# Investigating Domain-oriented Approaches to Optimization in Timeline-Based Planning

Riccardo De Benedictis, Gloria Beraldo, Amedeo Cesta and Gabriella Cortellessa

CNR - Italian National Research Council, ISTC

#### Abstract

Automated planning involves devising a sequence of actions or decisions to attain specific goals within defined constraints. Within the realm of automated planning, optimization techniques are employed to enhance plan quality. These techniques target the minimization or maximization of various parameters, such as time, cost, resource utilization, and other relevant criteria, all while ensuring plan feasibility and adherence to problem constraints.

Timeline-based planning has emerged as an effective alternative to classical planning, offering robust mechanisms for constraint handling and facilitating adaptive plan adjustments during execution. Despite its growing adoption, especially due to the complexity of reasoning caused by highly expressive languages, the exploration of optimization within the context of timeline-based planning remains underrepresented in the scientific community.

This paper delves into the uncharted territory of leveraging optimization techniques within timelinebased planning. Importantly, this investigation is conducted slightly modifying the planner itself; instead, it mostly involves the introduction of suitable operators into the planning problem domains. Through this investigation, we aim to shed light on the potential of optimizing timeline-based planning processes for enhanced plan efficiency and effectiveness.

#### Keywords

Automated Planning, Timeline-based Planning, Heuristic search, Optimization, Scheduling

# 1. Introduction

Preference-based planning is a branch of automated planning and scheduling that emphasizes the generation of plans while considering and attempting to fulfill a maximum number of user-specified preferences. In numerous problem domains, achieving a task can involve multiple sequences of actions, which are commonly referred to as plans. The quality of these plans can vary significantly, with some being more desirable due to factors such as cost-effectiveness, speed, and safety. When generating a plan for a particular problem, preference-based planners

IPS-RCRA-SPIRIT 2023: Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy. November 7-9th, 2023, Rome, Italy [1]

<sup>\*</sup>Corresponding author.

<sup>&</sup>lt;sup>†</sup>These authors contributed equally.

<sup>☆</sup> riccardo.debenedictis@istc.cnr.it (R. De Benedictis); gloria.beraldo@istc.cnr.it (G. Beraldo);

amedeo.cesta@istc.cnr.it (A. Cesta); gabriella.cortellessa@istc.cnr.it (G. Cortellessa)

**b** 0000-0003-2344-4088 (R. De Benedictis); 0000-0001-8937-9739 (G. Beraldo); 0000-0002-0703-9122 (A. Cesta); 0000-0002-9835-1575 (G. Cortellessa)

<sup>© 0 2023</sup> Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CEUR Workshop Proceedings (CEUR-WS.org)

take these user-defined preferences into consideration. Examples of software for preferencebased planning include PPLAN [2] and HTNPlan-P [3], which specialize in preference-based hierarchical task network (HTN) planning. The Planning Domain Definition Language (PDDL) introduces preferences and plan quality metrics in its third version [4] and, since then, different approaches have emerged for their management in classical planning [5, 6].

In this context, timeline-based planning [7] has emerged as a promising departure from conventional planning paradigms. By embracing the principles of partial-order planning [8], timeline-based planning yields plans that exhibit heightened adaptability during execution, in stark contrast to the rigidly ordered plans generated by solvers employing the aforementioned heuristics. While offering distinct advantages, timeline-based planners grapple with challenges attributed to the rich expressiveness of their formalisms, often resulting in performance bottlenecks.

This paper embarks on an exploration of a relatively uncharted territory: the realm of optimization within the domain of timeline-based planning. The essence of this endeavor lies in our commitment to achieve plan quality optimization without the need for substantial modifications to the underlying planner. Instead, we adopt and adapt existing heuristics, leveraging their strengths to enhance the planning process. Our approach involves the introduction of specific operators into the planning problem domains, strategically guiding the decision-making process towards more effective solutions.

# 2. Timeline-based planning

Timeline-based planning constitutes a form of deliberative reasoning which, in an integrated way, allows to carry out different forms of semantic and causal reasoning. This form of planning was first introduced in [9] and, since then, many solvers, relying on this approach, have been proposed like, for example, *IxTeT* [10], EUROPA [11], ASPEN [12], the TRF [13, 14] on which the APSI framework [15] relies and, more recently, PLATINUm [16, 17].

Theoretical work on timeline-based planning, such as [18, 11], focused on identifying connections with classical planning in PDDL [19]. *IxTeT* and TRF emphasized time and resource reasoning [20, 21]. CHIMP adopted a Meta-CSP approach with meta-constraints resembling timelines [22], while FAPE tightly integrated timeline-like structures with acting [23]. ANML combined HTN decomposition methods with the expressiveness of the timeline representation [24]. Timeline-based approaches often incorporate resource management capabilities, leveraging constraint-based methods [25, 26, 27, 28] for planning and scheduling integration.

Despite these approaches share similarities, the lack of a common formalization hindered effective comparison and combination of their features. This limitation made it challenging to identify strengths, weaknesses, and develop comprehensive solutions. To provide a reference point, we mainly refer to the formalization proposed in [29], which covers a significant portion of problems solvable by existing approaches.

Understanding timeline-based planning requires introducing fundamental concepts of *constraint networks* [30], consisting in a set of *variables* and a set of *constraints*. Variables have names and can take different values from their *domain*. The domain is initially defined and can evolve over time. Variables can be continuous (with infinite initial domains) or discrete

(with a finite number of values). Constraints restrict combinations of values for a set of variables. Assigning values to variables is called an evaluation, which is consistent if it satisfies all constraints. An evaluation is complete when it includes all variables.

In timeline-based planning, constraint networks provide the foundation for modeling and reasoning about the problem. The central concept in this paradigm, however, is the *timeline*, which represents a function of time over a specific domain. The timeline can be either discrete or continuous, and its domain can be symbolic or numeric. Numeric domains can further be categorized as discrete or continuous.

To standardize the representation of timelines and make the reasoning process independent of their nature, the concept of *tokens* is introduced. Tokens are expressions that derive values on timelines through a timeline extraction procedure. They provide a unifying element for consistent representation and reasoning. A token  $n(x_0, ..., x_i)_{\chi}$  consists of a predicate name n,  $x_0, ..., x_i$  parameters (temporal, symbolic, and numerical constants or variables), and a  $\chi$  class (fact or goal). The parameters, in particular, can be variables within a constraint network and, as such, can be constrained, thereby narrowing down the range of acceptable values to the desired ones. Constraints can be imposed among the token parameters, as well as between parameters and other variables, encompassing temporal, symbolic, and numerical relationships. Tokens provide a higher-level semantics by grouping variables into structured data, enabling reasoning for planners. A timeline, in these context, is a *global constraint* [30]) over the tokens, preventing undesired temporal overlaps.

The combination of tokens and constraints forms the primary data structure for representing plans in timeline-based planning: the *token network*. The token network serves as the primary representation of plans in timeline-based planning. It is manipulated throughout the reasoning process by adding constraints among token variables for consistency and applying rules to establish causality. Tokens are categorized as facts (inherently true) or goals (to be achieved), with causality defined by rules that outline the necessary conditions for goal achievement. Rules consist of a head (goal) and a body (requirements), which can include slave tokens, constraints, conjunctions, and *priced* disjunctions. The selection of disjuncts during the resolution process plays a critical role in determining the cost associated with the resulting plan. Essentially, these costs establish preferences among potential admissible solutions. Consequently, this paper places its primary emphasis on these costs, recognizing them as the key factor within our investigation.

Finally, a timeline-based planning *problem* consists of a set of typed objects, a set of rules, and a requirement. The typed objects are used to instantiate the initial domains of variables in the constraint network and token parameters. A solution to the problem is a token network that is consistent with the rules and satisfies the requirement. This means that the token network's variable evaluation is consistent with the constraints, and for each goal in the token network, the corresponding rule's body is present.

Consider, as an example, the travel planning scenario presented in Figure 1. In this scenario, the agent's starting point is location A, and its objective is to reach location D, which is not directly connected to A. To achieve this, the agent must pass through location B, accessible by either metro, with a cost of 7, or bus, with a cost of 10. Once at location B, it faces two alternatives: taking a direct train to destination D, with a cost of 5, or embarking on a two-step journey involving a train ride to the airport in C, with a cost of 5, followed



**Figure 1:** A planning problem for devising a travel itinerary from location A to location D. Since these locations are not directly linked, it becomes necessary to traverse through location B which can be reached by either metro or bus. From there, the options are to either board a train directly to destination D or opt for a two-step journey involving a train to the airport in C and then a plane to reach destination D.

by a flight to reach destination *D*, with a cost of 20. These costs, importantly, do not merely reflect travel times; instead, they represent preferences regarding the user's choice of transportation modes. In essence, timing constraints might compel the agent to utilize modes of transportation that may not align with their preferences. In this particular problem, we have a defined set of locations (A, B, C, and D) along with corresponding facts that denote the connections between these locations using different modes of transportation, including the associated travel times. Additionally, there is a fact such as  $At(l : A, s : origin)^1$  indicating the agent's initial position, a goal At(l : D) representing the desired destination, and a set of rules outlining various routes to reach a specific location. These rules are accompanied by constraints pertaining to the use of vehicles and limitations on travel duration. The rule for reaching a location is  $At(l, s, e) \leftarrow \{TakeBus(to : l, e : s)_g\}_{10} \lor \{TakeSubway(to : l, e : s)_g\}_7 \lor$  $\{TakeTrain(to : l, e : s)_g\}_5 \lor \{TakePlane(to : l, e : s)_g\}_{20}, indicating a cost of 10, 7, 5 and 20, re$ spectively, for each taken bus, subway, train and plane. The rule for taking, for example, a bus, instead, is  $TakeBus(from, to, s, e) \leftarrow BusLink(from : from, to : to, dur)_g \land [e - s \ge dur]$ . Finally, there is a single state-variable timeline, preventing the agent to be in different positions or in different means of transport at the same time. As you can see from the rules, these activities have a duration. In the event of a deadline, regardless of preferences, not all plans may be acceptable.

## 3. Optimization Techniques

Timeline-based solvers heavily rely on partial-order planning techniques [8], extending the definition of threats to include potential inconsistencies arising from timeline constraints. These solvers aim to identify *flaws* in the token network and utilize *resolvers* to address them. Flaws can be unachieved goals, threats, disjunctions, or unassigned variables. Resolvers are

<sup>&</sup>lt;sup>1</sup>We use the notation "var: expr" to indicate the direct assignment of an expression to a variable. In this case, the l variable indicates the location.

mechanisms designed to resolve specific types of flaws. They can involve applying rules, unifying semantically equivalent tokens, introducing ordering constraints, selecting options from disjunctions, or assigning values to unassigned variables. The main resolution principle involves systematically improving the token network by applying appropriate resolvers while maintaining the consistency of constraints until the token network is flaw-free. A *solution* to a timeline-based planning problem is a token network without flaws and consistent constraints.

Solving timeline-based planning problems involves non-deterministic resolver selection and deterministic flaw processing. To manage computational complexity, deterministic implementations can use algorithms like A<sup>\*</sup> or IDA<sup>\*</sup> for efficient solution generation. The main challenge lies in accurately defining the current state and measuring the distance to the desired state, which hampers the use of traditional planning heuristics. To address this, [29] propose a separation of temporal and causal elements, allowing adaptation of classical planning heuristics to the causal aspects. The proposed approach employs an AND/OR graph to represent causal relationships between flaws and resolvers, enabling efficient exploration of a disjunctive token network. By analyzing the topology of the generated graph, heuristics like  $h_{add}$  and  $h_{max}$  [31] can estimate resolver and flaw costs, guiding the resolution process. Specifically, we have that:

$$G(\varphi) = \min_{\rho \in res(\varphi)} G(\rho)$$

$$G_{add}(\rho) = c(\rho) + \sum_{\varphi \in precs(\rho)} G(\varphi)$$

$$G_{max}(\rho) = c(\rho) + \max_{\varphi \in precs(\rho)} G(\varphi)$$

where  $G(\varphi)$  represents the estimated cost for a flaw  $\varphi$ ,  $G_{add}(\varphi)$  represents the estimated cost for a resolver  $\rho$  computed through the  $h_{add}$  heuristic,  $G_{max}(\rho)$  represents the estimated cost for a resolver  $\rho$  computed through the  $h_{max}$  heuristic, and  $c(\rho)$  is the *intrinsic cost* of the  $\rho$  resolver, i.e., a positive number representing the *cost* of disjuncts, in case of priced disjunctions, or the value 1, in other cases.

During the resolution process, the solver employs a strategy that prioritizes addressing the most costly flaw with the least expensive resolver. This approach serves the dual purpose of early inconsistency detection and efficient solution attainment. However, it's crucial to acknowledge that constructing the causal graph has inherent limitations, and in many cases, it cannot be fully realized.

Specifically, the process commences by assigning a zero cost to the problem's facts, as these facts are considered inherently true. Conversely, both flaws and resolvers are initially assigned an infinite estimated cost. The process then unfolds in a breadth-first manner, moving backward through the graph. It continues in this manner until it encounters facts or other goals that can be associated with a finite estimated cost.

Upon such an encounter, the process proceeds to propagate the estimated costs, taking into account the strategy employed to estimate the goals (i.e., either using the  $h_{add}$  or  $h_{max}$  heuristic). This iterative process perseveres until the high-level flaws are ultimately attributed a finite estimated cost, marking significant progress in the resolution process.

When pursuing optimal plans, a significant challenge arises during the graph construction process, where the procedure may terminate before incorporating the optimal plan into the



**Figure 2:** An example of causal graph for the planning of a physical rehabilitation session. Tokens' parameters are omitted to avoid burdening the notation.

graph. An illustrative scenario is exemplified by the concept of "favorite goals", which represent objectives that one would ideally like to achieve but are not strictly mandatory. Failure to attain a favorite goal incurs a penalty for the planner. For instance, let's consider the problem illustrated in Figure 2. On the right-hand side, there is a requirement represented as the disjunction *PhysicalWorkout* ()<sub>g</sub>  $\vee$  {}<sub>50</sub>. This disjunction is designed to generate a plan that includes a physical workout, provided that the constraints permit it. The workout should encompass exercises targeting both the upper and lower parts of the body.

The complexity arises when, during the graph construction phase, the cost associated with an empty option (i.e., no plan for the goal) seems preferable compared to the initially infinite cost assigned to the preferred goal. This situation can prematurely conclude the graph construction procedure, suggesting to the planner that an empty plan is the best choice. In the example represented in Figure 2, in particular, the graph building procedure would not introduce the  $\rho_3$  resolver, it would keep an infinite estimated cost for the  $\varphi_1$  flaw since there already is a possible better (compared to the current infinite one) solution thoward the  $\rho_2$  resolver. In essence, concerning preferable goals, this strategy can result in plans that, while technically valid, exhibit very poor quality. Such plans may not align with the planner's actual preferences and objectives, emphasizing the importance of refining the resolution process for optimizing the quality of the generated plans.

The solution we propose involves the incorporation of "jamming operations" when necessary, aiming to prevent the premature termination of the graph construction procedure. Specifically, we introduce a type of goal denoted as *NoOp*, equipped with an ID parameter, ensuring that different "jamming goals" do not interfere with each other, and a positive integer parameter known as "look-ahead". The corresponding rule for these goals is structured as follows:  $NoOp(id, lookahead) \leftarrow NoOp(id : id, lookahead : lookahead - 1)_{\sigma} \lor [lookahead = 0].$ 

As an example, consider the prior disjunction, which could be modified to *PhysicalWorkout*()<sub>g</sub>  $\vee$  {*NoOp*(*id* : 1, *lookahead* : 42)<sub>g</sub>}<sub>50</sub>. This modification ensures that a depth of 42 is guaranteed for the empty (and more costly) branch. The estimated cost for this branch remains infinite until the lookahead parameter reaches zero. In the meantime, this





(a) A comparison of plan costs without and with (b) A comparison of execution times without and the utilization of jamming actions.



approach encourages exploration of the branch containing the physical activity, thus mitigating the issues related to prematurely favoring empty plans.

# 4. Experimental Setup

We have conducted some experiments to demonstrate the effectiveness of the proposed approach. Since we are working on a Active Assisted Living project, we focused on planning problems similar to those described in the previous section, in which the user has to carry out some physical and cognitive rehabilitation exercises to keep active and prolong his/her health wellbeing. In particular, series of physical exercises chosen from 14 different types (e.g., Chest press, Biceps curl, etc.) are planned in order to guarantee the training of all parts of the body. The exercises are repeated several times and with different characteristics depending on the profile of the user. Some constraints, however (e.g., lack of time on some days), might prevent the user from carrying out all the activities, so the planner must optimize by putting in as many workout sessions as possible.

In our experimental study, we performed a comparative analysis by introducing the proposed jamming actions in scenarios with an escalating number of physical training activities. We carefully assessed the time needed to find a solution for each case. Our expectation was that, despite the additional computational load imposed by the jamming actions, their incorporation would lead to the creation of notably superior plans. We remained hopeful that the introduction of these actions would not unduly compromise the efficiency of the resolution process.

As depicted in Figure 3b, the execution times remain comparable to those observed in scenarios without the presence of jamming actions. However, a noteworthy improvement is observed in the quality of solutions, as illustrated in Figure 3a. These improvements entail a shift from plans that accumulate penalties to plans in which costs align with the intrinsic costs of the activities, effectively reflecting optimal<sup>2</sup> plans.

Nevertheless, a crucial caveat must be considered. The heuristic guiding the resolution algorithm encourages the pursuit of optimal solutions. In the tested cases, optimal solutions are readily attainable. With the incorporation of jamming actions, the planner introduces all the necessary activities into the graph to discover the optimal plan. The heuristic then guides

<sup>&</sup>lt;sup>2</sup>We know the solutions are optimal because the generated plans contain all the preferred goals.

the resolution towards this solution. Attempts were made to introduce constraints preventing the planner from reaching the optimal solution, such as imposing a makespan shorter than the total number of activities. This was intended to compel the planner to add tasks following the heuristic but subsequently trigger backtracking. Regrettably, the performance in such cases proved unsatisfactory. We attribute this issue to the limitations of scheduling algorithms, which could benefit from optimization. For instance, there's no need to exhaustively explore all possible orderings of ten tasks of unit duration to determine that they cannot be sequenced in a way that achieves a duration of less than ten. In summary, while jamming actions significantly enhance solution quality, there is room for improving the scheduling algorithms to handle more complex scenarios efficiently.

# 5. Conclusion

This paper has delved into the realm of timeline-based planning, which offers a departure from traditional planning paradigms by providing adaptable and dynamic plans. It has explored, to the best of our knowledge, for the first time, the optimization of timeline-based planning, without requiring extensive modifications to the underlying planner. Instead, the paper introduces specific operators strategically into the planning domains, leveraging existing heuristics to guide decision-making effectively.

Timeline-based planning relies on the foundation of constraint networks and tokens, allowing for modeling and reasoning about complex problems. These tokens and constraints are manipulated to construct plans through a resolution process. In particular, this paper places a significant emphasis on the selection of disjuncts during resolution, which determines the cost and, thus, preferences among potential solutions.

The introduction of jamming actions, aimed at preventing premature termination of the graph construction process, has been proposed as a solution to the generation of optimized plans. These actions have been demonstrated through experiments focusing on physical and cognitive rehabilitation exercises, showcasing their ability to improve plan quality without significantly impacting execution times.

While the incorporation of jamming actions has shown promise in generating high-quality plans, there remains room for further improvement in scheduling algorithms to handle more complex scenarios efficiently. This paper marks a step towards enhancing the capabilities of timeline-based planners and paves the way for future research in this field.

## Acknowledgments

Authors' work is partially supported by FOCAAL - FOg Computing in Ambient Assisted Living. "ACCORDI DI INNOVAZIONE - Bando" Decreto MiSE 05/03/2018 e Decreto Direttoriale 27/09/2018, and CLEVERNESS - "Tecnologie a supporto delle fasce più fragili: giovani e anziani". FINANZIAMENTO PROGETTUALITA' STRAORDINARIA - FOE 2020.

# References

- [1] R. De Benedictis, M. Castiglioni, D. Ferraioli, V. Malvone, M. Maratea, E. Scala, L. Serafini, I. Serina, E. Tosello, A. Umbrico, M. Vallati, Preface to the Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (IPS-RCRA-SPIRIT 2023), in: Proceedings of the Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (IPS-RCRA-SPIRIT 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI\* IA 2023), 2023.
- [2] M. Bienvenu, C. Fritz, S. A. McIlraith, Specifying and computing preferred plans, Artificial Intelligence 175 (2011) 1308–1345. URL: https://www.sciencedirect.com/science/ article/pii/S0004370210002109. doi:https://doi.org/10.1016/j.artint.2010.11.021, representing, Processing, and Learning Preferences: Theoretical and Practical Challenges.
- [3] S. Sohrabi, J. A. Baier, S. A. McIlraith, Htn planning with preferences, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09), Pasadena, CA, USA, 2009, pp. 1790–1797. URL: 1242050782\_Sohrabi-IJCAI09.pdf, this paper also appears in Commonsense 2009.
- [4] A. Gerevini, D. Long, Preferences and soft constraints in pddl3, in: ICAPS workshop on planning with preferences and soft constraints, 2006, pp. 46–53.
- [5] S. Edelkamp, On the compilation of plan constraints and preferences., in: ICAPS, Citeseer, 2006, pp. 374–377.
- [6] E. Giunchiglia, M. Maratea, Planning as satisfiability with preferences, in: PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, volume 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 987.
- [7] N. Muscettola, HSTS: Integrating Planning and Scheduling, in: Zweben, M. and Fox, M.S. (Ed.), Intelligent Scheduling, Morgan Kauffmann, 1994, pp. 169–212.
- [8] D. S. Weld, An Introduction to Least Commitment Planning, AI Magazine 15 (1994) 27-61.
- [9] N. Muscettola, S. Smith, A. Cesta, D. D'Aloisi, Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Architecture, IEEE Control Systems 12 (1992).
- [10] M. Ghallab, H. Laruelle, Representation and Control in IxTeT, a Temporal Planner, in: AIPS-94. Proceedings of the 2nd Int. Conf. on AI Planning and Scheduling, 1994, pp. 61–67.
- [11] A. Jonsson, P. Morris, N. Muscettola, K. Rajan, B. Smith, Planning in Interplanetary Space: Theory and Practice, in: AIPS-00. Proceedings of the Fifth Int. Conf. on AI Planning and Scheduling, 2000, pp. 177–186.
- [12] S. Chien, D. Tran, G. Rabideau, S. Schaffer, D. Mandl, S. Frye, Timeline-Based Space Operations Scheduling with External Constraints, in: ICAPS-10. Proc. of the 20<sup>th</sup> Int. Conf. on Automated Planning and Scheduling, 2010, pp. 34–41.
- [13] A. Cesta, G. Cortellessa, S. Fratini, A. Oddi, Developing an End-to-End Planning Application from a Timeline Representation Framework, in: IAAI-09. Proceedings of the 21<sup>st</sup> Innovative Applications of Artificial Intelligence Conference, Pasadena, CA, USA, 2009, pp. 66–71.
- [14] S. Fratini, F. Pecora, A. Cesta, Unifying Planning and Scheduling as Timelines in a

Component-Based Perspective, Archives of Control Sciences 18 (2008) 231-271.

- [15] S. Fratini, A. Cesta, R. De Benedictis, A. Orlandini, R. Rasconi, APSI-based Deliberation in Goal Oriented Autonomous Controllers, ASTRA 11 (2011).
- [16] A. Umbrico, A. Cesta, M. Cialdea Mayer, A. Orlandini, Platinum: A new framework for planning and acting, in: AI\*IA 2017 Proceedings, 2017, pp. 498–512.
- [17] M. Cialdea Mayer, A. Orlandini, A. Umbrico, Planning and execution with flexible timelines: a formal account, Acta Informatica 53 (2016) 649–680. URL: http://dx.doi.org/10.1007/ s00236-015-0252-z. doi:10.1007/s00236-015-0252-z.
- [18] J. Frank, A. K. Jónsson, Constraint-Based Attribute and Interval Planning, Constraints 8 (2003) 339–364.
- [19] M. Fox, D. Long, PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains, Journal of Artificial Intelligence Research 20 (2003) 61–124.
- [20] P. Laborie, Algorithms for propagating resource constraints in AI planning and scheduling: existing approaches and new results, Artificial Intelligence 143 (2003) 151–188.
- [21] A. Cesta, A. Oddi, Gaining Efficiency and Flexibility in the Simple Temporal Problem, in: L. Chittaro, S. Goodwin, H. Hamilton, A. Montanari (Eds.), Proceedings of the Third International Workshop on Temporal Representation and Reasoning (TIME-96), IEEE Computer Society Press: Los Alamitos, CA, 1996, pp. 45–50.
- [22] S. Stock, M. Mansouri, F. Pecora, J. Hertzberg, Hierarchical hybrid planning in a mobile service robot, in: KI 2015 Proceedings, 2015, pp. 309–315.
- [23] F. Dvorák, A. Bit-Monnot, F. Ingrand, M. Ghallab, Plan-Space Hierarchical Planning with the Action Notation Modeling Language, in: IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Limassol, Cyprus, 2014. URL: https://hal.archives-ouvertes. fr/hal-01138105.
- [24] D. E. Smith, J. Frank, W. Cushing, The ANML language, in: ICAPS Workshop on Knowledge Engineering for Planning and Scheduling (KEPS), 2008.
- [25] G. Verfaillie, C. Pralet, M. Lemaître, How to model planning and scheduling problems using constraint networks on timelines, The Knowledge Engineering Review 25 (2010) 319–336.
- [26] A. Cesta, A. Oddi, S. F. Smith, A Constraint-Based Method for Project Scheduling with Time Windows, Journal of Heuristics 8 (2002) 109–136. URL: https://doi.org/10.1023/A: 1013617802515. doi:10.1023/A:1013617802515.
- [27] D. E. Smith, J. Frank, A. K. Jónsson, Bridging the Gap Between Planning and Scheduling, Knowledge Engineering Review (2000).
- [28] P. Laborie, M. Ghallab, Planning with Sharable Resource Constraints, in: Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., 1995, pp. 1643–1649.
- [29] R. De Benedictis, A. Cesta, Lifted Heuristics for Timeline-based Planning, in: ECAI-2020, 24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 2020, pp. 498–2337.
- [30] R. Dechter, Constraint Processing, Elsevier Morgan Kaufmann, 2003.
- [31] B. Bonet, H. Geffner, Planning as Heuristic Search, Artificial Intelligence 129 (2001) 5–33.