# ASP and PDDL+ Applications in Urban Traffic Distribution and Control

Mauro Vallati[1,*], Carmine Dodaro[2], Francesco Doria[2], Salvatore Fiorentino[2], Marco Maratea[2,**], Francesco Percassi[1] and Alice Tarzariol[3]

[1]*School of Computing and Engineering, University of Huddersfield, UK*
[2]*DeMaCS, University of Calabria, Via P. Bucci, 87036, Rende (CS)*
[3]*AICS, University of Klagenfurt, Austria*

**Abstract**

Answer Set Programming (ASP) and the mixed discrete-continuos variant of the Planning Domain Definition Language (PDDL+) are well-known knowledge representation and reasoning methodologies to solve configuration, scheduling and planning problems arising in real-life applications.

In this paper, we focus on the recent problems modeled and solved with ASP and PDDL+ in the context of urban traffic distribution and control, including Traffic Signal Optimization and the optimization of pre-computed routes of Connected Autonomous Vehicles in urban networks.

**Keywords**

Answer Set Programming, Mixed Discrete-Continuous Planning, Applications

## 1. Introduction

Answer Set Programming (ASP) [1] and the mixed discrete-continuous variant of the Planning Domain Definition Language (PDDL+) [2] are well-known knowledge representation and reasoning methodologies. The ASP language [3] is declarative, rule-based and general purpose, and allows for the easy specification of both decision and optimization problems, while PDDL+ is an action-based language. Together with the availability of robust ASP [4, 5, 6, 7] and PDDL+ [8] solvers, fostered by competitions in the fields [9, 10], they have been shown to be particularly useful in real-applications: ASP particularly to solve configuration and scheduling (optimization) problems [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21], while PDDL+ has been successfully applied to solving planning problems on a variety of applications [22, 23, 24, 25, 26, 27, 28, 29].

In this paper, we overview the recent problems modelled and solved with ASP and PDDL+ in the context of urban traffic distribution and control. We first present the applications in which ASP is employed, i.e., Traffic Signal Optimization, which aims of minimizing the average traffic delay for a region of interest by determining the optimal green length, and the optimization of pre-computed routes of Connected Autonomous Vehicles in urban areas. Then, we move to PDDL+, and review several models, from flexible to deployable, related to the Traffic Signal Optimization problem. The paper ends by drawing some conclusions and by discussing a solution which integrates ASP and PDDL+ for efficiently solving the Traffic Signal Optimization problem.

*1st Workshop on Cognitive Architectures for Robotics: LLMs and Logic in Action (CARLA), September 12–13, 2025, Rende, Italy*
*Corresponding author.
**Corresponding author.
M.Vallati@hud.ac.uk (M. Vallati); carmine.dodaro@unical.it (C. Dodaro); francesco.doria@student.unical.it (F. Doria); salvatore.fiorentino@unical.it (S. Fiorentino); marco.maratea@unical.it (M. Maratea); F.Percassi@hud.ac.uk (F. Percassi); Alice.Tarzariol@aau.at (A. Tarzariol)
0000-0002-8429-3570 (M. Vallati); 0000-0002-5617-5286 (C. Dodaro); 0009-0004-6531-4139 (S. Fiorentino); 0000-0002-9034-2527 (M. Maratea); 0000-0001-7332-0636 (F. Percassi); 0000-0001-6586-3649 (A. Tarzariol)

## 2. ASP in Urban Traffic Distribution

In this section, we describe ASP applications for tackling problems appearing in Urban Traffic Distribution.

### 2.1. A Constraint ASP Solution for Traffic Signal Optimization

In [30] we introduced an ASP solution to tackle the *Traffic Signal Optimization Problem*, which aims to minimize the average traffic delay for a region of interest by determining the optimal green length for each traffic signal in a given set. We focus on a major corridor in the Kirklees council area within West Yorkshire (UK) containing 6 junctions and 34 road links. We frame the problem using a mesoscopic traffic model, where the approximate number of vehicles in road links is considered (instead of individual vehicles) and the traffic signals in a junction are abstracted as *stages*, each representing a set of simultaneous traffic movements. A *cycle* is the ordered sequence of these stages, separated by *intergreen* times. Constraints apply to the minimum and maximum durations of both stages and full cycles, while the order of stages remains fixed. The SCOOT [31] system, a traffic reactive control mechanism, is in operation in the focus area for handling cycle-to-cycle changes in traffic demand. It stores in a dedicated database data obtained from local sensors and operational information, which can be exploited to simulate historical data and generated solutions. This allows for the use of external tools, e.g., ASP and PDDL+ solvers, to perform traffic signal optimization by injecting generated strategies to be deployed in the region. However, to operate on legacy infrastructure, additional constraints must be respected, i.e., for each junction, the length of the stages is defined according to a set of possible predefined cycle configurations, and junction cycles must remain of similar duration to preserve synchronization and green waves. Performance is measured with *counters*, which track the number of vehicles that transited across each link over time. Maximizing these counters serves as a proxy for reducing delays, since higher counters imply less queuing.

In the simulation, we discretise the time in seconds and consider a horizon of up 900 seconds (i.e., 15 minutes), which is the maximum duration for which the SCOOT data remains representative. The ASP solver decisions only concern which configuration is selected for each junction. Since (*i*) the duration of each cycle is the same (regardless of the configuration), and (*ii*) once a configuration is selected, it cannot be changed unless the cycle ends, the decision points can be precomputed, drastically reducing their number. Despite this observation, calculating the occupancy and counter for the 34 road links at each second produces a number of ground rules that cannot be handled, even for short horizons (around 100 seconds). To handle this issue, we applied the system *clingcon 3* [32], which extends the well-known ASP solver *clingo* with theory atoms and propagators for linear constraints. By redefining through theory atoms the link road occupancies and counters, we managed to model and effectively apply Constraint ASP (CASP) to tackle the traffic signal optimization problem with meaningful horizons. In particular, thanks to ASP expressivity, we can maximize the counters for a set of given links, or even define more articulated optimization criteria.

### 2.2. AI-Enabled Connected Autonomous Vehicles Sustainable Routing in Urban Areas

In [33], we proposed a new framework to explicitly incorporate sustainability aspects in the context of AI-Enabled Connected and Autonomous Vehicles (CAVs). We assumed a Vehicle-to-Infrastructure (V2I) communication system that allows traffic controllers to interact with individual vehicles, hence actively managing traffic flow by directing vehicles along optimised routes to their destinations. The framework is an extension of [34] and is divided into four modules: Preprocessing, Search, Optimisation, and Monitoring & Execution. The first module was defined to abstract the network, reducing its complexity and preparing it for the search step. One of our contributions was to extend this module. We introduced an information extraction phase where relevant information regarding sustainability is extracted from the network. In particular, for each pair of streets $s_1$, $s_2$ (a street is an edge connecting two junctions) its pollution risk index for traversing $s_1$ and going towards $s_2$ is estimated, considering important factors such as the emission class of the vehicle and the street congestions while traversing them. More formally, given two streets $s_1$,$s_2$, a specific emission class $e$ and a congestion level $c$ the index of pollution is computed using the following formula:

$$risk_s(s_1, s_2, e, c) = \sum_{t \in \{1 \dots T\}} emissions(s_1, s_2, e, c, t) \cdot \ln(t)$$

where $emissions(s_1, s_2, e, c, t)$ is the emission value obtained from a Urban Mobility Simulator (e.g., using SUMO) at time step $t$ for a specific pair of streets, emission class, and congestion condition. To ensure that highly congested streets are likely to be avoided, a logarithmic weighting factor $ln(t)$ is introduced, emphasizing later stages of the simulation, and using time also as an indicator of potentially highly congested streets. Given a road (i.e., a sequence of connected streets), the overall emission risk is computed by summing all the local risks for each pair of streets. More formally, the pollution risk associated with assigning vehicle $v$ to route $r$ is defined as:

$$risk(v, r) = \sum_{(s_1, s_2) \in r} risk_s(s_1, s_2, e, c)$$

After the Preprocessing a Search phase is introduced: its objective is to define a set of candidate solutions, which will be then provided to the optimization phase. The Optimisation module was our second contribution with respect to the initial framework. Given the set of candidate routes produced by the search step, the Optimisation step is responsible for selecting the most suitable route to assign to each vehicle. In our approach, for each new vehicle entering the network, a new route is provided, taking into account all routes already assigned to previously scheduled vehicles. As in [34], we adopted ASP [35] in the underlying approach for the optimisation phase. The pollution risk associated with a segment, previously defined as the function $risk_s(s_1, s_2, e, c)$, is represented as:

```
risk(S1,S2,EC,CL,RV)
```

where `S1` and `S2` are consecutive streets, `EC` identifies the emission class of the vehicle, `CL` reflects the local traffic condition, and `RV` quantifies the pollution risk. To define the objective function for minimising emission risk associated with a specific candidate route (given by the Search module), the following optimation in term of ASP weak constraint has been introduced:

```
:~ vehicle(V), solutionRoute(V,R), emissionClass(V,EC),
indexStreetOnRoute(S1,R,I),indexStreetOnRoute(S2,R,I+1), enter(V,S1,T),
congestion(S1,T,CL), risk(S1,S2,EC,CL,RV). [RV@1]
```

This rule penalises the selection of routes with a high pollution risk, where `RV` is a numeric value that measures the pollution risk. The optimiser will favour solutions that minimise this objective function, which are solutions with a lower pollution risk.

To evaluate the effectiveness of our proposed approach, we employed the SUMO traffic simulator and examined two distinct urban environments: Bologna and Milton Keynes. In both scenarios, our approach performed well. The results were particularly strong in the Milton Keynes scenario, likely due to the wider range of possible routes from a single starting point, confirming the ability of the designed optimisation approaches in reducing emissions.

## 3. PDDL+ in Urban Traffic Distribution

This section reviews PDDL+-based approaches to urban traffic distribution. Our research has focused on designing planning models that capture diverse requirements, which shape the resulting formulations. Moreover, it has also focused on developing domain-specific heuristics to efficiently solve these models within the "planning as heuristic search" approach [36]. The models described here can be roughly characterized as *flexible* or *deployable*, according to the needs that guided their design.

Flexible models maximize controller freedom by admitting a large set of valid signal plans. This class of models is useful for exploring the upper-bound performance in an idealized setting. In particular, they highlight the potential gains that can be achieved by upgrading the underlying control infrastructure and removing specific technological constraints. However, prioritizing flexibility can undermine the deployability of the approach in relation to the target UTC infrastructure. Indeed, these models abstract away controller capabilities, e.g., limits on how frequently configurations may change, and let the solver dynamically generate the most suitable configurations. Therefore, signal plans can prove ineffective in practice or fail traffic authority validation. From a computational perspective, these expressive models are typically more challenging to solve and require the use of domain-dependent heuristics to obtain efficient performance in plan signal generation.

In contrast, deployable models are designed with a target UTC infrastructure in mind, for example, SCOOT [31], widely adopted in the UK. They restrict control to a limited set of pre-validated signal configurations, typically uploaded at least one day in advance, and require all cycles to have identical durations to preserve network synchronization and green wave coordination. Moreover, they limit how often configurations may change over time. Consequently, signal plans derived from such models are more likely to be executable as-is on existing UTC systems, requiring little or no additional validation effort. Prioritizing deployability over flexibility, however, comes at the cost of reduced control and potential performance losses due to suboptimal use of green time. Deployable models, by significantly restricting the space of valid signal plans, are typically more manageable to solve and allow the use of domain-independent approaches.

All the models we studied share a common representation of the environment in terms of PDDL+ processes and events, which capture how the urban network evolves over time under the applied control. In particular, each link of the network is associated with a numeric variable, the occupancy, representing the number of vehicles present. Processes model vehicle flows across network links. Specifically, for each permitted movement defined by a stage of a cycle, there is a process that linearly decreases the occupancy of the corresponding incoming link and increases the occupancy of the associated outgoing link whenever the green phase for that stage is active. Events model the triggering of stages within the cycles, capturing the discrete transitions between different traffic light phases. Finally, actions influence signal configurations, indirectly affecting the occurrence of events and the evolution of processes.

In the following, while keeping the environmental representation fixed, we describe how the different models implement traffic signal control.

### 3.1. Flexible Models

In the first proposed model [22, 37], traffic control relies on the *switchPhase* action, which represents the planner's decision to change the state of an intersection. Each intersection is associated with a token that identifies which stage is currently running; executing *switchPhase* increases the token value and transfers the green phase to the next stage. The action is subject to temporal constraints: it can only be applied once the minimum green time has elapsed; reaching the maximum green time triggers an exogenous event that enforces the phase change. This model was solved using the PDDL+ planner UPMurphi [38], extended with a domain-specific heuristic to guide the exploration.

Whereas the first model allowed control of minimum and maximum stage durations, it did not enforce any constraint on the overall cycle length. To address this, the subsequent *Extend and Reduce* model (ExRe) we propose [39] introduced two actions, *extendStage* and *reduceStage*, which allow increasing or decreasing the duration of the current stage with a fixed granularity. By acting on global variables associated with the cycle under consideration, these actions ensure that the resulting configurations remain within specified minimum and maximum bounds for the overall cycle length.

To solve this model, we developed a domain-specific heuristic implemented on top of ENHSP [8] and combined it with a Greedy Best First Search (GBFS). In simulation, the resulting signal plans showed competitive performance on real-world scenarios, specifically on a major traffic corridor in the city of

Huddersfield, when compared with the reactive plans generated by the existing infrastructure.

The signal plans generated by the ExRe model, although competitive, are characterized by variable cycle lengths and frequent configuration changes, which make them undeployable. This limitation motivated the design of the following deployable models.

### 3.2. Deployable Models

In [23], we addressed the gap between theoretical traffic signal optimisation and real-world deployment by providing three new PDDL+ models that enable to generate deployable signal plans compatible with the UTC infrastructure. The three models, denoted as *Cycle by Cycle* (CbC), *Fixed Repetition* (FiRe), and *Variable Repetition* (VaRe), differ in their flexibility regarding when and how often predefined cycle configurations can be changed.

Specifically, CbC model allows the configuration of a junction to be selected at every cycle transition (during the intergreen phase of the cycle's final stage), therefore significantly increasing the number of decision points compared to other models. This behaviour is achieved by the *changeConfiguration* action, which switches a junction from one predefined cycle configuration to another when triggered at the end of a cycle.

FiRe model requires a selected configuration to be retained for a minimum number of cycles before allowing a change, reducing the number of decision points compared to the CbC model. A cycle counter tracks the number of completed cycles for the current configuration, and the *changeConfiguration* action is used to switch configurations only after the minimum cycle limit is reached.

VaRe model allows decisions on how many times a configuration repeats at each junction. The minimum number of repetitions can vary within a specified range and is set using the *changeLimit* action, which executes after *changeConfiguration* to assign the repetition count for that junction. Once this repetition limit is established, the handling of stage durations and cycle counts follows the FiRe model unchanged.

As a preliminary step, we compared the three models under different search strategies and heuristic configurations to identify the most suitable one. This analysis revealed that FiRe was the most promising candidate, particularly when solved using GBFS combined with the heuristic $h^{max}$. We then evaluated this configuration of FiRe on the same benchmark used to assess the ExRe [39]. Under this setting, FiRe produced signal plans that are better than those historically generated by SCOOT and proved competitive with the plans obtained from the ExRe model equipped with a domain-specific heuristic.

### 3.3. Balancing Flexibility and Deployability

Although deployable models, and in particular FiRe, offer clear benefits, such as generating signal plans that are almost ready for execution, they are strongly limited in expressiveness, since they rely on a fixed set of precompiled configurations. For this reason, in a recent work [40], we investigated a model that lies in between in terms of flexibility and deployability, namely Trade. Our goal was to increase the expressiveness of deployable models while preserving a certain degree of deployability. Intuitively, Trade works by keeping the cycle length fixed while allowing the redistribution of green time among its stages. In practice, this is achieved through the *tradeTime* action, which, during intergreen periods, transfers a small amount of time from one future stage to another. The model is parameterised by two factors: the granularity of time that can be traded between stages and the maximum number of trades allowed per cycle. From an experimental perspective, we observed that, when solved with the same heuristic originally designed for ExRe, the Trade model achieves performance comparable to ExRe while maintaining a higher degree of deployability.

So far, we have characterised the presented models in qualitative terms. However, the distinction between deployable and flexible models can also be formally understood by relating the sets of valid signal plans that they generate for the same traffic scenario. Consider a traffic scenario encoded with the various models, and let *plans*($M$) denote the set of valid signal plans under model $M$. Roughly, the spaces of valid plans can be expressed as *plans*(FiRe) ⊆ *plans*(VaRe) ⊆ *plans*(CbC) ⊆ *plans*(Trade) ⊆ *plans*(ExRe).

## 4. Discussion and Conclusion

In this paper, we have overviewed the recent problems in the context of urban traffic distribution and control modelled and solved with ASP and PDDL+, separately. However, ASP and PDDL+ can be also combined to exploit the benefits provided by both approaches to solve the Traffic Signal Optimization problem. For instance, in [30], we showed how the *clingcon* encoding can be used to improve the quality of solutions returned by the PDDL+ approach. We implemented an automated pipeline to exploit the synergies of the two approaches as follows: from the solution found by the PDDL+ planner, the values of *counter* at a given horizon of each target link can be extracted. This information can be encoded as ASP atoms that appear in constraints forcing *clingcon* to return a solution that is strictly better than the PDDL+ one. In this way, thanks to PDDL+, we have the guarantee to have a solution quickly, while the subsequent application of ASP allows for trying to improve it (which was observed for almost half of the instances in the benchmark).

## Declaration on Generative AI

During the preparation of this work, the authors may have used ChatGPT in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References


[1] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Commun. ACM 54 (2011) 92–103. URL: https://doi.org/10.1145/2043174.2043195. doi:10.1145/2043174.2043195.

[2] M. Fox, D. Long, Modelling mixed discrete-continuous domains for planning, J. Artif. Intell. Res. 27 (2006) 235–297.

[3] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, T. Schaub, ASP-Core-2 input language format, Theory and Practice of Logic Programming 20 (2020) 294–309.

[4] M. Alviano, G. Amendola, C. Dodaro, N. Leone, M. Maratea, F. Ricca, Evaluation of disjunctive programs in WASP, in: Proc. of LPNMR, volume 11481 of *LNCS*, Springer, 2019, pp. 241–255.

[5] M. Gebser, B. Kaufmann, T. Schaub, Conflict-driven answer set solving: From theory to practice, Artificial Intelligence 187 (2012) 52–89.

[6] C. Dodaro, G. Mazzotta, F. Ricca, Blending grounding and compilation for efficient ASP solving, in: KR, 2024.

[7] G. Mazzotta, F. Ricca, C. Dodaro, Compilation of aggregates in ASP systems, in: Proc. of AAAI 2022, AAAI Press, 2022, pp. 5834–5841.

[8] E. Scala, P. Haslum, S. Thiébaux, M. Ramirez, Subgoaling Techniques for Satisficing and Optimal NumericPlanning, J. Artif. Intell. Res. 68 (2020) 691–752.

[9] A. F. Bocchese, C. Fawcett, M. Vallati, A. E. Gerevini, H. H. Hoos, Performance robustness of AI planners in the 2014 international planning competition, AI Commun. 31 (2018) 445–463.

[10] M. Gebser, M. Maratea, F. Ricca, The design of the sixth answer set programming competition - - report -, in: F. Calimeri, G. Ianni, M. Truszczynski (Eds.), Proc. of LPNMR 2015, volume 9345 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 531–544.

[11] M. Gebser, A. Ryabokon, G. Schenner, Combining heuristics for configuration problems using answer set programming, in: F. Calimeri, G. Ianni, M. Truszczynski (Eds.), Proc. of LPNMR 2015, volume 9345 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 384–397.

[12] R. Taupe, G. Friedrich, K. Schekotihin, A. Weinzierl, Solving configuration problems with ASP and declarative domain specific heuristics, in: M. Aldanondo, A. A. Falkner, A. Felfernig, M. Stettinger (Eds.), Proc. of CWS/ConfWS 2021, volume 2945 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 13–20.

[13] M. Gebser, P. Obermeier, T. Schaub, M. Ratsch-Heitmann, M. Runge, Routing driverless transport vehicles in car assembly with answer set programming, Theory Pract. Log. Program. 18 (2018) 520–534.

[14] C. Dodaro, G. Galatà, M. K. Khan, M. Maratea, I. Porro, An ASP-based solution for operating room scheduling with beds management, in: P. Fodor, M. Montali, D. Calvanese, D. Roman (Eds.), Proc. of RuleML+RR, volume 11784 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 67–81.

[15] M. Cardellini, P. D. Nardi, C. Dodaro, G. Galatà, A. Giardini, M. Maratea, I. Porro, A two-phase ASP encoding for solving rehabilitation scheduling, in: S. Moschoyiannis, R. Peñaloza, J. Vanthienen, A. Soylu, D. Roman (Eds.), Proc. of RuleML+RR, volume 12851 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 111–125.

[16] G. Grasso, S. Iiritano, N. Leone, V. Lio, F. Ricca, F. Scalise, An asp-based system for team-building in the gioia-tauro seaport, in: M. Carro, R. Peña (Eds.), Proc. of PADL 2010, volume 5937 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 40–42.

[17] C. Dodaro, G. Galatà, M. Maratea, I. Porro, Operating room scheduling via answer set programming, in: AI*IA, volume 11298 of *LNCS*, Springer, 2018, pp. 445–459.

[18] C. Dodaro, G. Galatà, M. Maratea, I. Porro, An ASP-based framework for operating room scheduling, Intelligenza Artificiale 13 (2019) 63–77.

[19] P. Bruno, F. Calimeri, C. Marte, M. Manna, Combining deep learning and asp-based models for the semantic segmentation of medical images, in: Proc. of RuleML+RR, volume 12851 of *LNCS*, Springer, 2021, pp. 95–110.

[20] P. Cappanera, M. Gavanelli, M. Nonato, M. Roma, Logic-based Benders decomposition in answer set programming for chronic outpatients scheduling, Theory and Practice of Logic Programming 23 (2023) 848–864.

[21] M. Alviano, R. Bertolucci, M. Cardellini, C. Dodaro, G. Galatà, M. K. Khan, M. Maratea, M. Mochi, V. Morozan, I. Porro, M. Schouten, Answer set programming in healthcare: Extended overview, in: IPS and RCRA 2020, volume 2745 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020.

[22] M. Vallati, D. Magazzeni, B. D. Schutter, L. Chrpa, T. L. McCluskey, Efficient Macroscopic Urban Traffic Models for Reducing Congestion: A PDDL+ Planning Approach, in: Proc. of AAAI 2016, 2016, pp. 3188–3194.

[23] A. E. Kouaiti, F. Percassi, A. Saetti, T. L. McCluskey, M. Vallati, PDDL+ models for deployable yet effective traffic signal optimisation, in: Proc. of ICAPS, AAAI Press, 2024, pp. 168–177.

[24] D. Aineto, E. Scala, E. Onaindia, I. Serina, Falsification of Cyber-Physical Systems Using PDDL+ Planning, in: Proc. of ICAPS, 2023, pp. 2–6.

[25] M. Cardellini, M. Maratea, M. Vallati, G. Boleto, L. Oneto, In-Station Train Dispatching: A PDDL+ Planning Approach, in: Proc. of ICAPS, 2021, pp. 450–458. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/15991.

[26] J. J. Kiam, E. Scala, M. R. Javega, A. Schulte, An AI-Based Planning Framework for HAPS in a Time-Varying Environment, in: Proc. of ICAPS, 2020, pp. 412–420.

[27] F. K. Alaboud, A. Coles, Personalized Medication and Activity Planning in PDDL+, in: Proc. of ICAPS, 2019, pp. 492–500.

[28] W. Piotrowski, Y. Sher, S. Grover, R. Stern, S. Mohan, Heuristic search for physics-based problems: Angry birds in PDDL+, in: Proc. of ICAPS, 2023, pp. 518–526.

[29] R. Bertolucci, A. Capitanelli, M. Maratea, F. Mastrogiovanni, M. Vallati, Automated planning encodings for the manipulation of articulated objects in 3d with gravity, in: M. Alviano, G. Greco, F. Scarcello (Eds.), Proc. of AI*IA 2019, volume 11946 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 135–150.

[30] A. Tarzariol, M. Maratea, M. Vallati, A casp-based solution for traffic signal optimisation, 2025. URL: https://arxiv.org/abs/2507.19061. `arXiv:2507.19061`.

[31] H. Taale, W. Fransen, J. Dibbits, The second assessment of the SCOOT system in Nijmegen, in: IEEE Road Transport Information and Control, 21-23, 1998.

[32] M. Banbara, B. Kaufmann, M. Ostrowski, T. Schaub, Clingcon: The next generation, Theory and Practice of Logic Programming 17 (2017) 408–461. doi:10.1017/S1471068417000138.

[33] S. Fiorentino, C. Dodaro, M. Maratea, M. Vallati, Ai-enabled connected autonomous vehicles sustainable routing in urban areas, in: 2025 IEEE 28th International Conference on Intelligent Transportation Systems, IEEE, United States, 2025. URL: https://ieee-itsc.org/2025/, 28th International Conference on Intelligent Transportation Systems, ITSC 2025 ; Conference date: 18-11-2025 Through 21-11-2025.

[34] M. Cardellini, C. Dodaro, M. Maratea, M. Vallati, A framework for risk-aware routing of connected vehicles via artificial intelligence, in: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2023, pp. 5008–5013.

[35] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, T. Schaub, ASP-Core-2 input language format, Theory Pract. Log. Program. 20 (2020) 294–309. URL: https://doi.org/10.1017/S1471068419000450. doi:10.1017/S1471068419000450.

[36] B. Bonet, H. Geffner, Planning as heuristic search, Artif. Intell. 129 (2001) 5–33.

[37] S. Franco, A. Lindsay, M. Vallati, T. L. McCluskey, An innovative heuristic for planning-based urban traffic control, in: Proc. of ICCS (1), volume 10860 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 181–193.

[38] G. D. Penna, D. Magazzeni, F. Mercorio, A universal planning system for hybrid domains, Appl. Intell. 36 (2012) 932–959.

[39] F. Percassi, S. Bhatnagar, R. Guo, K. McCabe, T. L. McCluskey, M. Vallati, An efficient heuristic for ai-based urban traffic control, in: Proc. of MT-ITS, IEEE, 2023, pp. 1–6.

[40] S. Castellanos, F. Percassi, M. Vallati, Improved Models for Automated Planning-based Urban Traffic Control, in: Proc. of MT-ITS, IEEE, 2025.