Toward Planning with Hierarchical Decompositions and Time-frames

Mica Gardone[1], **Rogelio E. Cardona-Rivera**[1 2]

Laboratory for Quantitative Experience Design
[1]Kahlert School of Computing, [2]Division of Games
University of Utah, Salt Lake City, UT, USA
{m.gardone | r.cardona.rivera}@utah.edu

Abstract

The semantics of temporal hierarchical planners are limited. In hierarchical paradigms, temporal reasoning has largely focused on durative constraints of primitive actions, which may be added directly or appear post-expansion. We propose extending temporal reasoning to composite actions, specifically within decompositional partial order causal linked planning. We outline how a general-purpose hierarchical planner can approach temporal reasoning outlined in a STRIPS-like formalism. We build upon existing temporal and hierarchical semantics, and sketch two novel approaches: time-frame planning and decompositional time-frame planning.

1 Introduction

Hierarchical planning has enjoyed uses in robotics, space, and business applications. However, space agencies (European Space Agency 2019; United States National Aeronautic Space Administration 2023) have posted open questions and concerns with time for a variety of reasons. One such open question is dealing with time as both a window and an end time. *Temporal hierarchical* planning has received comparatively less time in research than its non-temporal cousins, and as such there are still many elements under- and undefined.

Temporal planning has focused a variety of topics like planner-schedulers (Parimi, Rubinstein, and Smith 2022), portfolio planning (Furelos-Blanco and Jonsson 2018), and many other forms of planning (Younes and Simmons 2003; Turi and Bit-Monnot 2022; Do and Kambhampati 2014). There still is not an agreed-upon, general purpose formalism, however, there are common attributes among all systems. There has been work to discover better heuristics through non-temporal means (Cavrel, Pellier, and Fiorino 2023). Representing time in planners – e.g., *timelines* (Frank 2013), *temporal constraint networks* (Dechter, Meiri, and Pearl 1991), and *chronicles* (Rahmani, Shell, and O'Kane 2021) – has been a major research area, for the sake of improving both knowledge representation and search. Some planners (Dvorak et al. 2014; Bit-Monnot et al. 2020) approach temporal hierarchical reasoning, yet continue to assume that composite actions are non-temporal.

Within the hierarchical community, there have been open challenges (Kiam, Bercher, and Schulte 2021) and proposing semantics (Smith and Cushing 2008; Pellier et al. 2022). The challenges produce domain-specific solutions that we could draw upon. Temporally-aware hierarchical reasoning proposals state only primitive actions are capable of having duration semantics for simplicity. However, this cannot be the case as an "instantaneous" composite action composed of durative primitive actions is logically inconsistent. In domains or specifications where expansion is particularly expensive, delayed, or non-desirable (Gréa, Matignon, and Aknine 2018), instantaneous composite actions are not representative enough and can lead to undesirable outcomes.

In this paper, we approach a potential solution to allowing composite actions to have temporal information by:

1. Proposing a novel time-frame based paradigm: time-frame planning, and
2. Propose how to combine time-frame planning and decompositional planning.

We believe the extensions provided here will help foster further discussion around a general purpose, temporal hierarchical planning formalism.

2 Related Work

There is an on-going discussion of the semantics of temporal hierarchical planning. We initially draw upon work done recently by (Pellier et al. 2022). We relax the need of duration being only on primitive actions/tasks. As well, we introduce potential search space constraints that can impact what expansions are selected. Action Notation Modeling Language (Smith and Cushing 2008) (ANML) and PDDL 2.1/2.2 (Fox and Long 2003; Edelkamp and Hoffman 2004) also discuss some basic forms of hierarchy, yet leave temporal hierarchical semantics undefined.

There have been many domain-specific planners that have built their own solution. While each solution is unique and ground breaking for its area, none build are a formal, universal framework. Some early temporally-aware software systems utilize a strongly built library of actions based on empirical data, one being the Heuristic Scheduling Testbed System (HSTS) (Muscettola et al. 1992). These actions, and environments, are believed to be common occurrences in the domain they are applied; making them tightly coupled to their originating domain area. Other planners that utilize a planner-scheduler hybrid planner (Cesta et al. 2007) are also difficult to generalize due to a specialized language and/or formalism specific to the problem. FAPE is one influential planning system that defines its own hierarchical and temporal planning for acting through a combined planner-executor (Bit-Monnot et al. 2020). One key difference between what we propose and what FAPE implements is that the latter is specific to ANML and chronicle planning. FAPE's temporal extents are only known on fully expanded hierarchical tasks. In this paper, we propose planners have the ability to reason over temporal extents on *un-expanded* hierarchical actions.

3 Background

In this section, we describe the elements necessary from simple temporal planning to understand time-frame planning. We will also establish a baseline for hierarchical planning to discuss how to combine the two paradigms.

3.1 Simple Temporal Domains & Problems

To begin our discussion of time-frame planning, we refer to the simple temporal problem. The notation we use is both STRIPs-like and derived from COLIN (Coles et al. 2012). A simple temporal problem with discrete effects from PDDL 2.1 can be represented as $\langle \mathcal{I}, \mathcal{A}, \mathcal{G} \rangle$ where:

1. $\mathcal{I}$ is the initial state which contains a set of propositions and an assignment of values to a set of numeric variables.
2. $\mathcal{A}$ is the set of actions, where each action (a) is defined as $\langle \text{pre}_{\vdash}, \text{pre}_{\leftrightarrow}, \text{eff}_{\vdash}, \text{pre}_{\dashv}, \text{eff}_{\dashv}, \text{dur} \rangle$, such that:
   (a) pre_x denotes the conditions that must be maintained both *at start* ($\text{pre}_{\vdash}$) and *at end* ($\text{pre}_{\dashv}$) of a.
   (b) eff_x denotes the effects that are applied after the conditions of a are met in the start ($\text{eff}_{\vdash}$) and end ($\text{eff}_{\dashv}$). Both effect collections are further defined as:
      i. eff_x^-, propositions to be removed from the world,
      ii. eff_x^+, propositions to be added to the world,
      iii. eff_x^n, modifications on numeric variables.
   (c) $\text{pre}_{\leftrightarrow}$ denotes the invariants (*over all*); these are conditions that must be maintained between the start and end of a.
   (d) dur denotes the duration constraints which defines the duration between a's start and end. These constraints are further refined with respect to ordering constraints. This allows for a special parameter, `?duration`.
3. $\mathcal{G}$ is the goal of the problem: a set of propositions and values that must be achieved.

Further, the definition of a duration in an action can take either one or two constraints. A constraint takes the form: $\langle \texttt{?duration}, \text{op}, c \rangle$ where `?duration` is the special purpose parameter, $\text{op} \in \{>, >=, <, <=, =\}$, and $c \in \mathbb{R}$.

Two constraints define two unique bounds on the operator's minimum and maximum. The equality operator cannot be used in the definition of two constraints. The two constraint tuple takes the form: $\langle\langle \texttt{?duration}, \text{op}_1, c1 \rangle, \langle \texttt{?duration}, \text{op}_2, c_2 \rangle\rangle$ where $\text{op}_1 \in \{>, >=\}$, $\text{op}_2 \in \{<, <=\}$, $c_1, c_2 \in \mathbb{R}$ and c_1 does not have to equal c_2. The two constraints can approximate the behavior of the equality operator.

Flaws & Refinements. Simple temporal planning in partial-order causally linked (POCL) planning entails two basic types of flaws: open conditions and causal threats. Open conditions are unsatisfied preconditions. In temporal planning, open conditions can be in either the *at start* condition block, *at end* condition block, or *invariant* block. An open condition is resolved one at a time, en-queuing all potential fixes either from instantiating new actions or reusing steps in the plan. Causal threats arise when a causal link would be undone by an inverse effect (e.g., p and $\neg p$). Causal threats are solved be one of three methods: promotion (ordering the threatening step after the causal link's consumer), demotion (ordering the threatening step before the causal link's producer), or non-codesignation (in the event of lifted actions).

Refinements to the plan are made per *refinement strategies*, which are processes of which flaws are selected in some order. All solutions generated by the flaw are then queued back onto the search fringe. Refinement strategies can come in different forms and deal with a variety of flaws (Pollack, Joslin, and Paolucci 1997). Planners can also change strategies at run-time (Younes and Simmons 2003).

Solutions. A solution to the given problem is a sequence of actions from $\mathcal{A}$ that establishes all goal conditions in the problem. A solution must respect the duration constraints of every action in the solution; that is, no action should last longer than its maximum defined duration or be scheduled such that it takes less time than minimally allowed. The solution is the tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ where:

1. $\mathcal{S}$ is the set of actions instantiated into the plan, referred to as *steps*. All $s \in \mathcal{S}$ correspond to an $a \in \mathcal{A}$ from the problem definition.
2. $\mathcal{O}$ is the orderings over the steps in the solution. The ordering system is temporally-aware. Every $o \in \mathcal{O}$ takes the form $p_{s/e} \prec c_{s/i/e}$, where $p, c \in S$. s, i, e correspond to start, invariant, and end blocks.
3. $\mathcal{L}$ are links between an effect and a precondition. A causal link $l \in \mathcal{L}$ is the tuple $\langle p_{s/e} \prec c_{s/i/e}, q \rangle$ where $p, c \in S$ and q is a predicate effect in the producer (p).

3.2 Hierarchical Reasoning

There are several different variations of hierarchical planning (Bercher, Alford, and Höller 2019). We specifically utilize the decompositional POCL (DPOCL) (Young and Moore 1994; Winer and Cardona-Rivera 2018) formalism as our approach to hierarchical reasoning.

A standard decompositional problem takes the same form as in Section 3.1. The key difference lies in the set of actions, $\mathcal{A}$, where each a is defined as $\langle \text{pre}, \text{eff}, \text{composite}, \Lambda \rangle$. Each element is defined as:

1. pre is the action's preconditions, which must be maintained at the start.
2. eff is the action's effects, which affect the world. Effects take the same form as in Section 3.1.
3. composite is a true/false flag to indicate it is a composite header step that needs to be decomposed or expanded. If the flag is true, then the step is a composite step.

4. Λ is the set of schemas that can be used to expand the composite action. A schema $\lambda \in \Lambda$ takes the form $\langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ where:
   (a) $\mathcal{S}$ is the set of pseudo-actions in the decomposition that must be added to the plan. All $s \in \mathcal{S}$ can either be a composite or a primitive, allowing for the nesting of composite pseudo-actions.
   (b) $\mathcal{O}$ is the set of orderings over the steps in $\mathcal{S}$.
   (c) $\mathcal{L}$ is the set of causal links in the decomposition that links effects to preconditions.

Flaws & Refinements. On top of the open condition and causal threat flaws in POCL, a decomposition flaw is introduced. This flaw signals to the planner that the given step is composite and thus must be expanded. All causal links that link to and from the composite step must be updated to the newly created dummy start and end.

DPOCL, as it was introduced, requires decomposition flaws to be resolved first before any other flaw. We do not make that strong of commitment to decomposition first, as there are situations in planning where this is not desired (Gréa, Matignon, and Aknine 2018).

Expanding Schemas. When expanding schemas, it is important to modify all existing orderings such that everything that comes after, before, and during the composite step is maintained. For DPOCL, a decompositional link is generated on expansion to keep associated actions together.

Solutions. A solution in a standard hierarchical problem is: $\langle \mathcal{S}, \mathcal{O}, \mathcal{L}, \mathcal{D} \rangle$. $\mathcal{S}, \mathcal{O}$, and $\mathcal{L}$ are similar to the simple temporal solution without time. $\mathcal{D}$ is the set of decomposition links.

4 Towards Temporal Decomposition

We extend on the prior notation of a simple temporal solution to a novel planning paradigm: time-frame planning. While we outline a sketch here, space precludes us from diving in to the deeper technical representations. A parameter is added to the problem representation to support new reasoning, creating the tuple of $\langle \mathcal{I}, \mathcal{A}, \mathcal{G}, \mathcal{T} \rangle$ where:

1. $\mathcal{I}, \mathcal{A}$, and $\mathcal{G}$ are the same as before.
2. $\mathcal{T}$ is a constraint on the duration of the solution much in the same way as dur is for actions. That is, $\mathcal{T}$ defines a minimum and/or maximum duration that bounds the solution. $\mathcal{T}$ utilizes the special parameter provided in temporal-metric planning, `total-time`, to define its own constraints. However, this duration constraint is not modified based on what is in the plan: it defines what a solution to the problem must satisfy. The parameter `total-time` also takes the form of a tuple, $\langle min, max \rangle$, which indicates the absolute minimum and maximum time.

A time-frame is composed in the same way the duration of an action is: there exists either a single constraint, or two constraints. The single constraint can define either one bound with the set $\{>, <, >=, <=\}$. The two constraint tuple can only use $\{>, >=\}$, which defines a minimum, and $\{<, <=\}$, which defines a maximum. The prior definitions of duration constraint tuples for both single- and two constraints applies to a time-frame as well.

New Flaws. As we have defined a new constraint to satisfy, there must also be some way for a planner to know when these issues arise. Overtime flaws are generated when at least one chain of actions in the plan could run longer than the maximum duration the problem defines. An overtime flaw can be defined as the tuple: $\langle \texttt{total-time}_{max}, >, c \rangle$ where $\texttt{total-time}_{max}$ is the max duration of the plan and $c \in \mathbb{R}$ is a constant defined as the solution's maximum. In this case `total-time` exceeds c, indicating we could potentially go over time. This type of failure can be found in space: attempting to facilitate a spacewalk longer than the available oxygen in the astronaut's system.

Conversely, undertime flaws are generated when no chain of actions reach the minimum threshold defined by the problem. This flaw can be expressed in the tuple: $\langle \texttt{total-time}_{min}, <, c \rangle$ where `total-time` is the special parameter and $c \in \mathbb{R}$ is a constant defined as the solution's minimum. This indicates that our longest minimum time is still below our threshold, and must be increased. This flaw type can be observed when attempting a slingshot maneuver: burn your engines for too little time, the rocket might end up being pulled in closer to the planet which results in negative consequences.

Solutions. A solution in time-frame planning adheres to the same principles as a solution in simple temporal problems: a series of actions that respect each other's duration constraints. A time-frame solution differs, however, as the duration constraint $\mathcal{T}$ must also be respected. Moreover, the solution has the potential to offer a family of solutions much in the same way a solution that is not total-ordered offers multiple solutions, conditioned on not using the equality ('=') operator in actions. The optimal solution returned from a simple temporal planner might not be complete in a time-frame setting.

4.1 Decompositional Time-frame Planning

Combining time-frame and hierarchical reasoning has a fair number of questions and concerns such as: 1. how should duration constraints be treated, 2. should decompositional actions always be bound, 3. should schemas be able to affect the top-level bounds, and many other considerations, not including to specific planning styles (e.g., HTNs vs DPOCL). Actions are expanded to be the tuple $\langle \text{pre}_{\vdash}, \text{pre}_{\leftrightarrow}, \text{eff}_{\vdash}, \text{pre}_{\dashv}, \text{eff}_{\dashv}, \text{dur}, \text{composite}, \text{bound}, \text{rel}, \Lambda \rangle$ where:

1. The following remain the same as in Section 3.1 and Section 3.2: pre_x, eff_x, dur, composite, and Λ.
2. *bound* states how to treat the action's temporal duration during parsing. There are two types of bounds: *strict* and *flexible*. *strict*-bounds are top-to-bottom: all schemas associated with the action must adhere to the constraints. *flexible*-bounds are a bottom-up approach to temporal bounds: the top-level action's duration is defined by the schemas. *flexible*-bounded actions may find that they are

infinite in maximum duration due to not all open conditions and causal threats being resolved in at least one schema. When planning, all bounds are treated as strict by the planner.

3. rel is how long schemas can take relative to their starting time: given the current minimum duration of the schema, how far is the maximum. If a schema has no open conditions or causal threats, rel is ignored. Otherwise, the minimum time is determined from orderings.

With the expansion of a composite action, we must also discuss changes to schemas. As expanding a composite action leads to the inheritance of all flaws that come with the given schema, we must contend with inherited temporal flaws. A schema's tuple is thus expanded as: $\langle \mathcal{S}, \mathcal{O}, \mathcal{L}, \text{pre}_{\leftrightarrow}, \text{dur}, \text{rel} \rangle$. $\mathcal{S}, \mathcal{O}$, and $\mathcal{L}$ are the representation of sets as in Section 3.2. However, the other components are defined as:

1. $\text{pre}_{\leftrightarrow}$ are *schema-level* invariants. These are added to the plan as open conditions to the dummy start, as schemas might have contradicting invariants.
2. rel is the same as in the composite operator definition. This rel overrides the composite operator's rel, if defined.
3. dur, unlike in the action scheme, cannot be directly defined by the domain engineer. The maximal and minimal extents, the components of dur, are defined by the schema's steps and orderings. Should there exist an open condition or causal threat, then the maximum duration of the schema is defaulted to infinite unless rel is defined either in the schema itself, or the top-level action.

Expanding Temporal Schemas. When expanding a temporal schema, the *at start*, *at end*, and *over all* blocks of the top-level action must be maintained. *at start* and *at end* can be decomposed into two pairs of actions with a duration of 0 with their respective blocks. For example, say a composite step has an at start condition a and at start effect b. The new stand-in for the composite start has a condition of a, and its at end effect has an effect of a and b. *at end* is decomposed in the same way, just using its condition and effect blocks.

over all, when expanded, asks: what does it mean to have an invariant over multiple actions? We represent such cases as causal links between the *at start-at end* effect and *at end-at start* condition. Schema-level invariants are appended to the end of at end effects. Of course, invariants also need to be satisfied as open conditions which we can solve by placing them as *at start* at end conditions.

Decomposition links should be generated the same; however, with the added notation of time-frames we must also record how long this decomposition can be. Modifying the duration of a top-level composite action before expansion should affect the search space. By further constraining the action, schemas are culled from potential expansions. Decomposition links should reflect these constrained bounds.

Bound Interactions. Another question we must consider is, what happens if a composite step is *strict*-bounded and a schema runs over or under time?

For example, a top level action that in some domain has a defined constraint of $\langle 5, 10 \rangle$ with a *strict*-bound. If there is a schema that has a calculated duration constraint of $\langle 6, 8 \rangle$, there are no issues. If a schema has a duration constraint of $\langle 6, 12 \rangle$ or $\langle 2, 8 \rangle$, the parser should not error out as there is a potential solution (reducing or increasing the actions). If there is a schema that is $\langle 2, 5 \rangle$ or $\langle 11, 17 \rangle$, then the parser should produce an error and prevent the planner from running as there are no actual solutions. If a schema has no maximum due to there being a flaw in it, then the maximum is either $\text{schema}_{\text{min}} + \text{rel}$, if rel is defined, or the maximum allowed by the action. Given the same duration constraints, if the action was *flexible*-bound then no error would be produced in any of the cases. The action's duration would not be $\langle 5, 10 \rangle$ but $\langle 2, 17 \rangle$.

Modified Flaws. How does temporal hierarchical planning interact with time-frame planning? The overtime and undertime flaws need to be modified slightly as both flaws exist as plan-wide issues. We fix this by adding a reference to a decomposition link (d) to the definitions. Thus, dur can be either the `total-time` or the duration of a specific decomposition, depending on if $d = \emptyset$ or $d \in \mathcal{D}$, respectively. The latter statement can be derived by examining the distance between the schema's start's end and the schema's end's start. A planner should resolve these flaws much in the same way as a plan-wide overtime/undertime flaw with the key difference being to begin at the expanded schema's end. Actions that are linked only to the schema's start are not considered for the solution, as they only shift the sub-plan and don't contribute to going over the duration.

Fundamentally, the decomposition flaw remains the same as described in Section 3.2.

Solutions. A given solution to a temporally-aware, hierarchical reasoning planner is: $\langle \mathcal{S}, \mathcal{O}, \mathcal{L}, \mathcal{D} \rangle$. $\mathcal{S}, \mathcal{O}$, and $\mathcal{L}$ are the same as in simple temporal and time-frame planning. $\mathcal{D}$ is the set of temporally-aware decomposition links. All solution requirements are the same as in time-frame planning and classical hierarchical planning.

5 Review & Future Work

In this paper, we defined two novel paradigms: time-frame planning and decompositional time-frame planning. The former operates over primitive operators and allows problems to specify further solution constraints in both time windows and time periods. We described two new flaws, overtime and undertime, how they're identified, and how a planner could resolve them. We extended these notions to decompositional planning, giving composite actions duration constraints. We defined new and necessary elements for decompositional time-frame planning at both the action level, and its schemas. We also added more information to the time-frame flaws to constrain decomposed actions' sub-plans to a specified, desired time. By doing so, we stated expansion itself can be impacted by the composite action's duration changing and that some schemas may not be added as potential solutions. Future implementations can utilize the same domains and problems seen in prior work (Yorke-Smith 2005) for testing.

Acknowledgements

This material is based upon work supported by the U.S. National Science Foundation (Grant #2046294). We also thank our anonymous reviewers for their feedback during peer review.

References

Bercher, P.; Alford, R.; and Höller, D. 2019. A Survey on Hierarchical Planning – One Abstract Idea, Many Concrete Realizations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 6267–6275.

Bit-Monnot, A.; Ghallab, M.; Ingrand, F.; and Smith, D. E. 2020. FAPE: a Constraint-based Planner for Generative and Hierarchical Temporal Planning. ArXiv:2010.13121 [cs].

Cavrel, N.; Pellier, D.; and Fiorino, H. 2023. On Guiding Search in HTN Temporal Planning with non Temporal Heuristics. *ArXiv*, abs/2306.07638.

Cesta, A.; Cortellessa, G.; Fratini, S.; Oddi, A.; and Policella, N. 2007. The MEXAR2 Support to Space Mission Planners. In *Proceedings of the 17th European Conference on Artificial Intelligence*.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. COLIN: Planning with Continuous Linear Numeric Change. *Journal of Artificial Intelligence Research*, 44: 1–96.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence*, 49(1-3): 61–95.

Do, M.; and Kambhampati, S. 2014. Sapa: A domain-independent heuristic metric temporal planner. In *Proceedings of the 6th European Conference on Planning*, 57–68.

Dvorak, F.; Bartak, R.; Bit-Monnot, A.; Ingrand, F.; and Ghallab, M. 2014. Planning and Acting with Temporal and Hierarchical Decomposition Models. In *Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, 115–121.

Edelkamp, S.; and Hoffman, J. 2004. PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition. Technical Report 195, Albert-Ludwigs-Universität Freiburg, Institut für Informatik.

European Space Agency. 2019. APSI - Advanced Planning and Scheduling Initiative. https://essr.esa.int/project/apsi-advanced-planning-and-scheduling-initiative (Last accessed: May 24, 2024).

Fox, M.; and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*, 20: 61–124.

Frank, J. 2013. What is a Timeline? In *Proceedings of the 4th Workshop on Knowledge Engineering for Planning and Scheduling at the 23rd International Conference on Automated Planning and Scheduling*, 31–38.

Furelos-Blanco, D.; and Jonsson, A. 2018. CP4TP: A Classical Planning for Temporal Planning Portfolio. In *Temporal Track of the International Planning Competition (IPC) 2018*.

Gréa, A.; Matignon, L.; and Aknine, S. 2018. HEART: HiErarchical Abstraction for Real-Time Partial Order Causal Link Planning. In *Proceedings of the 1st ICAPS Workshop on Hierarchical Planning at the 28th International Conference on Automated Planning and Scheduling*, 17–25.

Kiam, J. J.; Bercher, P.; and Schulte, A. 2021. Temporal Hierarchical Task Network Planning with Nested Multi-Vehicle Routing Problems – A Challenge to be Resolved. In *Proceedings of the 4th ICAPS Workshop on Hierarchical Planning at the 31st International Conference on Automated Planning and Scheduling*, 71–75.

Muscettola, N.; Smith, S.; Cesta, A.; and D'Aloisi, D. 1992. Coordinating space telescope operations in an integrated planning and scheduling architecture. *IEEE Control Systems*, 12(1): 28–37.

Parimi, V.; Rubinstein, Z. B.; and Smith, S. F. 2022. T-HTN: Timeline Based HTN Planning for Multiple Robots. In *Proceedings of the 5th ICAPS Workshop on Hierarchical Planning 32nd International Conference on Automated Planning and Scheduling*, 59–67.

Pellier, D.; Fiorino, H.; Grand, M.; Albore, A.; and Bailon-Ruiz, R. 2022. HDDL 2.1: Towards Defining an HTN Formalism with Time. ArXiv:2206.01822 [cs].

Pollack, M. E.; Joslin, D.; and Paolucci, M. 1997. Flaw Selection Strategies for Partial-Order Planning. *Journal of Artificial Intelligence Research*, 6: 223–262.

Rahmani, H.; Shell, D. A.; and O'Kane, J. M. 2021. Planning to Chronicle. In *Proceedings of the Algorithmic Foundations of Robotics XIV Conference*, 277–293.

Smith, D. E.; and Cushing, W. 2008. The ANML Language. In *Proceedings of the Workshop on Knowledge Engineering for Planning and Scheduling at the 18th International Conference on Automated Planning and Scheduling*.

Turi, J.; and Bit-Monnot, A. 2022. Guidance of a Refinement-based Acting Engine with a Hierarchical Temporal Planner. In *Proceedings of the Workshop on Integrated Action and Execution at the 32nd International Conference on Automated Planning and Scheduling*.

United States National Aeronautic Space Administration. 2023. Planning & Scheduling Group. https://www.nasa.gov/intelligent-systems-division/autonomous-systems-and-robotics/planning-and-scheduling-group/ (Last accessed: May 24, 2024).

Winer, D. R.; and Cardona-Rivera, R. E. 2018. A Depth-Balanced Approach to Decompositional Planning for Problems where Hierarchical Depth is Requested. In *Proceedings of the 1st Hierarchical Planning Workshop at the 28th Conference on Automated Planning and Scheduling*, 1–8.

Yorke-Smith, N. 2005. Exploiting the Structure of Hierarchical Plans in Temporal Constraint Propagation. In *20th AAAI National Conference on Artificial Intelligence*.

Younes, H. L.; and Simmons, R. G. 2003. VHPOP: Versatile Heuristic Partial Order Planner. *Journal of Artificial Intelligence Research*, 20: 405–430.

Young, R. M.; and Moore, J. D. 1994. DPOCL: A Principled Approach to Discourse Planning. In *Proceedings of the 7th International Workshop on Natual Language Generation*, 13–20.