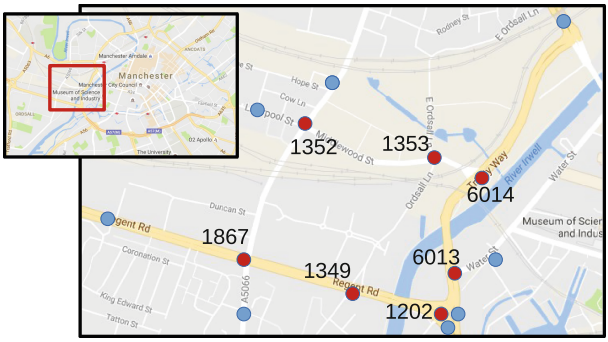

Fig. 2. The Modelled Area (large picture) and the position of the modelled area with regards to the city centre of Manchester, UK (small picture, red-limited area). Blue points indicate the sources (destinations) of incoming (outgoing) vehicles. (Color figure online)

we will use *Ad-Tb* for referring to UPMurphi enhanced with the admissible version of the Time-Based heuristic, and *In-Tb* for referring to the inadmissible version of the proposed heuristic. UPMurphi has been selected due to its ability to handle PDDL+ features, and because it has been used in previous works involving PDDL+ for controlling urban traffic control, as well as other real-world applications. We compare Ad-Tb and In-Tb with UPMurphi extended using the previously introduced *queue-based* heuristic. For the sake of completeness, we also considered UPMurphi with no heuristic and DiNo [7] in this experimental analysis. The former could provide some insights into the performance of non-heuristically guided search, while the latter is a state-of-the-art PDDL+ planner, guided by a domain-independent heuristic. Unfortunately, they did not solve any of the considered benchmarks, and are therefore excluded from the rest of this empirical evaluation.

4.1 Results

The results of the full range of experiments are shown in Table 1. The three scenarios have been tested by considering different initial states in which different traffic light phases are active for the road sections which are in the planning task goal. As a first remark, we observed that the Queue heuristic is very sensitive to this aspect. Specifically, if vehicles can not leave the road section(s) from the initial state, because all possible traffic flows are on red signal, then the queue heuristic is not informative, and UPMurphi is not able to find a solution within the 10 min CPU-time limit. This condition has been named as Queue-R in Table 1. Queue-G shows the performance delivered when traffic lights are initially on green for the considered road section(s). The results indicate that, as expected, the Time-Based heuristic is robust with regards to the traffic light phase that is initially active.

Table 1. Average performance, in terms of plan quality (time needed to reach a goal state), number of visited states during search, and CPU-time, delivered by UPMurphi using the admissible Time-Based heuristic (Ad-Tb), the inadmissible version (In-Tb), and the Queue heuristic. Queue heuristic shows very different performance when the traffic light on the goal road sections is on green (Queue-G) or on red (Queue-R). ATPVS stands for Average Time per Visited State.

	Plan quality	Visited states	Runtime	ATPVS
Scenario A				
Queue-G	350	492	0.5	$10.16 * 10^{-3}$
Queue-R	–	–	–	–
Ad-Tb	350	497	0.5	$10.06 * 10^{-3}$
In-Tb	350	497	0.5	$10.06 * 10^{-3}$
Scenario B				
Queue-G	1710	2343	10.0	$4.27 * 10^{-3}$
Queue-R	–	–	–	–
Ad-Tb	1805	6270	314.1	$50.09 * 10^{-3}$
In-Tb	1360	4687	180.0	$38.40 * 10^{-3}$
Scenario C				
Queue-G	280	1814	5.5	$3.03 * 10^{-3}$
Queue-R	–	–	–	–
Ad-Tb	420	2743	10.5	$3.83 * 10^{-3}$
In-Tb	185	1435	3.3	$2.99 * 10^{-3}$

In Scenario A, Queue-G, Ad-Tb, and In-Tb allow UPMurphi to deliver very similar performance, this is mainly because the goal includes a single road section that is on the border of the controlled region, so the incoming flow of traffic is modelled as continuous in the PDDL+ model and is not explicitly considered by any of the heuristics. Scenarios B and C allows to shed some light into the usefulness and informativeness of the different heuristics. Ad-Tb is usually the slowest, and the quality of provided plans tends to be lower than those of plans found using different heuristics. This is mainly due to the fact that, for the sake of admissibility, useful sources of information can not be considered by the heuristic. In Scenario B, the number of states expanded by In-Tb and Ad-Tb is significantly higher than for Queue-G. Our analysis indicates that the focus on the maximum heuristic value, among values calculated for road sections in the planning task goal, can lead to a jeopardised exploration of the search space, by focusing on the road section that is more distant from its goal. Nevertheless, the In-Tb heuristic outperforms the Queue heuristic in terms of quality of the generated plans. The delivered plan allows to de-congest the road sections 20% faster than when using the plan generated by the Queue-G heuristic.

Regarding scenario C, the In-Tb heuristic finds very quickly a significantly better quality plan than the Queue heuristic (34% better). This is because In-

Tb takes fully into account the dynamics of both the inflows and outflows to the goal's road section. On the other hand, Ad-Tb not only takes significantly longer to find a solution than the queue heuristic, but it is also significantly worse. According to our analysis, this is because its time prediction is over-optimistic, as it does not consider at all the very relevant input flows. Under such conditions, the queue heuristic is then more accurate than the Ad-Tb, as it can find a monotonic path towards a solution.

One would expect the queue heuristic to be faster to compute on average than the Time-Based. The ATPVS data shows the combined average expansion, generation and heuristic evaluation times per visited state. In some cases the Time-Based heuristic can significantly increase the average cost per visited state. Interestingly, the inadmissible version is generally cheaper than its counterpart, this is to be expected as the admissible version requires some search, based on the current active phase.

It should be noted that the better quality of generated plans is an extremely important aspect for the UTC application domain. In the real-world application, this would have an impact on the air quality of the area, due to a noticeable emission reduction, and to a reduced level of stress for drivers in the network.

5 Conclusion

In this paper, we proposed a domain-specific heuristic designed for improving the performance of PDDL+ planning-based urban traffic control, called Time-Based. We introduced two variants of the Time-Based heuristic: an admissible version, that can be exploited for optimal planning, and an inadmissible one, which instead focuses on maximising the informativeness. The performed experimental analysis, conducted using historical data describing the traffic in the region of a large European city, indicates that: (i) existing domain-independent heuristics are not able to cope effectively with mixed discrete-continuous planning-based UTC; (ii) the Time-Based heuristic –particularly the inadmissible variant– outperforms the state-of-the-art queue-based heuristic in terms of quality of the generated plans; and (iii) the Time-Based heuristic is robust with regards to the initial conditions of the network.

For the future, we propose to extensively test the proposed heuristic on significantly different urban networks, and using different domain-independent PDDL+ planning engines. We are also interested in extending the heuristic, and the PDDL+ model, for handling more traffic control actions, such as variable-message signs for route guidance or variable speed limits.

References

1. Chong-White, C., Millar, G., Shaw, S.: SCATS and the environment study: definitive results. In: Proceedings of the 19th World Congress on Intelligent Transportation Systems (ITS) (2012)
2. Della Penna, G., Magazzeni, D., Mercorio, F., Intrigila, B.: UPMurphi: a tool for universal planning on PDDL+ problems. In: Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS) (2009)
3. Fox, M., Long, D.: Modelling mixed discrete-continuous domains for planning. J. Artif. Intell. Res. **27**, 235–297 (2006)
4. Gulić, M., Olivares, R., Borrajo, D.: Using automated planning for traffic signals control. PROMET-Traffic Transp. **28**(4), 383–391 (2016)
5. Lin, S.: Efficient Model Predictive Control for Large-Scale Urban Traffic Networks. TU Delft, Delft University of Technology, Delft (2011)
6. McCluskey, T.L., Vallati, M.: Embedding automated planning within urban traffic management operations. In: 27th International Conference on Automated Planning and Scheduling (ICAPS) (2017)
7. Piotrowski, W.M., Fox, M., Long, D., Magazzeni, D., Mercorio, F.: Heuristic planning for PDDL+ domains. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI, pp. 3213–3219 (2016)
8. Taale, H., Fransen, W., Dibbits, J.: The second assessment of the SCOOT system in Nijmegen. In: IEEE Road Transport Information and Control, no. 21–23, April 1998
9. Vallati, M., Magazzeni, D., De Schutter, B., Chrpa, L., McCluskey, T.L.: Efficient macroscopic urban traffic models for reducing congestion: a PDDL+ planning approach. In: Thirtieth AAAI Conference on Artificial Intelligence (AAAI), pp. 3188–3194 (2016)