The Power of Reformulation: From Validation to Planning in PDDL+

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

The 32nd International Conference on Automated Planning and Scheduling
ICAPS 2022

PDDL+ planning problem

- A PDDL+ planning problem is a tuple $\Pi = \langle F, X, I, G, A, E, P\rangle$ where: F and X are the sets of **propositional** and **numeric variables**; I and G are the **initial** and **goal state**; A is the set of **actions** and E and P are the sets of **events** and **processes**.
- A PDDL+ plan is a tuple $\langle \pi, \langle t_s, t_e\rangle\rangle$ where: $\pi = \langle\langle a_1, t_1\rangle, \dots, \langle a_n, t_n\rangle\rangle$, with $t_i \in \mathbb{Q}$, is a sequence of time-stamped actions and $\langle t_s, t_e\rangle$, with $t_s, t_e \in \mathbb{Q}$, is the **temporal envelope** within π is executed.

Processes and events

Processes and **Events** are used to model the **exogenous** and **environmental changes**.

- A **process** ρ is a tuple $\langle pre(\rho), eff(\rho)\rangle$ where $pre(\rho)$ is a logical formula involving conditions over F and X and $eff(\rho)$ is a set of **continuous numeric effects** having the form $\langle x, \xi\rangle$, where $x \in X$ and ξ is a mathematical expression defined over X and $\mathbb{Q}$; if $pre(\rho)$ holds and time passes continuously then ρ contributes additively to the **first derivative** of x with ξ for each $\langle x, \xi\rangle \in eff(\rho)$.
- An **event** ε is a tuple $\langle pre(\varepsilon), eff(\varepsilon)\rangle$ where $pre(\varepsilon)$ is a logical formula and $eff(\varepsilon)$ is a set of conditional effects $c \triangleright e$ where c is a logical formula and e is a set of **numeric** and **Boolean assignments** having the form $\langle\{inc, dec, asgn\}, x, \xi\rangle$ and $\langle f := \{\perp, \top\}\rangle$, respectively; if $pre(\varepsilon)$ is triggered then the state changes instantaneously according to $eff(\varepsilon)$.

PDDL+ semantics: *Intuitively, a PDDL+ problem consists of finding a number of time-stamped actions along with a potentially infinite timeline, whilst conforming to a number of processes and events that may change the state of the world in a **continuous** or an **instantaneous** manner as time goes by.*

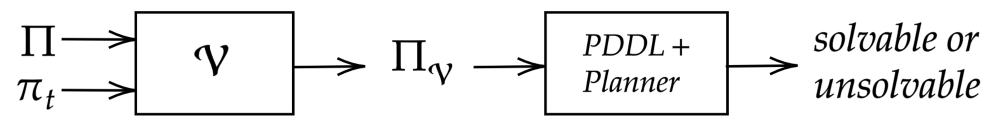
Research Question

Definition 1 ((Discrete) Validation Problem). *Let Π be a PDDL+ problem, π_t be a plan for Π. The* validation problem *aims at establishing whether π_t is valid for Π. The* discrete validation problem *aims at establishing whether π_t is valid for Π under $\delta \in \mathbb{Q}$ discretisation.*

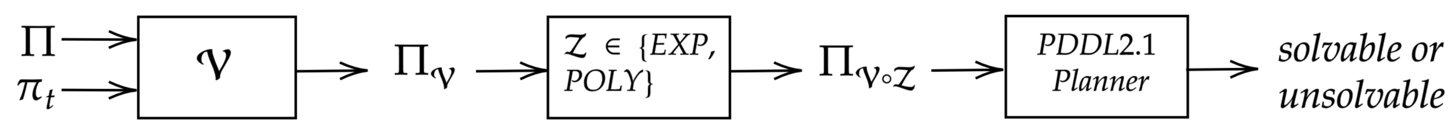
- **Can reformulation be used to validate PDDL+ plans using planning techniques?**

Methodology

Given Π and π_t, we **reformulate** the validation problem into a PDDL+ planning problem $\Pi_{\mathcal{V}}$ which is solvable (unsolvable) iff π_t is valid (invalid) w.r.t. Π.



We also reformulate $\Pi_{\mathcal{V}}$ through $\mathcal{Z} \in \{$POLY, EXP$\}$, which are translations for **discretising** a PDDL+ problem into a numeric one (Percassi et al., 2021), to produce a PDDL2.1 planning problem $\Pi_{\mathcal{V}\circ\mathcal{Z}}$ used for doing discrete PDDL+ validation.



Depending on the planning engine used to reason over $\Pi_{\mathcal{V}}$, it is possible to validate plans w.r.t. **discrete** or **continuous semantics**.

	Continuous Semantics	Discrete Semantics	Polynomial Dynamic	Non-Polynomial Dynamic
VAL	✓	✗	✓	✗
OUR	✓	✓	✓	✓

Baseline translation $\mathcal{V}_0$

Given $\Pi = \langle F, X, I, G, A, E, P\rangle$ and $\pi_t = \langle \pi, \langle t_s, t_e\rangle\rangle$ plan for Π, $\mathcal{V}_0$ produces a new validating PDDL+ problem $\Pi^{\pi_t}_{\mathcal{V}_0} = \langle F \cup F_A \cup \{\mathsf{T}\}, X \cup \{time\}, I \cup \{a\text{-}d_0, \langle time := t_s\rangle, \mathsf{T}\}, G \wedge a\text{-}d_n \wedge \langle time = t_e\rangle, A_\pi, E, P \cup \{\rho_{time}\}\rangle$ such that $\rho_{time} = \langle \mathsf{T}, \{\langle time, 1\rangle\}\rangle$ and:

$$F_A = \bigcup_{i=0}^{n=|\pi|} \{a\text{-}d_i\}$$

$$A_\pi = \bigcup_{\langle a_i, t_i\rangle \text{ in } \pi} \{\langle pre(a_i) \wedge a\text{-}d_{i-1} \wedge \neg a\text{-}d_i \wedge \langle time = t_i\rangle, eff(a_i) \cup \{a\text{-}d_i\}\rangle\}$$

Key aspects: the only actions executable in $\Pi^{\pi_t}_{\mathcal{V}_0}$ are those from π_t. These actions can only be executed in the same ordering and at the same time-stamps prescribed by π_t.

Constrained translations $\mathcal{V}_{\mathsf{U}}$ and $\mathcal{V}_{\mathsf{D}}$

$\mathcal{V}_{\mathsf{U}}$: Given Π and π_t plan for Π, $\mathcal{V}_{\mathsf{U}}$ produces $\Pi^{\pi_t}_{\mathcal{V}_{\mathsf{U}}} = \langle F \cup F_A \cup \{\mathsf{T}\}, X \cup \{time\}, I \cup \{a\text{-}d_0, \langle time := t_s\rangle, \mathsf{T}\}, G \wedge a\text{-}d_n \wedge \langle time = t_e\rangle, A_\pi, E, P_{\mathsf{U}}\rangle$, where:

$$P_{\mathsf{U}} = \bigcup_{\rho \in P \cup \{\rho_{time}\}} \{\langle \langle time < t_e\rangle \wedge pre(\rho), eff(\rho)\rangle\}$$

The main idea of $\mathcal{V}_{\mathsf{U}}$ is to avoid some dead-ends by disallowing the occurrence of any process whenever time gets beyond what is prescribed by the plan. So, when time flows beyond t_e, then the processes P_{U} can not cause any change.

$\mathcal{V}_{\mathsf{D}}$: Given Π and π_t plan for Π, $\mathcal{V}_{\mathsf{D}}$ produces $\Pi^{\pi_t}_{\mathcal{V}_{\mathsf{D}}} = \langle F \cup F_A \cup \{\mathsf{T}\}, X \cup \{time\}, I \cup \{a\text{-}d_0, \langle time := t_s\rangle, \mathsf{T}\}, G \wedge a\text{-}d_n \wedge \langle time = t_e\rangle \wedge \mathsf{T}, A_\pi, E \cup E_{\mathsf{D}}, P_{\mathsf{D}} \cup \{\rho_{time}\}\rangle$, where:

$$E_{\mathsf{D}} = \bigcup_{\substack{\pi(t') \text{ in } \pi \\ \pi(t') \neq \langle\rangle \wedge t' \neq t_e}} \{\langle \langle time > t'\rangle \wedge \neg a\text{-}d_{last(\pi(t'))} \wedge \mathsf{T}, \{\neg \mathsf{T}\}\rangle\}$$

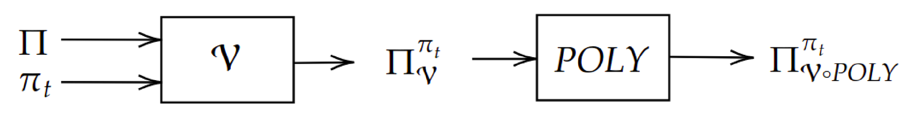
$$P_{\mathsf{D}} = \bigcup_{\rho \in P} \{\langle pre(\rho) \wedge \mathsf{T}, eff(\rho)\rangle\}$$

The main idea of $\mathcal{V}_{\mathsf{D}}$ is to prevent time from flowing unless all the actions prescribed by π_t up to the current instant have been executed.

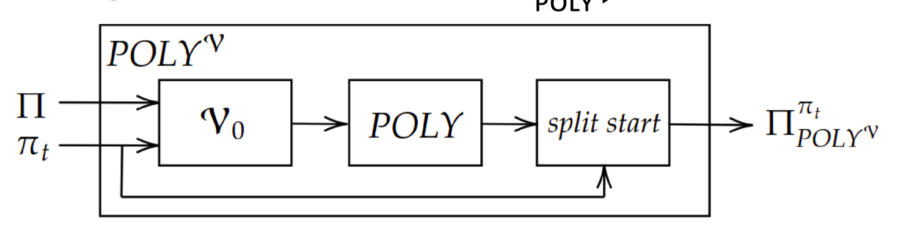
$\mathcal{V}_{\mathsf{UD}}$: Jointly usage of $\mathcal{V}_{\mathsf{U}}$ and $\mathcal{V}_{\mathsf{D}}$, i.e., $\mathcal{V}_{\mathsf{UD}} = \mathcal{V}_{\mathsf{U}} \circ \mathcal{V}_{\mathsf{D}}$.

Can we use PDDL2.1 engines to validate PDDL+ plans?

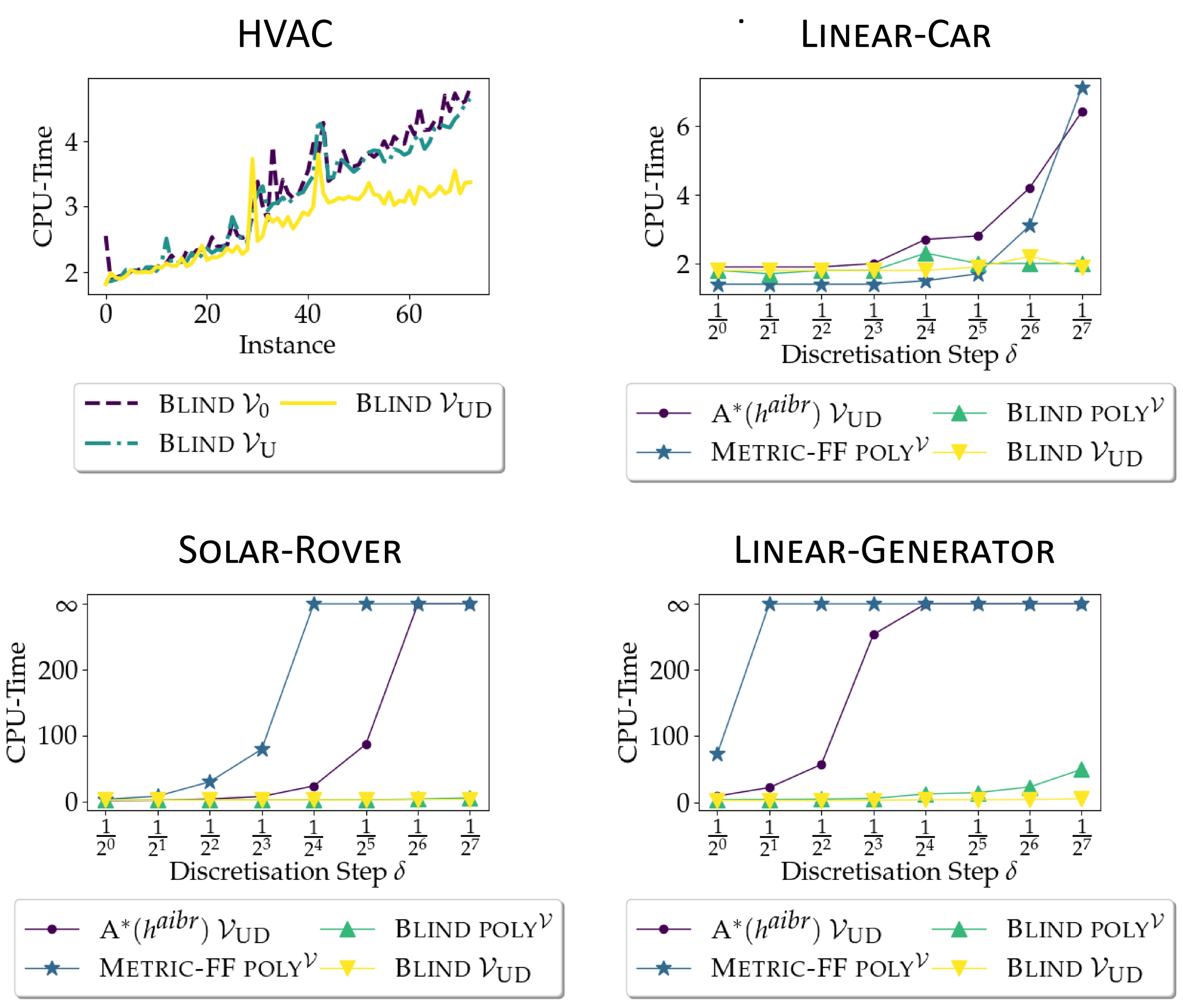
To **enable** the usage of numeric planners for doing PDDL+ validation we concatenate $\mathcal{V} \in \{\mathcal{V}_0, \mathcal{V}_{\mathsf{U}}, \mathcal{V}_{\mathsf{D}}, \mathcal{V}_{\mathsf{UD}}\}$ with POLY. The resulting translation is denoted as $\mathcal{V} \circ$ POLY and the resulting PDDL2.1 validating task is denoted as $\Pi^{\pi_t}_{\mathcal{V}\circ\text{POLY}}$.



This chaining **obscures** the search properties provided by the constrained translations when $\mathcal{V} \in \{\mathcal{V}_{\mathsf{U}}, \mathcal{V}_{\mathsf{D}}\}$. We designed a variant of POLY, namely POLY$^{\mathcal{V}}$, which is **aware of the validation problem**. POLY$^{\mathcal{V}}$ is **equivalent** in terms of search properties to $\mathcal{V}_{\mathsf{UD}}$. The resulting PDDL2.1 validating task is denoted as $\Pi^{\pi_t}_{\text{POLY}^{\mathcal{V}}}$.



Experimental Analysis



We can efficiently and effectively validate plans, even complex ones.

Selected Bibliography

- Abdulaziz, M.; and Lammich, P. 2018. A Formally Verified Validator for Classical Planning Problems and Solutions. In Proc. of ICTAI 2018, 474–479.
- Fox, M.; and Long, D. 2003. PDDL2.1: PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *J. Artif. Intell. Res.* 20: 61–124.
- Fox, M.; and Long, D. 2006. Modelling Mixed Discrete-Continuous Domains for Planning. *J. Artif. Intell. Res.* 27:235–297.
- Percassi, F.; Scala, E.; and Vallati, M. 2021. Translations from Discretised PDDL+ to Numeric Planning. In Proc. of ICAPS 2021, 252–261.