Against the Clock: Lessons Learned by Applying Temporal Planning in Practice*

Andrea Micheli[1][0000-0002-6370-1061]

Fondazione Bruno Kessler, Trento, Italy
amicheli@fbk.eu

Abstract. Automated Planning is a foundational area of AI research, focusing on the automated synthesis of courses of actions to achieve a desired goal within a formally-modeled system. When dealing with time and temporal constraints, this problem is known as Temporal Planning. In this paper, we will present our research on the application of temporal planning to real-world scenarios, and highlight the open research directions in this field. Starting from a series of projects in different application domains – including robotics, manufacturing, and logistics – we will explore key challenges encountered, the (sometimes hard) lessons learned, and the techniques, tools, and methodologies that have emerged from these efforts. Additionally, we will introduce and discuss preliminary results on applying Reinforcement Learning techniques to tailor temporal planners to specific application contexts.

Keywords: Automated Temporal Planning · Planning and Scheduling · Applications of Planning.

1 Introduction

Automated planning is a historical research area in Artificial Intelligence focusing on the synthesis of "plans" to achieve specified goals in formally modeled systems [10]. Several concrete problems have been defined by requiring certain formulations of plans or by limiting the system models to some expressiveness class. As a motivating example, consider a fleet of robots that can move among a set of locations and perform logistic operations (such as picking objects, transporting and depositing cargo); further suppose that each operation has known duration and that some operations might consume resources (such as the robot batteries). Temporal planning formalisms are designed to faithfully model a situation like this[1] and allow the automated synthesis of courses of actions to achieve a desired objective, possibly within a specified deadline.

This paper surveys the approaches to Temporal Planning developed in recent years in the Planning, Scheduling and Optimization unit (PSO) that I lead at

* This paper summarizes the contents of the homonymous AIxIA 2024 invited talk.
[1] Disregarding, for the sake of computational efficiency, uncertainties in the robot and environment behaviors.

Fondazione Bruno Kessler[2]. We have in fact participated in a number of research and technology transfer projects concerning planning and scheduling: we will here discuss the methodology developed in the most significant ones, focusing on the practical challenges that generated research ideas, and introducing the reusable assets we implemented to address them.

We will first present the iLAADR project, aiming at automating the intra-logistics operations in a factory, which served as a real-world use-case for the advanced modeling features of the ANML language. These features are rarely supported by off-the-shelf automated planners and motivated a line of theoretical and practical research to offer them effectively. Then, we will summarize our collaboration in the area of underwater robotics, where the planner is tasked to decide and schedule the operations needed to perform a surveillance mission and to safely overcome problematic situations. In this context, the optimization of resources is of paramount importance: we tackled the open problem of Optimal Temporal Planning with an approach grounded in Optimization Modulo Theory (OMT) [21]. Third, we motivate and report on the technological effort to bring a convenient and reusable platform for the modeling, manipulation and solving of automated planning problems. We built and validated such a platform in the context of the AIPlan4EU project; other project partners, us, and also third parties are currently re-using this open-source infrastructure for new projects and research. Finally, we discuss the extremely challenging MAIS project, aimed at the automated control of electroplating production factories, which inspired an ambitious line of research focused on the combination of Reinforcement Learning and Automated Planning for the synthesis of specialized planners.

Structure of the paper. This paper is structured as follows. We first report some minimal background notions needed to set the stage of the paper; then, we discuss the four mentioned projects with the technical results that emerged and lessons learned in four separate sections. Finally, we draw our conclusions in section 7.

2 Background

Before delving into the projects and the research ideas, we provide a brief overview of the area of temporal planning, relevant to explain the contributions reported in this paper.

Temporal planning is a vast area of Artificial Intelligence and over the years a number of models and techniques have been proposed within it. At the core of the problem lays the interplay between deciding which actions/activities need to be performed to reach a desired goal (the planning part) and choosing an appropriate timing (or a set of possible timings) for such activities (the scheduling part). The combination of planning and scheduling is what makes the problem hard and challenging. Being at the border of planning and scheduling implies

[2] `https://pso.fbk.eu`

that often a problem could be addressed both by temporal planning and scheduling techniques: for example, if the number of possible activities to choose from could be bound, a problem might be equivalently framed as a scheduling problem with optional activities or as a temporal planning problem. The best approach is not always clear and strongly depends on the problem size and the kind and number of constraints.

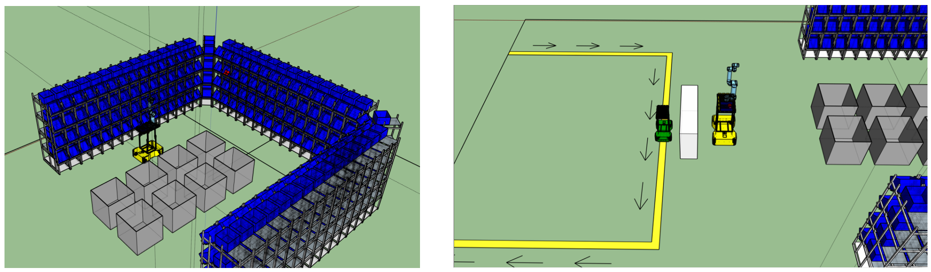
In the area of temporal planning, two major formalisms emerged: action-based planning [3,2,27] and timeline-based planning [9,1,26]. In extreme simplification, action-based approaches and languages augment classical planning formalisms with time and temporal constraints, while timeline-based planning consists in augmenting scheduling techniques to constrain the possible instantiations of activities and their constraints.

In our work, we focused on action-based planning. In this area, two major languages have been proposed, namely PDDL 2.1 [6] and ANML [22]. PDDL 2.1 is by far the most common language for temporal planners: it extends the well-known PDDL language by allowing "durative-actions", i.e. actions that correspond to an interval of time and can have conditions and effects either at the beginning or at the ending, plus an invariant (overall) condition. In addition to temporal constraints, PDDL 2.1 also supports continuous change, but for the sake of this paper we limit ourselves to the temporal and numeric fragment (generally called PDDL 2.1 Level 3). As shown by Cushing et al., PDDL 2.1 is a temporally-expressive language, meaning it can express problems in which all valid plans require actions to run concurrently [4]. Despite its widespread adoption and its theoretical expressiveness, PDDL 2.1 is not always easy to use in practical settings and requires several compilation constructions to express interesting behaviors such as Intermediate Conditions and Effects (ICE) [23,8]. The Action Notation Modeling Language (ANML) is an alternative action-based language proposed by NASA, designed to be more user-friendly by providing higher-level constructs such as a richer type system, ICE, structured types, richer temporal constraints and hierarchical structures. In the following, we will motivate how we chose to use ANML over PDDL 2.1 for several projects and how we formally studied the complexity of some of the features offered by either languages. For the sake of completeness, other action-based languages exists: PDDL+ [7] is an evolution of PDDL 2.1 that retains the durative-action concept, but focuses on continuous and exogenous processes and events; NDL [20] is another temporally-expressive language with an explicit notion of resources and where actions are not intervals, but rather events with conditions and effects scheduled in the future.

3 iLAADR: Temporal Planning Expressiveness

The first project experience we report on is iLAADR[3], a project funded by the European Institute of Technology aiming at the automation of intra-logistic

[3] `https://robotik.dfki-bremen.de/en/research/projects/ilaadr`



(a) (b)

Fig. 1: 3D renderings of the iLAADR scenario. (a) the robot navigates the warehouse to pick the components of a kit. (b) the kit is exchanged between two robots.

operations using robots. The use-case of the project, provided by a leading automotive manufacturer, is centered on "kitting" operations: a robot equipped with a robotic arm is tasked to pick a set of pieces from a small warehouse (Figure 1a) and to bring it to a second automated guided vehicle serving the production line (Figure 1b) just in time for a human operator to get the kit for the specific car being manufactured in that moment. Consider the following example: the car currently being assembled in the production line is red, whereas the next one is green. This means that the operator needs to receive a "red kit" in time for completing the red car assembly and before receiving the "green kit". In this project, we were in charge of the automated planning operations, consisting in the synthesis of plans for each of the robots involved in the scenario.

We focused on the faithful modeling of the system constraints and developed an automated procedure to construct planning problems from the factory Warehouse Management System (WMS) and Manufacturing Execution System (MES). The key challenges we encountered concerned the complexity and maintainability of the models, and the scalability of tools on such models.

We tried to modeled the problem in different languages, and we soon realized that the "queue of orders" is not easy (although possible) to model in PDDL 2.1: to order objectives in time, one needs to use "Timed Initial Literals" [5] (which are not part of PDDL 2.1, but are supported by several planners) and monitor fluents to record what has been achieved so far. Modeling the scenario with ANML is much easier: one can use object fluents and structured types to represent the sequence of orders[4] and ANML natively supports both absolute timing constraints for timed goals as well as Intermediate Conditions and Effects (ICE) to express constraints happening during the execution of the activities. Maintaining a high-level representation, close to the problem domain, allows the creation of domain-specific heuristics (in the project, we constructed

[4] In particular, we could define a list of n orders as a fluent with a numeric parameter in the domain $\{1, \ldots, n\}$ and type $Order$. This essentially represents an array of variables that can then be filled by actions in a very natural way.

a simple domain-dependent goal-counting heuristic that was extremely effective in practice): these can be embedded in the final planner together with standard heuristics in a portfolio approach.

The use of the ANML language in this project motivated a line of research that is both practical and theoretical. We developed a planner, called TAMER[5], for solving planning problems modeled in ANML and we focused on the problem of how to embed ICE as a native feature of our planner; we extended the decoupled approach used by the POPF [2] and other planners to support these features. The resulting planner is shown [27] to be much more efficient than state-of-the-art competitors on problems having ICE with respect to different ways this feature can be compiled into PDDL 2.1 [23,8].

In a theoretical line of papers, we set ourselves to understand the computational complexity of temporal planning when time is interpreted over a dense domain (as prescribed by the ANML semantics). Interestingly, we discovered that temporal planning is not harder than classical planning (PSPACE-complete) if we forbid self-overlapping of ground actions: this means that two instances of the same action with the same parameters are not allowed to overlap in time, although the same action can be repeated multiple times, as needed. Instead (and surprisingly), if a separation of a known time quantum (generally referred to as ϵ) is enforced between interfering events (e.g., between an effect setting a fluent to true and a condition requiring the fluent to be true) the temporal planning problem is shown to be EXPSPACE-complete. When no ϵ-separation is assumed and actions can self-overlap, the problem becomes undecidable [11]. These results gave a clear theoretical view on the role of self-overlapping and ϵ-separation in the realm of temporal planning inspiring new planners such as CTP, the first decision procedure for temporal planning in dense time without action self-overlapping [19]. Finally, we also proved that advanced temporal planning features, such as conditional effects and ICE do not impact these core results [12].

4 HyDrone: Optimal Temporal Planning

A second technology-transfer project, named HyDrone[6], concerned the use of automated planning for the synthesis of mission plans and recovery procedures for an underwater surveillance drone. The characteristics of the drone have been presented in [24], as well as the general architecture of the automated decision system we designed and implemented. Focusing on the computational challenges of the project, we needed a system capable of optimizing resources in addition to finding valid plans. In the project, we had three major quantities to consider and optimize: the total mission time (i.e., the makespan of the plan), the data being produced (as the system has limited data storing capacity) and, naturally, the battery level to allow for safer and more efficient operation of the robot. In this respect, we started exploring the area of optimal temporal planning, consisting

[5] `https://tamer.fbk.eu`
[6] `https://pso.fbk.eu/articles/hydrone`

in finding a valid plan that is optimal with respect to a specified cost function. The problem itself is extremely hard, with very few approaches in the literature either aimed at minimizing the plan makespan or limited to specific problem formulations.

We took inspiration from this challenge to tackle the optimal temporal planning problem in a principled way. We started from the work by Leofante et al. for optimal numeric planning via Optimization Modulo Theory (OMT) [13], and we generalized it to the case of temporal planning. The basic idea behind the approach is to use a bounded encoding into OMT that can capture in a single formula both the concrete plans of the system within the bound (in terms of number of steps) as well as an abstraction of plans that are longer (and might not exist); if the OMT solver returns a concrete plan while optimizing the objective function, we can prove that such a plan is globally optimal, because no other plan, however long, can have a better objective value [17].

Our generalization to temporal planning is far from trivial, because temporal planning (unlike classical and numeric planning) requires to tackle "future commitments": if an action is started, but not yet completed, one must take care of the consequences of the inevitable termination of such action. In this line of research, we also devised specialized encodings for the optimization of the makespan and we fully support the linear combination of makespan and action cost objectives in our planner [18].

5 AIPlan4EU: Making Planning Easier to Use

Among the major hindrances for practitioners wanting to explore the use of automated planning (and temporal planning in particular), we note the steep learning curve needed to gain familiarity with the modeling principles, the heterogeneity of input languages and dialects, and the diverse technical characteristics of available tools. Moreover, real-world applications require the use of planning as a component integrated in a wider ICT solution, not as a standalone software as it is the case for most planning tools. In the AIPlan4EU project[7], we worked to mitigate these issues by providing a convenient programmatic interface to model, manipulate and ultimately solve planning problems of various kind.

In practice, we developed a Python library, called Unified Planning (UP)[8], for representing, manipulating and solving classical, numerical, temporal, hierarchical and other kinds of planning problems. A user can either model a problem directly using the provided Python API, or by employing one of the provided parsers (PDDL 2.1 and ANML). Since UP is library, it is easy to use data sources (e.g. the WMS of a plant or the sensors of a robot) to dynamically construct the planning problems. UP also allows the manipulation of the problem, for example by compiling away some modeling features such as disjunctive preconditions, or performing the grounding of the problem. Moreover, the library offers a plug-in

[7] `https://aiplan4eu-project.eu`
[8] `https://github.com/aiplan4eu/unified-planning`

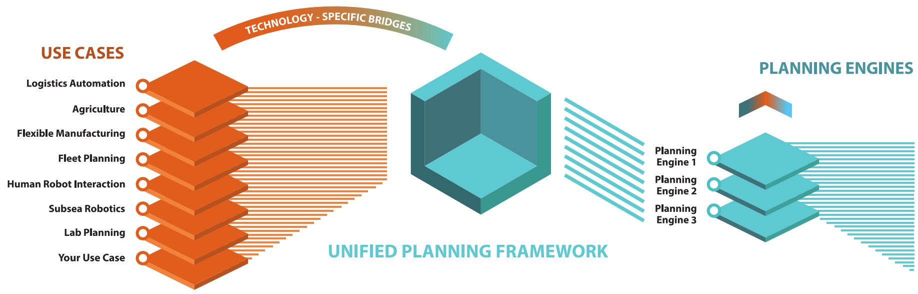


Fig. 2: Overview of the The Unified Planning framework.

system for interfacing external planning engines that can be used to further manipulate or solve a planning problem. Finally, the solution plans are exposed as Python objects for easy inspection and use of produced results.

In the project, we demonstrated the technology on a wide variety of use-cases and scenarios; in addition, project partners and third parties developed specialized libraries and tools for some application domains. We call such integrations "Technology-Specific Bridges" (TSB): they allow the easy re-use of the library in a certain ICT environment. To mention some examples, the Embedded Systems Bridge[9] permits the use of UP in a framework-independent robotic setting, while UP4ROS2[10] focuses on the Robotic Operating System integration. Figure 2 depicts the general, high-level architecture of the project solution.

In addition to simplifying technology transfer, this technical effort enables the exploration of advanced features, such as procedural modeling of effects or custom heuristics. The former consists of specifying the effects of an action as a Python function that can be executed but not inspected, whereas the latter allows to programmatically describe a domain-specific heuristic within the framework. Moreover, the library offers support for the simulation and validation of planning problems, which are essential ingredients for approaches combining planning with other technologies such as Reinforcement Learning. In fact, the library is currently being used as the basis for new research: we recently extended it to model Task and Motion Planning (TAMP) problems, exploiting the manipulation capabilities of the library to automate the refinements in a Benders decomposition schema [25].

6 MAIS: Specialization of Temporal Planners

The final project we report in this survey is called Mechanical Automation Integration System (MAIS). The context of the project is the automation of electroplating plants, where hoists are used to move the products being treated along

[9] `https://github.com/aiplan4eu/embedded-systems-bridge`
[10] `https://github.com/aiplan4eu/UP4ROS2`

a predefined sequence of chemical (and electro-chemical) baths. The goal of the project was to develop a planning solution capable of automating the decisions of when and where each hoist should move the products around to achieve the maximal throughput of the plant. We approached the problem in different ways and with diverse technologies, but no planner nor scheduler was capable of getting close to the real-world scale of the problem we faced. In the scheduling literature, a simplified version of the problem is called Hoist Scheduling Problem (HSP) and is shown to be strongly NP-hard even with very strong assumptions [14].

The complexity of the problem lies in the interaction of planning decisions (i.e., where to send the hoist and which operation to perform) with the scheduling constraints emerging from the very precise timings each piece can stay in each bath. Practically, in driving the hoists to bring many pieces into production, we are creating a lot of deadlines for taking the pieces out of the respective bath and such deadlines quickly become unachievable due to the finite speed of the hoist. A search-based approach discovers these constraints by extensive search, but fails to effectively back-jump to the root-cause of a problem. Instead, a scheduling-based approach quickly explodes due to the very high number of hoists movements involved in the problem. Moreover, it is very hard to model the spatial constraints emerging from the relative positioning of the hoists in scheduling.

The solution we settled on was developing a domain-specific planner embedding lots of domain knowledge into the search, together with a strategy to combine solutions for parts of the plant into a global one. To cope with the deadlines emerging from the electroplating process, we devised a method to impose long-term constraints in a heuristic-search approach.

Our solution was adequate for the MAIS project, but we are unsatisfied with the generality of the approach; yet, this project was very instrumental in terms of lessons learned. First, we realized that the constraints expressible in either PDDL 2.1 or ANML are too "local" for some problems, whereas scheduling often requires expressing global constraints. As a first step to address these limitations we developed the TPACK planner [15], which can express temporal constraints as a quantified logic over time points, allowing the user to express complex global constraints such as the electroplating "recipes" of the MAIS project.

The second, and perhaps more radical, research idea stemmed from this project consists in tackling the problem of automatically specializing a planner for a certain domain: in fact, during the MAIS project we had to manually adapt our planner to the characteristics of the domain and we embedded in the planner heuristic the knowledge gathered from domain experts. We started working on this idea using Reinforcement Learning (RL) methods to synthesize domain-specific heuristics from a simulator of the distribution of planning problems of interest. Importantly, and differently from other approaches in the state of the art, we do not assume that a set of example plans is given. Instead, we take as input a set of planning problems that is intended to be a representative sample of the problem expected at run-time, and we use RL to devise a general policy for such a distribution. Then, instead of using the policy directly, we convert it

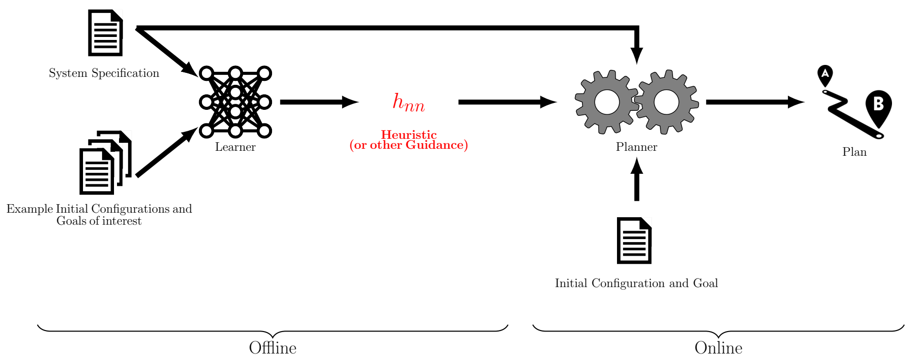


Fig. 3: Overview of the general learning schema for the specialization of temporal planning by synthesizing heuristic guidance. The offline phase is devoted to the synthesis of a guidance artifact, which is the used online by the planner to solve planning problems.

into a planning heuristic to balance the exploitation of learned information and systematic search, thus retaining the formal guarantees of a temporal planner. The general schema of this solution is depicted in Figure 3, and our initial solution beats our purely-symbolic planner on some domains [16]. This idea has been proposed and articulated in my own ERC project, called STEP-RL[11] (Specializing TEmporal Planning using Reinforcement Learning), which will focus on exploring, in a principled and well-founded way, the combination of temporal planning and reinforcement learning.

7 Conclusions

In this paper, we surveyed a series of works inspired by challenges emerged in a variety of technology-transfer projects centered on the theme of temporal planning we have been involved in. Considering both theoretical and practical aspects, we worked to extend the applicability of temporal planning also to make it more usable by practitioners.

Our quest is still far from over, as many challenges still need more research to be addressed; because of this, we are currently working along different lines. First of all, we are expanding our approach using RL in combination with planning along two major directions. The first concerns the learning of a residual of a planning heuristic, instead of learning a heuristic from scratch: the idea is to simplify the learning effort and exploit the volume of work in domain-independent planning. The second direction aims at learning macro-actions for temporal planning, which are "shortcuts" in the search space of a planner, from RL explorations.

[11] `https://pso.fbk.eu/articles/step-rl`

Furthermore, we are exploiting the UP library to advance our Task and Motion Planner to support time and temporal constraints. Finally, we are exploring the use of simulated entities as a mean to incorporate learned knowledge into a digital-twin model for space applications.

Acknowledgments

This work has been partially supported by the STEP-RL project funded by the European Research Council under GA n. 101115870.

I am very grateful to all my co-authors over the years in this line of research. In particular, Alessandro Cimatti, Nicola Gigante, Marco Roveri, Enrico Scala, and Alessandro Valentini hugely contributed with discussions, insights and technical effort.

References

1. Cesta, A., Cortellessa, G., Fratini, S., Oddi, A., Rasconi, R.: The APSI Framework: a Planning and Scheduling Software Development Environment. In: Working Notes of the ICAPS-09 Application Showcase Program. Thessaloniki, Greece (September 2009)
2. Coles, A.J., Coles, A., Fox, M., Long, D.: Forward-chaining partial-order planning. In: ICAPS 2010 (2010)
3. Coles, A., Fox, M., Long, D., Smith, A.: Planning with problems requiring temporal coordination. In: AAAI (2008)
4. Cushing, W., Kambhampati, S., Mausam, Weld, D.S.: When is temporal planning really temporal? In: IJCAI 2007 (2007)
5. Edelkamp, S., Hoffmann, J.: PDDL2.2: The language for the classical part of the 4th international planning competition. Tech. rep., Technical Report 195, University of Freiburg (2004)
6. Fox, M., Long, D.: PDDL2.1: An extension to PDDL for expressing temporal planning domains. Journal of artificial intelligence research (2003)
7. Fox, M., Long, D.: Modelling mixed discrete-continuous domains for planning. Journal of Artificial Intelligence Research (2006)
8. Fox, M., Long, D., Halsey, K.: An investigation into the expressive power of PDDL2.1. In: ECAI 2004 (2004)
9. Frank, J., Jónsson, A.: Constraint-based Attribute and Interval Planning. Constraints (2003)
10. Ghallab, M., Nau, D.S., Traverso, P.: Automated planning - theory and practice. Elsevier (2004)
11. Gigante, N., Micheli, A., Montanari, A., Scala, E.: Decidability and complexity of action-based temporal planning over dense time. Artif. Intell. **307**, 103686 (2022). `https://doi.org/10.1016/j.artint.2022.103686`, `https://doi.org/10.1016/j.artint.2022.103686`
12. Gigante, N., Micheli, A., Scala, E.: On the expressive power of intermediate and conditional effects in temporal planning. In: Kern-Isberner, G., Lakemeyer, G., Meyer, T. (eds.) Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel, July 31 - August 5, 2022 (2022), `https://proceedings.kr.org/2022/18/`

13. Leofante, F., Giunchiglia, E., Ábrahám, E., Tacchella, A.: Optimal planning modulo theories. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. pp. 4128–4134. ijcai.org (2020). https://doi.org/10.24963/IJCAI.2020/571, https://doi.org/10.24963/ijcai.2020/571
14. Manier, M.A., Bloch, C.: A classification for hoist scheduling problems. International Journal of Flexible Manufacturing Systems **15**, 37–55 (2003)
15. Micheli, A., Scala, E.: Temporal planning with temporal metric trajectory constraints. In: AAAI 2019. pp. 7675–7682 (2019)
16. Micheli, A., Valentini, A.: Synthesis of search heuristics for temporal planning via reinforcement learning. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. pp. 11895–11902. AAAI Press (2021). https://doi.org/10.1609/AAAI.V35I13.17413, https://doi.org/10.1609/aaai.v35i13.17413
17. Panjkovic, S., Micheli, A.: Expressive optimal temporal planning via optimization modulo theory. In: Williams, B., Chen, Y., Neville, J. (eds.) Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. pp. 12095–12102. AAAI Press (2023). https://doi.org/10.1609/AAAI.V37I10.26426, https://doi.org/10.1609/aaai.v37i10.26426
18. Panjkovic, S., Micheli, A.: Abstract action scheduling for optimal temporal planning via OMT. In: Wooldridge, M.J., Dy, J.G., Natarajan, S. (eds.) Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada. pp. 20222–20229. AAAI Press (2024). https://doi.org/10.1609/AAAI.V38I18.30002, https://doi.org/10.1609/aaai.v38i18.30002
19. Panjkovic, S., Micheli, A., Cimatti, A.: Deciding unsolvability in temporal planning under action non-self-overlapping. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022. pp. 9886–9893. AAAI Press (2022). https://doi.org/10.1609/AAAI.V36I9.21225, https://doi.org/10.1609/aaai.v36i9.21225
20. Rintanen, J.: Discretization of temporal models with application to planning with SMT. In: Bonet, B., Koenig, S. (eds.) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 3349–3355. AAAI Press (2015). https://doi.org/10.1609/AAAI.V29I1.9644, https://doi.org/10.1609/aaai.v29i1.9644
21. Sebastiani, R., Tomasi, S.: Optimization modulo theories with linear rational costs. ACM Trans. Comput. Log. **16**(2), 12:1–12:43 (2015). https://doi.org/10.1145/2699915, https://doi.org/10.1145/2699915
22. Smith, D., Frank, J., Cushing, W.: The ANML language. In: KEPS 2008 (2008)
23. Smith, D.E.: The case for durative actions: A commentary on PDDL2.1. Journal of Artificial Intelligence Research (2003)

24. Tosello, E., Bonel, P., Buranello, A., Carraro, M., Cimatti, A., Granelli, L., Panjkovic, S., Micheli, A.: Opportunistic (re)planning for long-term deep-ocean inspection: An autonomous underwater architecture. IEEE Robotics Autom. Mag. **31**(1), 72–83 (2024). https://doi.org/10.1109/MRA.2024.3352810, https://doi.org/10.1109/MRA.2024.3352810
25. Tosello, E., Valentini, A., Micheli, A.: A meta-engine framework for interleaved task and motion planning using topological refinements. In: Proceeding of the European Conference in AI (to appear) (2024)
26. Umbrico, A., Cesta, A., Mayer, M.C., Orlandini, A.: Integrating resource management and timeline-based planning. In: ICAPS (2018)
27. Valentini, A., Micheli, A., Cimatti, A.: Temporal planning with intermediate conditions and effects. In: AAAI 2020 (2020)