On the Notion of Fixability of PDDL+ Plans [Extended Abstract]*

Francesco Percassi[1], **Enrico Scala**[2], **Mauro Vallati**[1]
[1] School of Computing and Engineering, University of Huddersfield, UK
[2] Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Brescia, Italy
f.percassi@hud.ac.uk, enrico.scala@unibs.it, m.vallati@hud.ac.uk

The PDDL+ Plan Fixing Framework

Automated planning is a field of artificial intelligence that aims to develop methods for synthesising decisions capable of transforming a given state, i.e., the initial state, into a desired state, i.e., the goal state, according to a given model of the world. Hybrid systems, which involve both discrete and continuous dynamics, are often encountered in real-world applications. The planning community designed PDDL+ to model this class of systems (Fox and Long 2006), which combines an action-oriented representation of an agent with an explicit representation of the environment and its exogenous dynamics. A PDDL+ problem involves finding a sequence of time-stamped actions along a continuous timeline that conforms to the changes prescribed by events and processes while meeting the preconditions for executing actions and achieving the desired goal state.

In real-world planning applications, plans can fail as circumstances may change over time, and it is crucial to efficiently fix issues in case of failures. In PDDL+, due to the complexity of the plan generation process, it is essential to reuse as much as possible of an existing plan, rather than generate a new one from scratch. To support the use of PDDL+ in planning applications, Percassi, Scala, and Vallati (2023b) provide an overarching definition of the plan fixing problem (FIXABILITY), which allows for a range of variations, including rescheduling and validation, for a given plan.

A PDDL+ plan is defined as $\pi_t = \langle \pi, t_e \rangle$, where $\pi = \langle \langle a_1, t_1 \rangle, .., \langle a_n, t_n \rangle \rangle$ is a sequence of time-stamped actions, while t_e is the duration of the plan (makespan).

The template for FIXABILITY is the tuple $\langle \Pi, \pi_t, \mathcal{C} \rangle$, where Π is a PDDL+ problem, π_t is the plan to be fixed (potentially invalid for Π), and $\mathcal{C}$ is a set of constraints. The aim is to find a fixed plan π'_t that is valid for Π and complies with $\mathcal{C}$. Intuitively, $\mathcal{C}$ defines the spectrum of manipulations for transforming π_t into π'_t. Depending on the constraints imposed through $\mathcal{C}$, we can obtain different FIXABILITY specialisations.

The simplest case occurs when $\mathcal{C} = \emptyset$, resulting in FIXABILITY$_\emptyset$; solving this problem is equivalent to generating a plan, as every valid plan for Π is also a valid solution for FIXABILITY$_\emptyset$. If we require that (i) every action instance of π_t must appear in π'_t, and (ii) no other actions can be added, we obtain FIXABILITY$_\mathcal{I}$, in which $\mathcal{C} = \mathcal{C}_\mathcal{I} = \mathcal{C}_\in \cup \mathcal{C}_\notin$, where $\mathcal{C}_\in$ and $\mathcal{C}_\notin$ enforce (i) and (ii), respectively. If, in addition to $\mathcal{C}_\mathcal{I}$, we want π'_t to preserve the original ordering of π_t, the resulting problem is FIXABILITY$_\mathcal{S}$, which has $\mathcal{C} = \mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{S}$, where $\mathcal{C}_\mathcal{S}$ is the set of constraints responsible for enforcing the ordering. Alternatively, it is possible to require that each action instance $\langle a_i, t_i \rangle$ of π_t be executed within a time window centred on t_i and wider as $\omega \in \mathbb{Q}_{\geq 0}$, i.e., $[t_i - \frac{\omega}{2}, t_i + \frac{\omega}{2}]$; this gives rise to FIXABILITY$_\mathcal{W}$, where $\mathcal{C} = \mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{W}(\omega)$, with $\mathcal{C}_\mathcal{W}(\omega)$ imposing the time window for each action instance of π_t. Finally, the constraints $\mathcal{C}_\mathcal{I}$, $\mathcal{C}_\mathcal{S}$, and $\mathcal{C}_\mathcal{W}$ can be combined, inducing FIXABILITY$_{\mathcal{WS}}$.

The problems described so far try to fix a plan compatibly with $\mathcal{C}$ by considering an unbounded temporal horizon. We can define a temporally bounded variant of them by requiring that the fixed plan is found within a finite horizon determined by a parameter $\sigma \in \mathbb{Q}_{\geq 0}$. So, given $Z \in \{\mathcal{I}, \mathcal{S}, \mathcal{W}, \mathcal{WS}\}$, we define FIXABILITY$^{\leq}_{Z}$ where $\mathcal{C} = \mathcal{C}_Z \cup \{\langle t'_e \leq t_e + \sigma \rangle\}$.

All of these problems, except for FIXABILITY$_\emptyset$, require $\mathcal{C}_\mathcal{I} \subseteq \mathcal{C}$, which implies that the actions of π'_t must be exclusively those of π_t. In this sense, FIXABILITY$_Z$ and FIXABILITY$^{\leq}_{Z}$ with $Z \in \{\mathcal{I}, \mathcal{S}, \mathcal{W}, \mathcal{WS}\}$ are *plan rescheduling* problems, in which the actions can be moved along the temporal dimension but cannot be deleted or added.

The PDDL+ validation problem (VALIDATION) evaluates whether a plan π_t is valid with respect to Π. VALIDATION can be expressed within the FIXABILITY framework as the tuple $\langle \Pi, \pi_t, \mathcal{C} \rangle$, where π_t is the plan to be validated and $\mathcal{C} = \mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{S} \cup \mathcal{C}_\mathcal{W}(\omega = 0) \cup \{\langle t'_e = t_e \rangle\}$. In qualitative terms, $\mathcal{C}$ requires that π'_t must exclusively contain the action instances of π_t ($\mathcal{C}_\mathcal{I}$), executed in the same order ($\mathcal{C}_\mathcal{S}$) and at the same timestamps. This last constraint is enforced by requiring that the actions be executed in a zero-width time window ($\mathcal{C}_\mathcal{W}(\omega = 0)$). Finally, it is required that π'_t and π_t have the same duration ($\{\langle t'_e = t_e \rangle\}$).

Percassi, Scala, and Vallati (2022) demonstrated how to convert plan validation into a PDDL+ problem, allowing any PDDL+ planner to be used for validating plans, regardless of the selected semantics, continuous or discrete (Percassi, Scala, and Vallati 2023a). This methodology relies on a re-

*The extended abstract reports on the work previously appeared in (Percassi, Scala, and Vallati 2022, 2023b).

FIX $\langle \Pi, \pi_t, \mathcal{C} \rangle$	Constraints $\mathcal{C}$	Complexity	FIX as PDDL+ Planning
FIX$_\emptyset$	$\emptyset$	Undecidable	Π
FIX$_\mathcal{I}$	$\mathcal{C}_\mathcal{I}$	NP-HARD	$\Pi^{\pi_t}_{\mathcal{R}_\mathcal{I}}$
FIX$_\mathcal{S}$	$\mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{S}$		$\Pi^{\pi_t}_{\mathcal{R}_\mathcal{S}}$
FIX$_\mathcal{W}$	$\mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{W}(\omega)$		$\Pi^{\pi_t}_{\mathcal{R}_\mathcal{W}}$
FIX$_{\mathcal{WS}}$	$\mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{W}(\omega) \cup \mathcal{C}_\mathcal{S}$		$\Pi^{\pi_t}_{\mathcal{R}_{\mathcal{WS}}}$
FIX$^{\leq}_{Z}$	$\mathcal{C}_Z \cup \{\langle t'_e \leq t_e + \sigma \rangle\}$	NP-COMPLETE	$\Pi^{\pi_t}_{\mathcal{R}^{\leq}_{Z}}$
VAL	$\mathcal{C}_\mathcal{I} \cup \mathcal{C}_\mathcal{W}(0) \cup$ $\mathcal{C}_\mathcal{S} \cup \{\langle t'_e = t_e \rangle\}$	P	$\Pi^{\pi_t}_{\mathcal{V}_0}$

Table 1: Theoretical results about FIXABILITY (shortened as FIX) and VALIDATION (shortened as VAL) problems.

formulation, namely $\mathcal{V}_0$, which generates a problem $\Pi^{\pi_t}_{\mathcal{V}_0}$ from a given VALIDATION task $\langle \Pi, \pi_t \rangle$. This problem admits a solution iff π_t is a valid plan for Π. The key concept is to create a novel PDDL+ problem where only the actions of π_t can be executed. These actions must be carried out based on the plan's specifications, which means they must be performed at their time-stamps and in the same order.

Since VALIDATION can be seen as a special case of the general FIXABILITY problem, we can approach plan rescheduling problems by relaxing action preconditions and goals (if needed) from $\Pi^{\pi_t}_{\mathcal{V}_0}$. Therefore, for each plan rescheduling problem FIXABILITY$_Z$, with $Z \in \{\mathcal{I}, \mathcal{S}, \mathcal{W}, \mathcal{WS}\}$, it is possible to define a translation $\Pi^{\pi_t}_{\mathcal{R}_Z}$ which admits a solution π'_t iff π_t is fixable under the constraints $\mathcal{C}_Z$.

The results presented by Percassi, Scala, and Vallati (2023b) demonstrate that VALIDATION belongs to P, while the most constrained plan rescheduling problem, namely FIXABILITY$^{\leq}_{\mathcal{WS}}$, is NP-HARD. Additionally, it has been proven that FIXABILITY$^{\leq}_{\mathcal{I}}$ is in NP. By combining the fact that FIXABILITY$_\mathcal{I}$ is NP-HARD and FIXABILITY$^{\leq}_{\mathcal{WS}}$ is in NP, it can be concluded that all problems between FIXABILITY$^{\leq}_{\mathcal{WS}}$ and FIXABILITY$^{\leq}_{\mathcal{I}}$ are NP-COMPLETE.

Table 1 provides a summary of the defined problems and their complexity results.

Experimental Results

We present some results for assessing the computational effort of rescheduling plans according to FIXABILITY$_\mathcal{I}$ and FIXABILITY$_{\mathcal{WS}}$, obtained by planning over the reformulated problems generated by $\mathcal{R}_\mathcal{I}$ and $\mathcal{R}_{\mathcal{WS}}$, i.e., $\Pi^{\pi_t}_{\mathcal{R}_\mathcal{I}}$ and $\Pi^{\pi_t}_{\mathcal{R}_{\mathcal{WS}}}$, compared to replanning from scratch (REPLAN). We tested two heuristics, i.e., h^{add} and h^{max}, implemented in ENHSP20 (Scala et al. 2020), and UPMURPHI (Penna, Magazzeni, and Mercorio 2012). As benchmarks, we considered those used by Percassi, Scala, and Vallati (2023b). The invalid plans to be fixed were generated by injecting uniformly distributed noise on the timestamps of valid plans. All experiments were run on an Intel Xeon Gold 6140M CPU with 2.30 GHz, with a cutoff time of $1,800$ seconds, and 8GB of RAM.

Figure 1 shows the number of expanded nodes for generating a plan and fixing an invalid plan. Intuitively, when

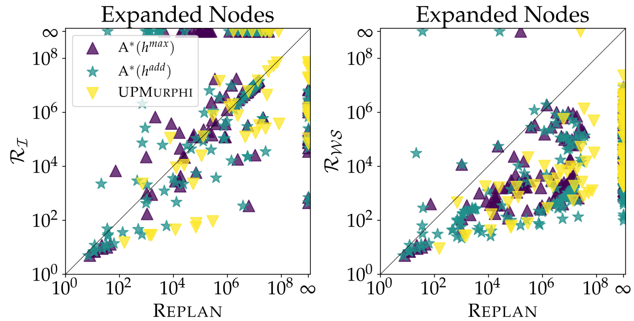

Figure 1: Expanded Nodes of replanning from scratch (REPLAN) versus plan fixing for $\mathcal{R}_\mathcal{I}$ and $\mathcal{R}_{\mathcal{WS}}$.

rescheduling a plan without additional constraints ($\mathcal{R}_\mathcal{I}$), the situation is mixed, while when the possibility of manipulating the plan is limited ($\mathcal{R}_{\mathcal{WS}}$), plan fixing is advantageous.

Discussion

We have presented the concept of PDDL+ FIXABILITY, which subsumes a range of problems such as plan rescheduling and validation, whose solutions can support the usage of PDDL+ in complex real-world scenarios. To solve FIXABILITY problems, we introduced a set of translations that allow leveraging existing planning engines. Future work will focus on expanding FIXABILITY specialisations to additional cases, such as supporting the deletion or addition of actions.

Acknowledgements

Francesco Percassi and Mauro Vallati were supported by a UKRI Future Leaders Fellowship [grant number MR/T041196/1]. Enrico Scala has been partially supported by AIPlan4EU, a project funded by EU Horizon 2020 research and innovation programme under GA n. 101016442.

References

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

Penna, G. D.; Magazzeni, D.; and Mercorio, F. 2012. A universal planning system for hybrid domains. *Applied Intelligence*, 36(4): 932–959.

Percassi, F.; Scala, E.; and Vallati, M. 2022. The Power of Reformulation: From Validation to Planning in PDDL+. In *Proc. of ICAPS 2022*, 288–296.

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

Percassi, F.; Scala, E.; and Vallati, M. 2023b. Fixing Plans for PDDL+ Problems: Theoretical and Practical Implications. In *Proc. of ICAPS 2023*.

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