A Domain-specific Heuristic for PDDL+-based Traffic Signal Optimisation

Francesco Doria[1], **Francesco Percassi**[2], **Marco Maratea**[1], **Mauro Vallati**[2]

[1]DeMaCS, University of Calabria, Rende, Italy
[2]School of Computing and Engineering, University of Huddersfield, Huddersfield, United Kingdom
doria.francesco27.fd@gmail.com, f.percassi@hud.ac.uk, marco.maratea@unical.it, m.vallati@hud.ac.uk

Abstract

Optimising traffic signals is crucial for mitigating urban congestion, and automated planning, particularly with PDDL+, has shown promise for real-world deployment due to its flexibility and centralised perspective. While existing PDDL+ models guarantee deployability on current infrastructure, they face significant limitations: reliance on domain-independent heuristics restricts their applicability and scalability, leading to slow solution generation and unclear plan quality.

To overcome these challenges and unlock the widespread adoption of planning-based traffic control, we introduce h^{CAFE}, a domain-specific heuristic for PDDL+-based traffic signal optimisation. Unlike prior approaches, h^{CAFE} is designed to work effectively across multiple problem encodings, addressing a key limitation of traditional domain-specific heuristics. We demonstrate its capabilities on real-world data from a region of the UK, showing significant improvements in solution generation time and search space exploration. Our evaluation also compares the strategies generated by h^{CAFE} against historical data from existing traffic control systems and a non-deployable benchmark, confirming the high quality of the resulting plans.

Code — https://github.com/Fd3231/ENHSP-UTC

Introduction

Over half of the world's population now lives in urban areas, and urbanisation trends indicate that this percentage will rise to 70% by 2050. This trend is putting increasing strain on urban transportation systems, where growing congestion is already affecting the economy, the environment, and citizens' quality of life. In this context, AI techniques are increasingly considered for supporting urban traffic control (Whig et al. 2024), particularly for traffic signal optimisation, which is the main mechanism available to traffic authorities for influencing and controlling traffic movements in an urban area (Sharon 2021). In a nutshell, traffic signal optimisation involves determining the optimal green time to allocate for each traffic signal stage within a controlled region to minimise congestion or achieve specific goals.

There is a growing interest in the use of automated planning techniques for traffic signal optimisation; their use yields the benefits of great flexibility in terms of goals that can be described and achieved, and a centralised overview of the target region (Smith 2020; Vallati and Chrpa 2023). The use of PDDL+ planning (Fox and Long 2006) has enabled real-world deployment and implementation of traffic signal optimisation techniques (McCluskey and Vallati 2017; Kouaiti et al. 2024), with significant benefits already harvested by local communities in the United Kingdom (UK). In particular, the work by Kouaiti et al. (2024) introduced three PDDL+ models that guarantee deployability on existing urban traffic control infrastructure and provide different levels of flexibility for the optimisation task.

While the PDDL+ models proposed by Kouaiti et al. (2024) guarantee deployable signal strategies and compatibility with domain-independent heuristics, significant limitations remain. First, domain-independent heuristics are effective only for the most constrained models, since the added complexity of more flexible models cannot be managed efficiently. Second, the use of domain-independent heuristics strongly limits both the scalability and real-time applicability of the overall approach. Solutions may take several minutes to generate, making them unsuitable for real-time use, where solutions are needed within seconds, and the approach cannot scale to large urban regions. Third, in the context of PDDL+, there is limited confidence in the quality of the generated plans: domain-independent heuristics, by definition, lack specific knowledge of the target domain, and they are likely to return very suboptimal solutions (Percassi, Scala, and Vallati 2025).

To address these issues, we present h^{CAFE}, a domain-specific heuristic for PDDL+, where CAFE stands for Configuration-Aware Flow Estimation. h^{CAFE} is truly domain-specific, as it has been designed to work across multiple encodings of the same problem, hence overcoming one of the major issues of this class of heuristics, which tend to be encoding-specific rather than domain-specific. Here, we present the heuristic and demonstrate its performance improvements using real-world data. Further, we compare the generated strategies with the historical ones implemented by the existing traffic control system in a target region of the UK and with a non-deployable approach to confirm the quality of the resulting plans. h^{CAFE} has been shown to deliver significant runtime improvements and to support the generation of high-quality signal plans.

Background

In this section, we review the background on PDDL+ planning in its discrete-time formulation, the one adopted in real-world traffic signal optimisation, and briefly outline how such planning tasks are solved via heuristic search.

PDDL+ Planning

A *discrete-time* PDDL+ *planning task*, denoted by Π, is defined as a tuple $\langle F, X, I, G, A, E, P \rangle$, accompanied by a *time discretisation step* $\delta \in \mathbb{Q}^+$, which determines the temporal granularity of the model. F is a finite set of Boolean variables, each assigned a value in $\{\bot, \top\}$, while X is a finite set of numeric variables with values in $\mathbb{Q}$. A *state* s is a complete assignment to all variables in $F \cup X$, and for each variable v, the notation $s[v]$ indicates its value in state s. The initial state I defines the system's configuration at time zero, and the goal G specifies the conditions required in the final state. Boolean conditions are of the form $\langle v = b \rangle$ for $v \in F$ and $b \in \{\bot, \top\}$, while numeric conditions are expressed as $\langle \varphi \bowtie 0 \rangle$, where φ is an arithmetic expression over X and $\bowtie \in \{>, \geq, =, \leq, <\}$. The sets A, E, and P contain the actions, events, and processes, respectively. Each element $z \in A \cup E \cup P$ is described by a pair $\langle \mathit{pre}(z), \mathit{eff}(z) \rangle$, where $\mathit{pre}(z)$ denotes the preconditions and $\mathit{eff}(z)$ the effects. Preconditions are given as propositional formulae involving Boolean and numeric conditions. For actions and events, effects are expressed as assignments of the form $\langle v := b \rangle$ or $\langle v := \varphi \rangle$. Processes behave differently: although they also have preconditions, their effects are defined as discrete-time numeric updates $\langle v, \varphi \rangle$, meaning that at each time step of size δ the value of v evolves according to $v(t + \delta) = v(t) + \varphi(t) \cdot \delta$. In this work, these effects are assumed to be constant, that is, φ is a fixed rational value, yielding linear increments or decrements of v over time.

A PDDL+ plan π_t is given by the pair $\langle \pi, t_e \rangle$, where π is a sequence of timestamped actions $\langle \langle a_1, t_1 \rangle, \ldots, \langle a_n, t_n \rangle \rangle$, and $t_e \in \mathbb{Q}_0^+$ denotes the makespan or overall duration.

The validity of π_t is assessed by simulating its execution over discrete timepoints $\{0, \delta, 2 \cdot \delta, \ldots, t_e\}$. At each time step, the state is updated by accounting for the effects of actions, triggered events, and active processes. Each action applied at time t_i causes an immediate transition based on its effects. Following the application of each action and each time increment δ, events are evaluated and applied if their preconditions are satisfied. Processes continuously affect the state at every time step by applying their incremental effects. The simulation proceeds until time t_e, and the plan π_t is considered valid for Π if every action is executable and the final state satisfies the goal condition G. Further semantic details can be found in the works of Fox, Long, and Magazzeni (2012) and Percassi, Scala, and Vallati (2025).

PDDL+ Planning as Heuristic Search

Automated planning has been the subject of extensive research over the past decades, focusing largely on propositional and numeric formulations. Among the methodologies explored, heuristic search (Bonet and Geffner 2001) emerged as the dominant approach (Taitler et al. 2024), although planning as satisfiability has recently experienced renewed attention and shown promising advances (Leofante 2023; Cardellini, Giunchiglia, and Maratea 2024). Within heuristic search, planning problems are represented as state-space search tasks, where domain actions induce transitions between states. A heuristic is a function that estimates the cost or distance from a state to the goal, whose use, combined with a search strategy, guides the expansion of states until the goal is reached.

PDDL+ sets itself apart from other planning formalisms primarily through its explicit modelling of exogenous events and autonomous processes. Much of the research on PDDL+ planning has focused on time-discretised variants (Della Penna et al. 2009; Piotrowski et al. 2016; Scala et al. 2016; Piotrowski and Perez 2024), where time progresses in fixed increments. Continuous approaches have also been explored (Batusov and Soutchanski 2019; Cashmore, Magazzeni, and Zehtabi 2020; Mathew and Soutchanski 2023); however, in the context of traffic signal optimisation, the discretised approach has proven viable and effective, particularly considering the linear nature of the underlying models. The advantage of adopting a discrete-time setting is that many well-established techniques from classical and numeric planning can be effectively adapted to work within the PDDL+ framework, with some caveats regarding successor generation and heuristic evaluation. In discrete-time PDDL+, the passage of time is made explicit through a fictitious domain action, referred to as *wait*, which advances the system by δ. Similar to domain actions, *wait* actions cause numeric transitions, which are governed by the set of active processes influencing numeric variables. This mechanism is made explicit in the compilation of discrete-time PDDL+ into numeric planning presented by Percassi, Scala, and Vallati (2023). Furthermore, the application of any action, including *wait*, can result in a cascade of events. These are handled through a fixed-point mechanism, ensuring that all events whose preconditions are satisfied are applied before the next decision point.

The Traffic Signal Optimisation Models

In this section, we describe the family of PDDL+ models introduced by Kouaiti et al. (2024), which provides the basis for the development of a specialised planning heuristic. We take the opportunity to formalise the underlying structure shared by the models, which captures the traffic signal

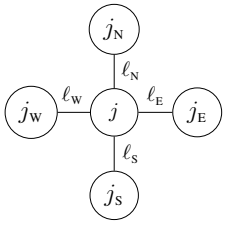

Figure 1: The network structure $\mathcal{N}$ of a configuration-controlled UTC network $\mathcal{U} = (\mathcal{N}, \mathcal{S}, \mathcal{C})$.

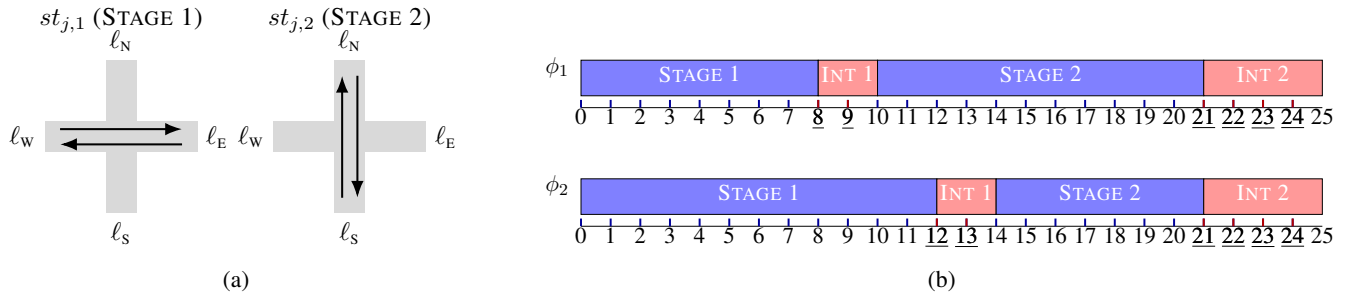

Figure 2: Considering the configuration-controlled UTC network $\mathcal{U} = (\mathcal{N}, \mathcal{S}, \mathcal{C})$ shown in Figure 1, the components: (a) stage structure $\mathcal{S}$; and (b) set of signal configurations $\mathcal{C}$.

optimisation problem with input signal configurations.

Unlike more expressive planning models (see, e.g., (Percassi et al. 2023)), where the agent can dynamically control the duration of each signal stage, in this setting the agent is restricted to selecting signal configurations from a predefined finite set. This mechanism ensures the deployability of the approach and the generated signal plans on heritage urban control infrastructure, where traffic signal configurations must be validated before they are put into operation. Informally, a configuration specifies the green times for the stages of a junction, taking into account existing constraints on the duration and sequence of stages as well as overall durations.

Within this framework, Kouaiti et al. (2024) introduce three deployable models, each differing in how configuration changes are handled over time. While these behavioural differences impact both the expressiveness of the models and the complexity of solving them, all are grounded in a common underlying structure that defines the network and its control mechanism. Informally, this structure captures the network's topology, the organisation of signal cycles through stage sequences, the vehicles' flows, and the set of available configurations used for control. We now proceed with a formal definition of the structural elements shared by all three deployable models.

A *Configuration-Controlled UTC Network* is defined as a tuple $\mathcal{U} = \langle \mathcal{N}, \mathcal{S}, \mathcal{C} \rangle$, where each element is detailed below.

Road Network The first element of the tuple, $\mathcal{N} = (J, L)$, defines the *road network* as a directed graph: J is the set of *junctions* (nodes), and $L \subseteq J \times J$ is the set of edges, referred to as links, which represent road segments connecting pairs of junctions. Each link is associated with a fixed *capacity*, specifying the maximum number of vehicles that can be held within the segment, and an *occupancy* value, which tracks the current number of vehicles occupying the link. A distinguished subset $\hat{J} \subseteq J$ identifies the *signalised junctions*, i.e., those equipped with traffic signal control.

Junction Stages The second element of the tuple, $\mathcal{S} = (ST, ig, mov, tr)$, referred to as the *stage structure*, describes how stages are organised at signalised junctions and how they affect vehicle movements within the network.

The component ST represents the set of signal stages across all signalised junctions. This set is partitioned by junction, with each $j \in \hat{J}$ having its own cycle over a dedicated stage set. Formally, we define $ST = \bigcup_{j \in \hat{J}} ST_j$ as the set of *stages*, where each $ST_j = \{st_{j,1}, st_{j,2}, \ldots, st_{j,n_j}\}$ is an ordered set of stages associated with junction $j \in \hat{J}$. Each stage defines a distinct traffic phase that enables a specific group of vehicle movements.

Each stage $st \in ST$ is associated with an *intergreen time*, representing the time required to clear the junction safely before the next stage begins. The function $ig : ST \to \mathbb{Q}^+$, assigns to each stage its corresponding intergreen duration.

Additionally, each stage enables a set of *vehicle movements*, representing permissible traffic flows through the junction when the stage is active. The movement function is $mov : ST \to 2^{L \times L}$, where $mov(st)$ returns the set of directed movements of the form $(\ell_{in}, \ell_{out}) \in L \times L$ enabled during stage st, such that vehicles enter the junction via $\ell_{in} = (j_{in}, j)$ and exit via $\ell_{out} = (j, j_{out})$, for some $j \in \hat{J}$.

Each movement is further associated with a *turn rate*, which reflects the expected number of vehicles performing that movement during stage activation. This is formalised as the function $tr : \bigcup_{st \in ST} mov(st) \to \mathbb{Q}^+$, where $tr(\ell_{in}, \ell_{out})$ denotes the expected vehicle flow through the movement during the corresponding stage. It is worth noting that turn rates are expressed as rational values; i.e., the model does not explicitly represent individual vehicles. For example, a turn rate of 0.15 means that, for each time unit δ, approximately $0.15 \cdot \delta$ vehicles are expected to leave the incoming link and enter the corresponding outgoing link. Consequently, both incoming and outgoing link occupancies may evolve over time as rational values. Link capacities, by contrast, act as upper bounds that limit the maximum occupancy a link may hold at any given time.

Signal Configurations The third element of the tuple, $\mathcal{C} = (C, T_{\text{cycle}})$, defines the available *signal configurations* used to control the network. The component C is the set of admissible configurations, partitioned by junction. In particular, let $C = \bigcup_{j \in \hat{J}} C_j$, where each C_j is the set of valid signal configurations for junction $j \in \hat{J}$. A configuration $\phi \in C_j$ is a function $\phi : ST_j \to \mathbb{Q}^+$, which assigns to each stage $st_{j,k} \in ST_j$ a green time $gt_{j,k} = \phi(st_{j,k})$, determining how long that stage remains active during a cycle.

Each configuration must comply with a fixed cycle dura-

tion $T_{\text{cycle}} \in \mathbb{Q}^+$, which is common to all controlled junctions. Specifically, a configuration $\phi \in C_j$ is valid if the sum of the green times and the intergreen times for all stages at junction j matches the cycle time:

$$\sum_{k=1}^{n_j} \phi(st_{j,k}) + \sum_{k=1}^{n_j} ig(st_{j,k}) = T_{\text{cycle}}.$$

In the following example, to provide an intuitive explanation, we show how, given a simple network, the elements of $\mathcal{U}$ are instantiated, focusing on a single junction.

Example 1. *Consider a simple network with five junctions and four links forming an intersection, with a single controllable junction $\hat{J} = \{j\}$ at its centre. The four junctions located at the edges of the network define its boundaries and serve as entry points for vehicles. Figure 1 provides the network structure.*

The controllable junction j has two stages, i.e., $ST_j = \{st_{j,1}, st_{j,2}\}$. The intergreen times are fixed, i.e., $ig(st_{j,1}) = 2$ and $ig(st_{j,2}) = 4$. The first stage $st_{j,1}$ allows movements from west to east and vice versa, i.e., $mov(st_{j,1}) = \{(\ell_{\text{W}}, \ell_{\text{E}}), (\ell_{\text{E}}, \ell_{\text{W}})\}$, while the second stage $st_{j,2}$ allows movements from north to south and vice versa, i.e., $mov(st_{j,2}) = \{(\ell_{\text{N}}, \ell_{\text{S}}), (\ell_{\text{S}}, \ell_{\text{N}})\}$. Each of these movements is associated with a turn rate prescribed by the function tr. In contrast, all boundary junctions are modelled with a single, fictitious stage that is always active. This stage injects vehicles into the network via a virtual incoming link that acts as a reservoir. The flow from this external source is regulated in line with the network demand. Figure 2a provides a representation of the stage structure.

The network cycles have a duration of $T_{cycle} = 25$ time units. Junction j is equipped with two alternative configurations $C_j = \{\phi_1, \phi_2\}$, where $\phi_1(st_{j,1}) = 8$, $\phi_1(st_{j,2}) = 11$, and $\phi_2(st_{j,1}) = 12$, $\phi_2(st_{j,2}) = 7$. Figure 2b provides a representation of the two configurations.

Optimising Configuration-Controlled UTC Networks All three deployable PDDL+ models operate over a hybrid structure that couples discrete control of traffic signals with an environmental component that models the network evolution in response to control decisions.

Each link $\ell \in \mathcal{L}$ in the network is associated with two key numeric variables: the occupancy variable, which represents the current number of vehicles (bounded by the link's capacity), and a counter, which tracks the cumulative number of vehicles that have traversed the link over time. Such variables are denoted as occ_ℓ and count_ℓ, respectively. While occupancy values may increase or decrease depending on vehicle movement, counters only grow monotonically. Configurations associated with controllable junctions, each prescribing a distribution of green times across the local stages, are encoded as numeric invariants. The currently active configurations are tracked using Boolean variables of the form active^j_ϕ, where $\phi \in C_j$ and $j \in \hat{J}$. Moreover, for each junction, we need to keep track of the current configuration state, specifically, whether we are in a stage or an intergreen phase. This is done using Boolean variables to represent the current state (e.g., which stage is active, whether the system is in an intergreen period), and numeric clock variables to measure the time spent in green and intergreen phases.

Control is applied through the shared action $changeConfiguration(j, \phi, \phi')$, which switches junction j from configuration ϕ to ϕ', provided that ϕ is currently used. This action is available only at the end of a signal cycle. The execution of this action indirectly alters the network dynamics through processes and events, primarily by activating different traffic stages and their associated movements. For each active stage st, the permitted movements $mov(st)$ are realised through a set of dedicated processes, one for each movement $(\ell_{in}, \ell_{out}) \in mov(st)$. Each of these processes is active only while the corresponding stage st is active, and models the continuous flow of vehicles from the incoming link ℓ_{in} to the outgoing link ℓ_{out}, at the rate $tr(\ell_{in}, \ell_{out})$.

A PDDL+ planning task based on $\mathcal{U}$ is defined by an initial state, which represents a snapshot of the current network conditions, and a goal condition. The goal is expressed as a conjunction of numeric conditions requiring that, for a given subset of links, the corresponding counter variables exceed a specified threshold, i.e., $\langle \mathsf{count}_\ell \geq K_\ell \rangle$.

Solving the PDDL+ planning task structured according to a network instance consists of computing a sequence of configuration assignments to be applied at the end of each control cycle for each junction. The objective is to find a plan that satisfies the goal condition upon its execution.

The models described by Kouaiti et al. (2024) differ essentially in the additional constraints they impose, which affect the set of admissible solutions and, consequently, the shape of the resulting plans, while sharing the same underlying infrastructure described above. Specifically, *Cycle by Cycle* (CBC) allows a configuration change at every cycle, *Fixed Repetition* (FIRE) restricts changes to occur only every fixed number of cycles, and *Variable Repetition* (VARE) introduces an additional control action to set the minimum number of cycles for which a configuration must be retained.

The CAFE Heuristic

This section introduces h^{CAFE}, the Configuration-Aware Flow Estimation heuristic, that is domain-specific and applicable to all presented deployable models.

The heuristic estimates the number of vehicles that can traverse a link, considering the current junction configurations at both ends. By evaluating inbound and outbound traffic flows, the heuristic can guide decisions about whether configuration changes are necessary to achieve the goal.

The heuristic is non-admissible by design: in some scenarios, it may underestimate the achievable flow and therefore overestimate the remaining cost. This heuristic is specifically designed for deployment within a real-time control architecture, where planning-based decisions must be made continuously. Leveraging the methodology proposed by Bhatnagar et al. (2022b), the system constructs planning tasks on the fly at regular intervals, capturing the current traffic conditions by querying the underlying legacy traffic infrastructure. The planning component must then produce a deployable plan as quickly as possible. In this context,

Algorithm 1: Heuristic h^{CAFE} computation for a given (restricted) state σ, goal condition G and network $\mathcal{U}$.

1: **function** HEURISTIC($\sigma, G, \mathcal{U}$)
2: $(\cdot, (ST, ig, mov, tr), (C, \cdot)) \leftarrow \mathcal{U}$
3: $h^{\text{CAFE}} \leftarrow 0$
4: **for** $\langle \mathsf{count}_\ell \geq K_\ell \rangle \in G$ **do**
5: $j_{in}, j_{out} \leftarrow \ell$
6: $\phi_{j_{in}}, \phi_{j_{out}} \leftarrow$ GETACTIVECONF(σ, C, ℓ)
7: $E_{j_{in}} \leftarrow 0$
8: **for** $st \in ST_{j_{in}}$ **do**
9: $tr_\Sigma \leftarrow \sum_{\substack{m \in mov(st): \\ m=(\ell', \ell)}} tr(m)$
10: $gt_{st} \leftarrow \phi_{j_{in}}(st)$
11: $E_{j_{in}} \leftarrow E_{j_{in}} + (tr_\Sigma \cdot gt_{st})$
12: **end for**
13: $E_{j_{out}} \leftarrow 0$
14: **for** $st \in ST_{j_{out}}$ **do**
15: $tr_\Sigma \leftarrow \sum_{\substack{m \in mov(st): \\ m=(\ell, \ell')}} tr(m)$
16: $gt_{st} \leftarrow \phi_{j_{out}}(st)$
17: $E_{j_{out}} \leftarrow E_{j_{out}} + (tr_\Sigma \cdot gt_{st})$
18: **end for**
19: $E \leftarrow \min(E_{j_{in}}, E_{j_{out}})$
20: $h^{\text{CAFE}} \leftarrow h^{\text{CAFE}} + \max(K_\ell - \sigma[\mathsf{count}_\ell] - E, 0)$
21: **end for**
22: **return** h^{CAFE}
23: **end function**

the most effective strategy is to combine a domain-specific, lightweight yet informative heuristic with Greedy Best-First Search (GBFS), which prioritises state expansion based on the heuristic, enabling faster search.

Given a configuration-controlled UTC network $\mathcal{U}$ and a deployable model $\mathcal{M}$ chosen among CBC, FIRE and VARE, we instantiate the corresponding PDDL+ planning task $\Pi_{\mathcal{U}}^{\mathcal{M}}$, where its structure reflects both the physical network and the control prescribed by $\mathcal{M}$. While each model differs in how configuration changes are constrained, the core state representation of $\Pi_{\mathcal{U}}^{\mathcal{M}}$ is largely shared across variants and reflects the same underlying domain structure. This makes it possible to define a heuristic specific to the traffic control setting yet independent of the particular control model implemented on the network, provided that the model exposes the same set of variables described below. We now focus on identifying the subset of variables $V \subset F \cup X$ in this shared representation that forms the basis for the heuristic. Such a set is defined as $V = F_{\mathsf{active}} \cup X_{\mathsf{count}}$, where $F_{\mathsf{active}} = \{\mathsf{active}_\phi^j \mid j \in \hat{J}, \phi \in C_j\}$ and $X_{\mathsf{count}} = \{\mathsf{count}_\ell \mid \langle \mathsf{count}_\ell \geq K_\ell \rangle \in G\}$.

We wish to emphasise a crucial design aspect of our heuristic. Unlike so-called domain-dependent heuristics, typically tied to specific encodings, ours is better described as *variable-dependent*. This makes it portable across various models, provided that the representation includes the variable set V and a counter-based goal. Accordingly, we distinguish between a state s, defined over the full set of variables $F \cup X$, and a restricted state σ, which is a complete assignment over V. While s includes all auxiliary and encoding-specific variables of $\Pi_{\mathcal{U}}^{\mathcal{M}}$, σ captures only the traffic-relevant variables leveraged by the heuristic.

The heuristic function takes as input a restricted state σ, a goal G, and a network $\mathcal{U}$; its pseudocode is provided in Algorithm 1.

It begins by initialising the heuristic value h^{CAFE} to zero (Line 3). This value is updated as the function processes each goal condition, which typically takes the form $\langle \mathsf{count}_\ell \geq K_\ell \rangle$, indicating that at least K_ℓ vehicles should have passed through link ℓ. For each such goal, the function retrieves the start and end junctions of the link ℓ (Line 5), along with their active configurations using GETACTIVECONF(σ, C, ℓ) (Line 6). Each configuration defines a set of signal stages and their associated green times.

To estimate the contribution to the link counter from the current configurations, the heuristic computes the expected inflow and outflow. For the start junction j_{in}, the function iterates over each stage in the active configuration, identifies movements whose target link is ℓ, sums their turn rates, multiplies this sum by the stage's green time, and accumulates the result to compute the expected inflow $E_{j_{in}}$ (Line 8 to 12). A symmetric process is applied at the end junction j_{out}, where movements originating from link ℓ are used to compute the expected outflow $E_{j_{out}}$ (Line 14 to 18).

The effective expected counter increment (Line 19), computed as the minimum of the two flows, captures the fact that the number of vehicles advancing through a link is constrained by both the inflow and the outflow allowed by the current configurations at its ends. The underlying settings implicitly favoured by the heuristic are those in which the flows at both ends of each link remain balanced, meaning that the junction configurations allow a comparable number of vehicles to be served. This behaviour aims to prevent congestion, which occurs when flows are imbalanced, and to maintain smooth traffic movement through the crucial links of the target region.

The heuristic value is then updated (Line 20) to reflect the remaining contribution required to reach the goal, given the current counter value and the estimated progress under the current configurations.

This process is repeated for each goal condition, and the final heuristic value h^{CAFE} accumulates the expected remaining difference between the current counter values and their targets. The resulting value provides an estimate of the additional traffic flow required, beyond what is expected under the current configuration, to satisfy the goal constraints.

After describing how the heuristic is implemented, we discuss its non-admissibility. For the sake of clarity, let us recall that a heuristic h is admissible if it never overestimates the distance of a state s to a goal state. In other words, for each state s, $h(s) \leq h^\star(s)$, where $h^\star(s)$ denotes the remaining vehicles to reach the goal, in our case.

We focus on a minimal setting involving a single link with two signalised junctions at its endpoints. By definition, $h^{\text{CAFE}}(s) = K_\ell - \sigma[\mathsf{count}_\ell] - E$, while $h^\star(s) = K_\ell - \sigma[\mathsf{count}_\ell] - E^\star$. Here, K_ℓ is the goal counter, and $\sigma[\mathsf{count}_\ell]$ is the current counter. The two expressions dif-

fer as E is the expected counter increment, while $E^\star$ is the actual one. To show non-admissibility, it suffices to identify an assignment of σ such that $E < E^\star$. Recall that $E = \min(E_{j_{in}}, E_{j_{out}})$, and suppose that for the configurations in use at the end points, we have $E_{j_{out}} < E_{j_{in}}$. The counter of ℓ, count_ℓ, is governed by the movements associated with j_{in}. Therefore, in any state where the upstream links feeding ℓ can supply $E_{j_{in}}$ vehicles and ℓ can absorb that inflow, we obtain $E^\star = E_{j_{in}}$. However, the heuristic uses $E = E_{j_{out}}$, thus overestimating the remaining cost.

Empirical Analysis

This experimental analysis aims to assess the performance of the proposed heuristic and compare the quality of the generated plans with that of the traffic control infrastructure currently operating in the target urban region.

Experimental Setting

We use as a benchmark the PDDL+ planning tasks introduced by Kouaiti et al. (2024). The evaluation focuses on a set of planning tasks centred on a key traffic corridor located within the metropolitan area of Kirklees, West Yorkshire, United Kingdom. The corridor is approximately 1.3 km long and consists of six controllable junctions and 35 road links. The benchmark comprises six scenarios derived from real traffic data collected on two representative weekdays: Wednesday, the 26th of January 2022, and Sunday, the 30th of January 2022. For each day, we consider three distinct time intervals, 08:30 (morning peak hour), 12:30 (noon), and 16:30 (evening peak hour), to reflect varying traffic conditions throughout the day. An additional scenario captures an exceptional traffic condition recorded on Tuesday, the 20th of June 2023, during a concert at John Smith's Stadium, which attracted approximately 30,000 attendees. Following the approach used by Kouaiti et al. (2024), five different benchmark problems are generated for each scenario, to have at least 350 vehicles navigate through an increasing number of links as quickly as possible. This simulates a range of conditions that an urban traffic control system may need to handle, from corridor-wide flushing to pulling vehicles from a specific input region. All presented results are averaged across the five problems. All scenarios are instantiated across three deployable models: CBC, FIRE, and VARE. In all experiments, we set the discretisation step to $\delta = 1$ second. These models operate over a shared set of traffic signal configurations, all assuming a fixed cycle time of $T_{\text{cycle}} = 90$ seconds. Six configurations are available for selection at each junction. The configurations were extracted from historical signal plans used by the SCOOT system in operation in the study area (Taale, Fransen, and Dibbits 1998), and correspond to the so-called G-HIST configurations described in Kouaiti et al. (2024). All heuristic-based planning approaches used in our experiments are implemented on top of the ENHSP planner (version 20) (Scala et al. 2020) and combined with Greedy Best-First Search (GBFS), which has been identified as the most performant approach by Kouaiti et al. (2024).

All experiments were run once, on a machine equipped with an AMD Ryzen 5 7535U, 32GB of RAM, OS Ubuntu 24.04, and Java 17.0.1 (required by ENHSP). Maximum runtime for each run was set to 300 CPU-time seconds.

Scen.	Heur.	Runtime			Expanded Nodes		
		CBC	FIRE	VARE	CBC	FIRE	VARE
A-morn	h^{CAFE}	**3,308**	**3,299**	**4,060**	**2,000**	**2,000**	**2,002**
	h^{add}	62,766	15,237	–	190,204	22,665	–
	h^{max}	27,616	10,946	43,076	65,301	18,916	88,877
A-noon	h^{CAFE}	**3,295**	**3,046**	**4,233**	**1,936**	**1,936**	**1,936**
	h^{add}	36,597	18,677	–	78,102	31,691	–
	h^{max}	5,462	4,503	12,377	3,958	3,164	18,848
A-eve	h^{CAFE}	**3,293**	**3,104**	**4,290**	**1,911**	**1,959**	**1,959**
	h^{add}	–	160,095	–	–	376,280	–
	h^{max}	17,922	8,597	–	31,232	9,200	–
B-morn	h^{CAFE}	**4,086**	**4,284**	**6,938**	**3,880**	**3,880**	**3,880**
	h^{add}	–	232,265	–	–	481,933	–
	h^{max}	252,237	92,952	–	692,955	244,317	–
B-noon	h^{CAFE}	**3,060**	**3,207**	**3,826**	**1,882**	**1,882**	**1,883**
	h^{add}	22,786	9,899	101,125	59,094	9,973	169,990
	h^{max}	9,042	4,714	9,556	9,120	3,201	12,194
B-eve	h^{CAFE}	**3,039**	**3,188**	**4,337**	**1,998**	**1,998**	**1,998**
	h^{add}	180,296	14,276	–	425,581	22,003	–
	h^{max}	38,135	14,207	35,163	108,090	23,891	71,852
Concert	h^{CAFE}	**3,037**	**3,112**	**4,118**	**1,904**	**1,928**	**1,928**
	h^{add}	–	11,628	–	–	15,361	–
	h^{max}	22,461	8,537	24,771	46,299	8,006	50,726

Table 1: Performance achieved by GBFS coupled with the considered heuristics in terms of runtime (milliseconds) and number of expanded nodes, across the three planning models. "–" indicates that the approach did not find a solution within the maximum runtime. Bold indicates the best results for a model in a scenario.

Planning Performance

First, we turn our attention to the performance improvement that the introduced heuristic can deliver when used in the considered planning architecture. To contextualise performance, we consider two domain-independent numeric heuristics available in the ENHSP framework and used in previous work to generate traffic signal control strategies via PDDL+, namely h^{add} and h^{max} (Scala et al. 2020). Numeric h^{max} and h^{add} extend their classical heuristics to numeric conditions, with h^{max} taking the maximum subgoal cost and h^{add} summing all subgoal costs.

Table 1 provides an overview of the performance achieved by the ENHSP planning engine using the considered heuristics and the three different planning models, in terms of runtime (milliseconds) and expanded nodes. It is evident that using h^{CAFE} significantly improves performance in both runtime and expanded nodes, indicating that the heuristic is effective at guiding the search. The most remarkable impact comes, unsurprisingly, from the VARE model, which is the most flexible one, where standard domain-independent heuristics struggle to navigate a large search space characterised by extensive plateaux. It is worth noting that the

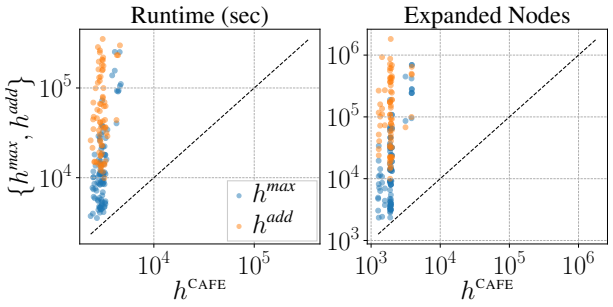

Figure 3: Scatter plots comparing the performance of h^{CAFE} and h^{add} (h^{max}) in orange (blue) in terms of expanded nodes (left) and runtime (right). Dots above the diagonal indicate that h^{CAFE} outperforms the considered competitor with regards to the target metric.

domain-independent heuristics do not always find solutions when CBC or VARE are used, whereas h^{CAFE} enables the planner to solve all instances across all models.

The extent of the improvement is also visible in Figure 3, where the scatter plots compare the performance of h^{CAFE} with that of h^{add} and h^{max} on the considered benchmarks. In terms of expanded nodes, h^{CAFE} always expanded the fewest, which is reflected in the significantly lower runtime required to find a solution.

State of the Art Comparison

We can now turn our attention to the quality of the solutions generated by h^{CAFE}. We consider the FIRE model, which offers the best trade-off between deployability and flexibility. To assess solution quality, we compare the generated plans with those historically implemented by the SCOOT system operating in the target region, denoted as $\mathcal{H}$ (Historical), which we use as our baseline.

We also consider as a reference the method introduced by Percassi et al. (2023), which serves as an upper bound on performance. This method relies on a domain-dependent heuristic, h^{TSO}, specifically designed for a more expressive model called *Extend and Reduce*, and it also adopts GBFS as the search strategy. Although capable of producing high-quality signal plans, this approach cannot be deployed on the existing UTC infrastructure, as it dynamically alters cycle layouts and violates the requirement for fixed cycle durations. Thus, it was not considered in the previous evaluation.

All the plans generated by the planning-based approaches and the historical data from the SCOOT system were evaluated through simulation, leveraging the environment introduced in (Bhatnagar et al. 2022a). This framework enables performing accurate traffic simulations within the urban area of interest. The key metrics used to assess the quality of both the signal plans and the historical data are as follows:

- $0 \leq \mu_Z \leq 1$ represents the average occupancy, normalised to the static capacity of the links in the west-to-east corridor. A value close to one indicates a high level of congestion.

Scenario	Approach	μ_Z	$count_C$	*in*	*middle*	*out*
A-morn	h^{CAFE}	0.26	940.2	**417.1**	198.9	189.4
	h^{TSO}	**0.13**	**1108.8**	**417.1**	**253.6**	**235.8**
	$\mathcal{H}$	0.28	887.4	417.1	221.0	181.7
A-noon	h^{CAFE}	0.38	1001.0	**551.5**	197.6	208.4
	h^{TSO}	**0.16**	**1268.4**	547.6	261.2	**264.8**
	$\mathcal{H}$	0.35	1138.3	551.5	**270.9**	227.3
A-eve	h^{CAFE}	0.40	1010.7	585.1	196.9	227.2
	h^{TSO}	**0.15**	**1437.0**	599.9	298.5	**292.8**
	$\mathcal{H}$	0.40	1317.9	**614.7**	**309.1**	271.9
B-morn	h^{CAFE}	**0.02**	466.4	**173.8**	106.0	**98.1**
	h^{TSO}	**0.02**	**468.3**	**173.8**	**108.6**	97.2
	$\mathcal{H}$	0.07	417.8	173.8	102.6	83.2
B-noon	h^{CAFE}	0.37	1020.0	554.8	199.4	215.0
	h^{TSO}	**0.17**	**1322.0**	557.7	**279.4**	**259.0**
	$\mathcal{H}$	0.73	612.9	**558.9**	146.1	74.9
B-eve	h^{CAFE}	0.17	912.7	**353.4**	193.3	196.7
	h^{TSO}	**0.10**	**943.9**	**353.4**	**208.2**	**207.0**
	$\mathcal{H}$	0.48	606.3	**353.4**	166.6	89.8
Concert	h^{CAFE}	**0.56**	1329.6	**612.8**	211.9	382.0
	h^{TSO}	0.64	**1492.8**	**612.8**	**269.2**	**386.6**

Table 2: Comparison between the plans generated using the h^{CAFE} in conjunction with the FIRE model, the historical strategy implemented by the SCOOT system ($\mathcal{H}$), and the upper bound h^{TSO}, which does not guarantee the deployability of the solutions. Bold indicates best results.

- $count_C$ denotes the number of vehicles that moved through the corridor during the simulation.
- $in/mid/out$ represent the total number of vehicles that entered the corridor from the western entry points, crossed its midpoint, and exited from the eastern exit points, respectively.

The results of the comparison are presented in Table 2. We note that historical results ($\mathcal{H}$) are not available for the *Concert* scenario, as the SCOOT system was overridden with a manually optimised plan strategy, generated starting from an automatically generated plan.

In most of the scenarios considered, using h^{CAFE} improves corridor performance compared with the deployed SCOOT infrastructure. This is reflected in the lower μ_Z, which indicates reduced average congestion, together with high counter values showing that the plans move similar or even larger volumes of vehicles without congesting the links. When comparing the performance of h^{CAFE} with h^{TSO}, it is possible to observe that in many cases, results are similar. However, the h^{TSO} approach is not deployable and can dynamically generate optimised traffic-signal cycles during planning, whereas h^{CAFE}, being based on the FIRE model, can only exploit a much more limited set of six predefined signal configurations per junction.

Overall, the empirical analysis confirms that (i) h^{CAFE} significantly improves the planning performance across all the deployable planning models, and (ii) it allows for delivering strategies that outperform the state of the art in operation, and that are close to those of the known upper bound.

Related Work

Several planning and scheduling approaches have been proposed for traffic signal optimisation, as they are based on knowledge models that support validation and verification, thereby supporting deployability and real-world implementation. Gulić, Olivares, and Borrajo (2016) introduced a system integrating an AI planning engine with the SUMO simulator (López et al. 2018) through an "Intelligent Autonomic System" module. Their PDDL2.1 model employs qualitative density descriptors (e.g., "low", "medium") to represent traffic on road links, abstracting from individual vehicle counts and enabling scalability to large networks. Building on this, Pozanco, Fernández, and Borrajo (2021) incorporated continuous learning and knowledge model evolution to enhance adaptability. Preliminary work by Ivankovic et al. (2022) applies planning techniques with global state constraints (Haslum et al. 2018), offering insights into the broader impact of signal changes.

The SURTRAC system adopts decentralised scheduling for urban traffic control (Xie, Smith, and Barlow 2012; Hu and Smith 2019; Smith 2020). Each intersection acts as an autonomous agent, coordinating with neighbours to predict demand and minimise delays. While this distributed approach scales well through localised decisions, it may lack the flexibility to achieve system-wide objectives compared to centralised AI planning.

There is also a significant body of work on multi-agent reinforcement learning approaches. This class of approaches (see, e.g., (Wu, Ma, and Kim 2020; Chu et al. 2020; Long and Chung 2023; Savithramma and Sumathi 2023; Othman et al. 2025)) typically aims to optimise local and network-wide performance, using deep reinforcement learning to handle large-scale and complex traffic scenarios. The well-known approach GPLight (and the subsequent GPLight+) (Liu et al. 2023; Liao, Mei, and Zhang 2025) introduces a grouped multi-agent reinforcement learning framework for large-scale traffic signal control, in which intersections are clustered based on traffic patterns and topology to improve coordination efficiency and scalability. In contrast, Pi-Light (Gu et al. 2024) focuses on interpretability and resource efficiency by representing policies as programmatic structures rather than deep neural networks, using a domain-specific language and search-based optimisation to deliver transparent, generalisable solutions suitable for deployment on edge devices. However, this class of approaches is not suitable for deployment on existing SCOOT infrastructure and might lack the guarantees needed for safe real-world deployment.

Finally, our problem of interest has also been solved using other logic-based languages such as Answer Set Programming (ASP) (Brewka, Eiter, and Truszczynski 2011; Calimeri et al. 2020). Eiter et al. (2020) introduce an approach to optimise the coupling of traffic movements at junctions, according to expected traffic demand in the area, and to simulate using a mesoscopic-level representation. The experiments considered a realistic area with two junctions and are compared to SUMO. Tarzariol, Maratea, and Vallati (2025) modelled and solved the problem via an extension of ASP with constraints, and compared to the PDDL+-based solution in the real setting presented in this work. They also proposed a method for combining ASP-based and PDDL+-based solutions.

Conclusion

To support and further extend the applicability of planning-based traffic signal optimisation approaches, this paper introduced the domain-specific heuristic h^{CAFE}. This heuristic estimates the number of vehicles that must pass through the goal links of a target region, thereby promoting balanced traffic flows across adjacent junctions. h^{CAFE} is truly domain-specific, as it can be exploited in different existing PDDL+ domain models that encode the traffic control optimisation problem via the notion of cycle configuration. The empirical evaluation based on historical data from a major UK urban corridor shows that h^{CAFE} significantly improves planning runtime by offering more effective search guidance. Furthermore, the quality of the traffic-signal strategies generated by h^{CAFE} is better than that of the existing traffic-control infrastructure and comparable to plans produced by a specialised non-deployable approach.

Future work will focus on investigating how the proposed heuristic can inform the selection of cycle configurations for each junction. Further, we are interested in exploring admissible variants of h^{CAFE} that can prioritise the quality of generated plans to support offline reasoning, for instance, in the case of planned events.

Acknowledgments

Francesco Percassi and Mauro Vallati were supported by a UKRI Future Leaders Fellowship [grant number MR/Z00005X/1]. Marco Maratea was supported by the European Union - NextGenerationEU and by Italian Ministry of Research (MUR) under PNRR project FAIR "Future AI Research", CUP H23C22000860006.

References

Batusov, V.; and Soutchanski, M. 2019. A Logical Semantics for PDDL+. In *ICAPS*, 40–48.

Bhatnagar, S.; Guo, R.; McCabe, K.; McCluskey, T. L.; Scala, E.; and Vallati, M. 2022a. Leveraging Artificial Intelligence for Simulating Traffic Signal Strategies. In *IEEE ITSC*, 607–612.

Bhatnagar, S.; Mund, S.; Scala, E.; McCabe, K.; McCluskey, T. L.; and Vallati, M. 2022b. On-the-Fly Knowledge Acquisition for Automated Planning Applications: Challenges and Lessons Learnt. In *ICAART*, 387–397.

Bonet, B.; and Geffner, H. 2001. Heuristic Search Planner 2.0. *AI Mag.*, 22(3): 77–80.

Brewka, G.; Eiter, T.; and Truszczynski, M. 2011. Answer set programming at a glance. *Communications of the ACM*, 54(12): 92–103.

Calimeri, F.; Faber, W.; Gebser, M.; Ianni, G.; Kaminski, R.; Krennwallner, T.; Leone, N.; Maratea, M.; Ricca, F.; and Schaub, T. 2020. ASP-Core-2 Input Language Format. *Theory and Practice of Logic Programming*, 20(2): 294–309.

Cardellini, M.; Giunchiglia, E.; and Maratea, M. 2024. Symbolic Numeric Planning with Patterns. In *AAAI*, 20070–20077.

Cashmore, M.; Magazzeni, D.; and Zehtabi, P. 2020. Planning for Hybrid Systems via Satisfiability Modulo Theories. *J. Artif. Intell. Res.*, 67: 235–283.

Chu, T.; Wang, J.; Codecà, L.; and Li, Z. 2020. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.*, 21(3): 1086–1095.

Della Penna, G.; Magazzeni, D.; Mercorio, F.; and Intrigila, B. 2009. UPMurphi: A tool for universal planning on PDDL+ problems. In *ICAPS*, volume 19, 106–113.

Eiter, T.; Falkner, A. A.; Schneider, P.; and Schüller, P. 2020. ASP-Based Signal Plan Adjustments for Traffic Flow Optimization. In *ECAI*, 3026–3033.

Fox, M.; and Long, D. 2006. Modelling Mixed Discrete-Continuous Domains for Planning. *J. Artif. Intell. Res.*, 27: 235–297.

Fox, M.; Long, D.; and Magazzeni, D. 2012. Plan-based Policies for Efficient Multiple Battery Load Management. *J. Artif. Intell. Res.*, 44: 335–382.

Gu, Y.; Zhang, K.; Liu, Q.; Gao, W.; Li, L.; and Zhou, J. 2024. π-Light: Programmatic Interpretable Reinforcement Learning for Resource-Limited Traffic Signal Control. In *AAAI*, 21107–21115.

Gulić, M.; Olivares, R.; and Borrajo, D. 2016. Using automated planning for traffic signals control. *PROMET-Traffic&Transportation*, 28(4): 383–391.

Haslum, P.; Ivankovic, F.; Ramírez, M.; Gordon, D.; Thiébaux, S.; Shivashankar, V.; and Nau, D. S. 2018. Extending Classical Planning with State Constraints: Heuristics and Search for Optimal Planning. *J. Artif. Intell. Res.*, 62: 373–431.

Hu, H.; and Smith, S. F. 2019. Using Bi-Directional Information Exchange to Improve Decentralized Schedule-Driven Traffic Control. In *ICAPS*, 200–208.

Ivankovic, F.; Vallati, M.; Chrpa, L.; and Roveri, M. 2022. Urban Traffic Control via Planning with Global State Constraints (Extended Abstract). In *SoCS*, 291–293.

Kouaiti, A. E.; Percassi, F.; Saetti, A.; McCluskey, T. L.; and Vallati, M. 2024. PDDL+ Models for Deployable yet Effective Traffic Signal Optimisation. In *ICAPS*, 168–177.

Leofante, F. 2023. OMTPlan: A Tool for Optimal Planning Modulo Theories. *J. Satisf. Boolean Model. Comput.*, 14(1): 17–23.

Liao, X.-C.; Mei, Y.; and Zhang, M. 2025. GPLight+: A Genetic Programming Method for Learning Symmetric Traffic Signal Control Policy. *IEEE Trans. Evol. Comput.*

Liu, Y.; Luo, G.; Yuan, Q.; Li, J.; Jin, L.; Chen, B.; and Pan, R. 2023. GPLight: Grouped Multi-agent Reinforcement Learning for Large-scale Traffic Signal Control. In *IJCAI*, 199–207.

Long, M.; and Chung, E. 2023. Transit Signal Priority for Arterial Road with Deep Reinforcement Learning. In *IEEE MT-ITS*, 1–5.

López, P. Á.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; and WieBner, E. 2018. Microscopic Traffic Simulation using SUMO. In *IEEE ITSC*, 2575–2582.

Mathew, S.; and Soutchanski, M. 2023. Heuristic Planning for Hybrid Dynamical Systems with Constraint Logic Programming. In *IPS-RCRA-SPIRIT@AI*IA*, volume 3585 of *CEUR*.

McCluskey, T. L.; and Vallati, M. 2017. Embedding Automated Planning within Urban Traffic Management Operations. In *ICAPS*, 391–399.

Othman, K.; Wang, X.; Shalaby, A.; and Abdulhai, B. 2025. Multimodal adaptive traffic signal control: A decentralized multiagent reinforcement learning approach. *Multimodal Transp.*, 4(1): 100190.

Percassi, F.; Bhatnagar, S.; Guo, R.; McCabe, K.; McCluskey, T. L.; and Vallati, M. 2023. An Efficient Heuristic for AI-based Urban Traffic Control. In *IEEE MT-ITS*, 1–6.

Percassi, F.; Scala, E.; and Vallati, M. 2023. A Practical Approach to Discretised PDDL+ Problems by Translation to Numeric Planning. *J. Artif. Intell. Res.*, 76: 115–162.

Percassi, F.; Scala, E.; and Vallati, M. 2025. On the Notion of Plan Quality for PDDL+. In *ICAPS*, 102–111.

Piotrowski, W.; and Perez, A. 2024. Real-world Planning with PDDL+ and Beyond. *arXiv preprint arXiv:2402.11901*.

Piotrowski, W. M.; Fox, M.; Long, D.; Magazzeni, D.; and Mercorio, F. 2016. Heuristic Planning for PDDL+ Domains. In *IJCAI*, 3213–3219.

Pozanco, A.; Fernández, S.; and Borrajo, D. 2021. On-line modelling and planning for urban traffic control. *Expert Syst. J. Knowl. Eng.*, 38.

Savithramma, R.; and Sumathi, R. 2023. Intelligent traffic signal controller for heterogeneous traffic using reinforcement learning. *Green Energy Intell. Transp.*, 2(6): 100124.

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2016. Interval-Based Relaxation for General Numeric Planning. In *ECAI*, 655–663.

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2020. Subgoaling Techniques for Satisficing and Optimal Numeric Planning. *J. Artif. Intell. Res.*, 68: 691–752.

Sharon, G. 2021. Alleviating Road Traffic Congestion with Artificial Intelligence. In *IJCAI*, 4965–4969.

Smith, S. F. 2020. Smart Infrastructure for Future Urban Mobility. *AI Mag.*, 41(1): 5–18.

Taale, H.; Fransen, W.; and Dibbits, J. 1998. The second assessment of the SCOOT system in Nijmegen. In *IEE Road Transport Information and Control*, 109–113.

Taitler, A.; Alford, R.; Espasa, J.; Behnke, G.; Fiser, D.; Gimelfarb, M.; Pommerening, F.; Sanner, S.; Scala, E.; Schreiber, D.; Segovia-Aguas, J.; and Seipp, J. 2024. The 2023 International Planning Competition. *AI Mag.*, 45(2): 280–296.

Tarzariol, A.; Maratea, M.; and Vallati, M. 2025. A CASP-based Solution for Traffic Signal Optimisation. *Theory and Practice of Logic Programming*, 25(4): 794–812.

Vallati, M.; and Chrpa, L. 2023. In Defence of Good Old-Fashioned Artificial Intelligence Approaches in Intelligent Transportation Systems. In *IEEE ITSC*, 4913–4918.

Whig, P.; Velu, A.; Nadikattu, R. R.; and Alkali, Y. J. 2024. Role of AI and IoT in Intelligent Transportation. In *Artificial Intelligence for Future Intelligent Transportation*, 199–220.

Wu, C.; Ma, Z.; and Kim, I. 2020. Multi-Agent Reinforcement Learning for Traffic Signal Control: Algorithms and Robustness Analysis. In *IEEE ITSC*, 1–7.

Xie, X.-F.; Smith, S.; and Barlow, G. 2012. Schedule-Driven Coordination for Real-Time Traffic Network Control. In *Proc. of ICAPS*, 323–331.