Temporal Planning with Temporal Metric Trajectory Constraints

Andrea Micheli **Enrico Scala**

Embedded Systems Unit, Fondazione Bruno Kessler, Italy

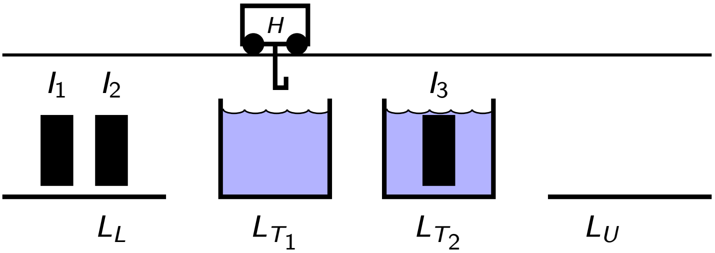
January 23, 2019

AAAI 2019, Honolulu, HA, USA

Context

- Industrial automation requires highly coupled planning and scheduling constraints
  - Logistics (e.g., deadlines and task allocation)
  - Synchronization in industrial production (e.g., processing durations, collision avoidance)
  - Autonomous agents: AGVs, UAVs (e.g., timed coordination, multi-goal)
- Model-based approaches are highly desirable to make automation as flexible as possible $\Rightarrow$ AI planning

Application Example: Hoist Scheduling Problem



- Each item I_i must follow a "recipe" consisting of a sequence of chemical baths (each in a type of tank) and **processing timing constraints** (min-max in each tank) - (Scheduling part)
- Multiple hoists can be used to move items, and need collision avoidance - (Planning part)
- Goal: synthesize hoists movement plan to process a set of items according to their recipes

Motivation of this work

- Temporal constraints are hard for AI Planning
  - Representation :: Encoding in standard AI planning languages is "possible" but cumbersome
  - Reasoning :: Existing solving techniques do not scale well with complex temporal constraints

Practically, people either choose classical planning (and abstract the scheduling problem away) or switch to scheduling (and abstract the planning problem away)

- **Research Questions:** How can we express and efficiently solve planning problems with expressive temporal constraints?
- **Main Idea:** Planning and scheduling at the very same level of representation. Express action time constraints explicitly, and not implicitly

Outline

Temporal Knowledge

Intuition

Allowing to predicate (temporally) over action instances within a plan using first-order temporal metric constraints

Example

$$\bar{\forall} l_0 : load(I_x, L_L).\bar{\exists} u_1 : unload(I_x, L_{T_1}).\bar{\exists} l_1 : load(I_x, L_{T_1}).$$
$$\bar{\exists} u_2 : unload(I_x, L_{T_2}).\bar{\exists} l_2 : load(I_x, L_{T_2}).\bar{\exists} u_3 : unload(I_x, L_U).$$
$$(l_0 \leq u_1) \wedge (10 \leq (l_1 - u_1) \leq 12) \wedge (l_1 \leq u_2) \wedge$$
$$(20 \leq (l_2 - u_2) \leq 21) \wedge (l_2 \leq u_3)$$

Formalization: the TPP Problem

Causal Model	Classical Planning Actions	$\models$	Planning Theory (Domain, Instance)
Temporal Model	Timing of Actions	$\models$	Temporal Knowledge (first-order logic)

- TPP = Classical Planning $\oplus$ Temporal Knowledge
- Temporal knowledge is a set of axioms:
  - Temporal Axiom:
    - Boolean combination of $v_1 - v_2 \leq k$ with v_1 and v_2 action vars, $k \in \mathbb{Q}$
    - $\bar{\forall} v : a.\phi$ where v is a fresh action variable, a is a ground classical action and ϕ is a temporal axiom
    - $\bar{\exists} v : a.\phi$ where v is a fresh action variable, a is a ground classical action and ϕ is a temporal axiom
- Solution: partial order over a set of action instances $a_1; a_2; \cdots ; a_n$. It is valid iff causally and temporally consistent

Formalization: the TPP Problem

Highlights

- Arbitrary nesting of action quantifiers is allowed!
- Can be used to capture significant fragments of well known temporal planning formalisms (i.e., ANML, PDDL2.1)
- Planning problem with TK proven decidable (with integral time; over continuous time is an open question)

How do we solve it?

Outline

Heuristic Forward Search :: Sketch of TPACK

Semi Symbolic State Space

- Classical state with temporal commitments (i.e., action orderings, existential matching)
- (LAZY search) Do classical planning and check temporal consistency on goal states
- (EAGER search) Splitting over different temporal scenarios within the search. Enables checking online with a theory solver (i.e., Simple Temporal Network)

Temporally Aware Subgoaling Based Heuristics

- Observation: $h^1(P)$ relaxes TPP too!
- Enhancement: temporal commitments as online subgoals. This yields deep version of $h^1(P)$, namely $h^1_{dtk}(P, TK)$
    - Existential matching yields request within the search (e.g., connection between unload and load in the HSP example)
    - Automatically exploiting temporal knowledge for heuristic purposes

Outline

Experimental Setup

Competitors

- TPACK: our planner
  - Different heuristics
- Other Planners (PDDL 2.1):
  - OPTIC
  - ITSAT
  - Temporal Fast Downward (TFD)

Benchmarks

- International Planning Competition (IPC) domains
- Highly temporally expressive domains
  - Industrial domains
    - HSP
    - MaJSP (Multi-agent planning variant of JSP)
  - IPC domains with intermediate effects

Coverage Results for IPC Problems

	Domain (# inst.)	ITSAT	OPTIC	TFD	TPACK (h_{dtk})	TPACK (h_{atk})	TPACK (h_{add})	TPACK (Lazy)
Temporal IPC (d)	DRIVERLOG (20)	**18**	15	6	14	14	14	5
	FLOORTILE (8)	**8**	7	6	4	4	4	2
	MAPANALYSER (20)	**20**	0	**20**	**20**	**20**	**20**	0
	MATCHCELLAR (10)	**10**	9	**10**	6	3	1	3
	SATELLITE (20)	**20**	14	**20**	9	9	7	3
	TMS (20)	**20**	1	1	1	1	0	0
	Total (98)	**96**	46	63	54	51	46	13

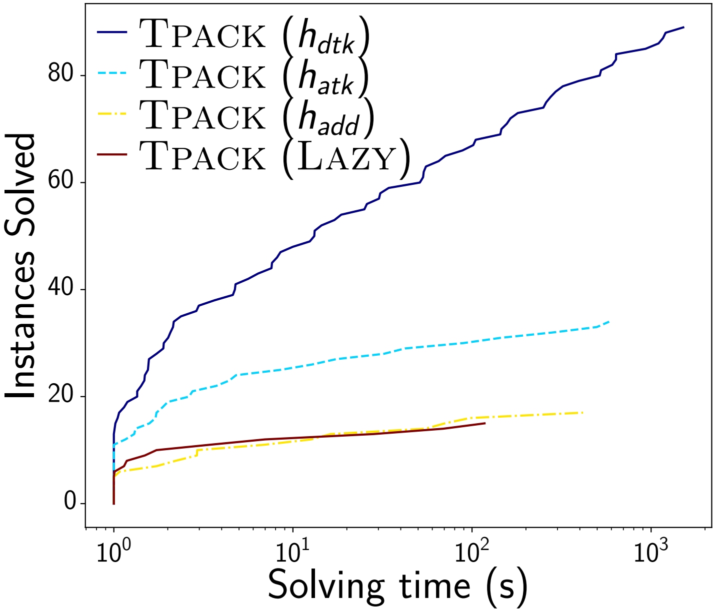
- Number of instances solved, higher is better
- TPACK complementary to forward search planner (OPTIC and TFD) but dominated by ITSAT

Coverage Results for Industrial Problems

HSP (b)	#Items	ITSAT clip*	ITSAT cont.*	OPTIC clip	OPTIC cont.	TPACK (h_{dtk})	TPACK (h_{atk})	TPACK (h_{add})	TPACK (Lazy)
	1	0	2	1	1	**10**	**10**	**10**	**10**
	2	0	0	0	0	**10**	9	2	3
	3	0	0	0	0	**10**	5	1	1
	4	0	0	0	0	**10**	3	1	1
	5	0	0	0	0	**10**	3	1	1
	6	0	0	0	0	**9**	1	1	0
	7	0	0	0	0	**9**	1	1	0
	8	0	0	0	0	**8**	1	1	0
	9	0	0	0	0	**7**	1	0	0
	10	0	0	0	0	**7**	1	0	0
	Tot.	0	2	1	1	**90**	35	18	16

MAJSP (c)	#Jobs	ITSAT clip*	ITSAT cont.*	OPTIC clip	OPTIC cont.	TPACK (h_{dtk})	TPACK (h_{atk})	TPACK (h_{add})	TPACK (Lazy)
	1	NA	NA	22	0	56	56	**57**	**57**
	2	NA	NA	0	0	41	40	37	**46**
	3	NA	NA	0	0	25	25	23	**29**
	4	NA	NA	0	0	11	10	10	**12**
	Tot.	NA	NA	22	0	133	131	127	**144**

Solving Times of TPACK for HSP



Temporally Expressive Variants of IPC

	Domain (# inst.)	ITSAT clip*	ITSAT cont.*	OPTIC clip	OPTIC cont.	TPACK (h_{dtk})	TPACK (h_{atk})	TPACK (h_{add})	TPACK (Lazy)
Uncertainty IPC (e)	DRIVERLOG (20)	0	0	0	5	6	**9**	8	7
	FLOORTILE (8)	0	0	0	0	**4**	**4**	**4**	1
	MAPANALYSER (20)	0	0	0	0	2	5	**6**	0
	MATCHCELLAR (10)	4	0	3	**10**	5	1	1	2
	SATELLITE (20)	1	0	0	1	**5**	**5**	4	2
	TMS (20)	**1**	0	0	0	0	0	0	0
	Total (98)	6	0	3	16	22	**24**	23	12

- Instances obtained by introducing intermediate effects. This encodes problems with uncontrollable durations
- TPACK achieves higher score, except in MATCHCELLAR

Outline

Conclusions

Contributions

- Formal account for temporally expressive planning problems
  - Grounded on Classical Planning
  - Provide powerful temporal constructs
- Heuristic search framework to solve TPP
  - Novel search schema and temporally informed heuristics
  - Extend reach of temporal planners to industrial problems

Future Work

- Further theoretical investigation on the expressiveness and complexity of TPP
- Additional temporal constructs in TK to capture common temporal patterns
- Experimental analysis with Timeline and ANML planners

Thanks for your attention!

Backup Slides

h_{add}

$$h_{add}(G) = \begin{cases} 0 & \text{if } s \models G \\ \min_{a \in ach(G)} h_{add}(\text{pre}(a)) + 1 & \text{if } |G| = 1 \\ \sum_{g \in G}(h_{add}(g)) & \text{if } |G| > 1 \end{cases}$$

$\textsc{Eager-Search}$ Successor Function

function $\textsc{Succ}(s$ - State, a - Action$)$
    $\mu' \leftarrow \textsc{ClassicalApply}(s.\mu,\ a)$
    $\mathcal{S} \leftarrow \{\langle \mu', s.\pi + \langle \mathit{ActionInstance}(a) \rangle, s.\rho, s.\sigma \rangle\}$
    for all $\bar{y} \in s.\rho$ s.t. $\mathit{type}(\bar{y}) = a$ **do**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{\langle \mu', s.\pi + \langle \bar{y} \rangle, s.\rho \setminus \{\bar{y}\}, s.\sigma \rangle\}$
    for all $\gamma \in \mathit{TK}$ and a is universally quantified in γ **do**
        $\mathcal{S} \leftarrow \bigcup_{s' \in \mathcal{S}} \textsc{apply}(s', \gamma', \emptyset)$
    return $\{s \in \mathcal{S} \mid \mathit{TN}(s)$ is consistent$\}$

function $\textsc{Apply}(s$ - State, γ - Temporal Axiom, b - $\bar{\forall}\mathit{Bind})$
    $\mathcal{S} \leftarrow \emptyset$
    if γ is Quantifier Free **then**
        $\mathcal{S} \leftarrow \{s\}$
    else if $\gamma = \bar{\forall} x : a.\gamma'$ **then**
        $\mathcal{S} \leftarrow \{s\}$
        for all $\bar{y} \in \pi$ s.t. $\mathit{type}(\bar{y}) = a$ **do**
            $\mathcal{S} \leftarrow \bigcup_{s' \in \mathcal{S}} \textsc{apply}(s', \gamma', b \cup \{x = \bar{y}\})$
    else if $\gamma = \bar{\exists} x : a.\gamma'$ **then**
        for all $\bar{y} \in \pi \cup s.\rho$ s.t. $\mathit{type}(\bar{y}) = a$ **do**
            $\sigma' \leftarrow s.\sigma \cup \{f_x(b) = \bar{y}\}$
            $\mathcal{S} \leftarrow \mathcal{S} \cup \textsc{apply}(\langle s.\mu, s.\pi, s.\rho, \sigma' \rangle, \gamma', b)$
        $\bar{a}' \leftarrow \mathit{ActionInstance}(a)$
        $\sigma' \leftarrow s.\sigma \cup \{f_x(b) = \bar{a}'\}$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \textsc{apply}(\langle s.\mu, s.\pi, s.\rho \cup \{\bar{a}'\}, \sigma' \rangle, \gamma', b)$
    return $\mathcal{S}$

Decidability

Theorem

TPP *is decidable if interpreted over integer time.*

Proof sketch.

We reduce TPP to the satisfiability of a TPTL formula with past operators, that is decidable. TPTL offers the usual LTL operators e.g., $\Box$ (for always in the future) and $\Diamond$ (for eventually); the past operators e.g., $\boxminus$ (for always in the past) and $\diamondminus$ (for once in the past); and "freezing quantifiers" of the form "$x.$" referring to the timing of states. The classical planning part of TPP has a known compilation in the LTL fragment of TPTL, while the temporal action axioms can be encoded using freezing quantifiers in combination with temporal operators. As for our formalism, TPTL allows Boolean combinations of binary temporal constraints expressed over the variables of freezing quantifiers, hence we only need to encode the universal and existential quantification as freezing quantifications. This can be done by translating $\bar{\forall} v : a.\alpha$ into $\Box v.(p_a \rightarrow [\alpha]) \vee \boxminus v.(p_a \rightarrow [\alpha])$ where $[\alpha]$ indicates the recursive compilation of the right-hand side of the axiom, and p_a encodes the states in which action a is triggered. Similarly, $\bar{\exists} v : a.\alpha$ translates to $\Diamond v.(p_a \wedge [\alpha]) \vee \diamondminus v.(p_a \wedge [\alpha])$. $\Box$

Solving the TPP Problem

Challenges

- Temporal Axiom in TK use dense time: enumeration seems not practical
- Given a candidate classical plan, matching quantified variables to actions is combinatorial (dealing with multiple occurrences of the same action, e.g. moves in HSP)
- The same classical state might need to occur multiple times in a valid plan (loops; subsumption is problematique)
- TK allows for arbitrary action quantifier nesting

Solution Schema

- Forward state-based search exploring causal valid plans
- Heuristics to capture temporal aspects and causal aspects together

Forward-Search Schema

- **Basic**
  - Classical states augmented with lifted time representation (similar to POPF)
  - Search records into states the matching of variables with actions in the explored prefix
- **Variants**
  - LAZY-SEARCH: solving the matching problem only for valid classical plans
  - EAGER-SEARCH: solving the matching problem online. This anticipates some of the temporal commitment during search increasing prediction. Lead to predict a set of existential quantified actions, i.e. matched $\bar{\exists}$

Heuristics

Devised as extensions of h_{add}

h_{add} $\xrightarrow{\text{relaxes}}$ Classical Planning $\xrightarrow{\text{relaxes}}$ TPP

h_{add} does not consider TK in any way, in particular existential quantifiers:

- $h_{atk} = h_{add} + |\text{matched } \bar{\exists}|$
- h_{dtk} injects artificial goals to the abstracted problem to account for the preconditions of actions needed because of the existential matching

Related work

- Durative actions
  - encode the execution of an action as an interval
  - can express complex temporal constraints yet requiring convoluted constructions (e.g. clip/strut/container)
    - E.g. HSP requires $3 \times (n \times m \times k)$ actions in PDDL 2.1
  - Practical only for domains with loose temporal constraints
- PDDL3
  - Trajectory constraints over state traces: different from scheduling where actions are constrained
  - Limitations: no LTL operator nesting nor metric constraints
- HTN/ANML
  - Can express intermediate effects and action decompositions, no direct encoding of complex temporal constraints (additional actions and constructions needed)
- Timelines
  - Rich temporal constructs but scarce support/efficiency for causal constructs