ON TIMELINE-BASED GAMES AND THEIR COMPLEXITY

Nicola Gigante
Free University of Bozen-Bolzano, Italy

joint work with **Angelo Montanari**
University of Udine, Italy

and **Andrea Orlandini**
CNR-ISTC, Rome, Italy

and **Marta Cialdea Mayer**
University of Roma Tre, Rome, Italy

and **Mark Reynolds**
The University of Western Australia

ICAPS 2020
October 19–30, 2020
~~Nancy, (France)~~
The Internet (World)

TIMELINE-BASED PLANNING

Timeline-based planning is an approach to planning mostly focused on temporal reasoning:

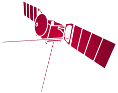
- no clear separation between actions, states, and goals;
- planning problems are modeled as systems made of a number of independent, but interacting, components;
- components are described by state variables;
- the timelines describe their evolution over time;
- the evolution of the system is governed by a set of temporal constraints called synchronization rules.

Timeline-based planning was born in the space operations field, and has been used in real-world mission planning and scheduling systems, both on-board and on-ground.

HSTS [6]
EUROPA [1]
ASPEN [3]

APSI-TRF [2]
GOAC [5]

Mars orbiter

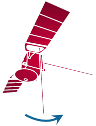


Toy example of a Mars orbiter doing scientific measurements:

1. Three “pointing modes”: Mars, Slewing, Earth
2. Four “activities”: Science, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

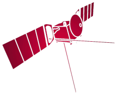
Mars orbiter



Toy example of a Mars orbiter doing scientific measurements:

1. Three “pointing modes”: Mars, Slewing, Earth
2. Four “activities”: Science, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

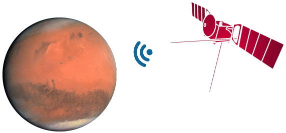
Mars orbiter


Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, Slewing, Earth
2. Four "activities": Science, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

Mars orbiter

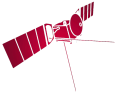


Toy example of a Mars orbiter doing scientific measurements:

1. Three “pointing modes”: Mars, Slewing, Earth
2. Four “activities”: **Science**, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

Mars orbiter

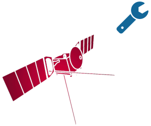


Toy example of a Mars orbiter doing scientific measurements:

1. Three “pointing modes”: Mars, Slewing, Earth
2. Four “activities”: Science, **Communication**, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

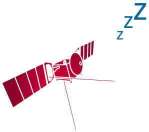
Mars orbiter



Toy example of a Mars orbiter doing scientific measurements:

1. Three “pointing modes”: Mars, Slewing, Earth
2. Four “activities”: Science, Communication, **Maintenance**, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

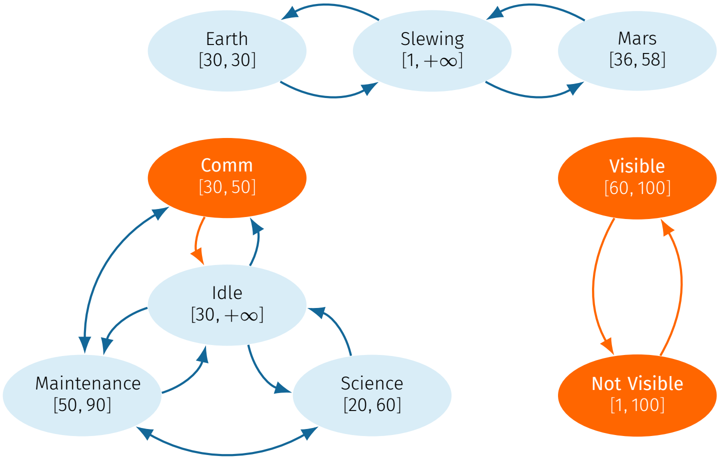
Mars orbiter

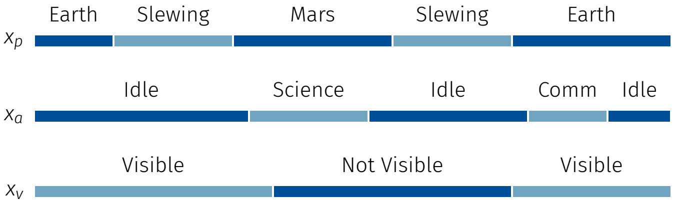

Toy example of a Mars orbiter doing scientific measurements:

1. Three “pointing modes”: Mars, Slewing, Earth
2. Four “activities”: Science, Communication, Maintenance, **Idle**
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

State variables



Timelines



Timelines are sequences of **tokens**;

- time intervals where the variable holds a single value

Synchronisation rules

The interaction of the components is governed by the synchronization rules.

Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = \mathit{Science}] \rightarrow \exists b[x_p = \mathit{Mars}]\ .\ \mathit{start}(b) \leqslant \mathit{start}(a) \wedge \mathit{end}(a) \leqslant \mathit{end}(b)$$

for all tokens a **where** $x_a = \mathit{Science}$,
there is a token b **where** $x_p = \mathit{Mars}$,
such that a **is contained in** b.

Synchronisation rules

The interaction of the components is governed by the synchronization rules.

Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \to \exists b[x_p = Mars]\ .\ start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

for all tokens a where $x_a = Science$,
there is a token b where $x_p = Mars$,
such that a is contained in b.

Synchronisation rules

The interaction of the components is governed by the synchronization rules.

Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \rightarrow \exists b[x_p = Mars] \,.\, start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

for all tokens a where $x_a = Science$,
  there is a token b where $x_p = Mars$,
    such that a is contained in b.

Synchronisation rules

The interaction of the components is governed by the synchronization rules.

Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = \mathit{Science}] \to \exists b[x_p = \mathit{Mars}]\ .\ \mathit{start}(b) \leqslant \mathit{start}(a) \wedge \mathit{end}(a) \leqslant \mathit{end}(b)$$

for all tokens a **where** $x_a = \mathit{Science}$,
there is a token b **where** $x_p = \mathit{Mars}$,
such that a is contained in b.

SYNTAX

Each rule has a fixed structure:

$$a[x = u] \to \underbrace{\exists b[y = v]\ .\ \langle body \rangle} \vee \exists c[z = w] d[k = r]\ .\ \langle body \rangle \vee \ldots$$

trigger

existential statement

The body is made of a **conjunction** of atomic **temporal relations**:

$$start(a) \leqslant_{[l,u]} end(b)$$

endpoint

token name

lower bound
$l \in \mathbb{N}$

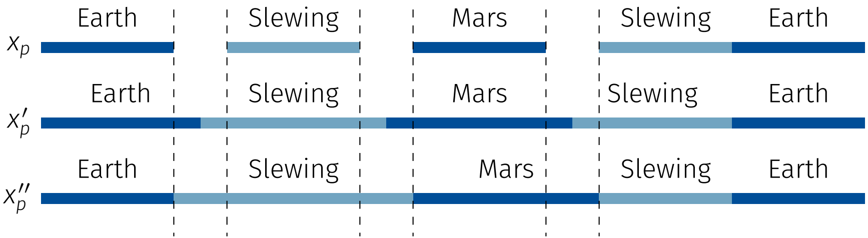
upper bound
$u \in \mathbb{N} \cup \{+\infty\}$

UNCERTAINTY

Current timeline-based systems excel at integrating planning with execution by handling temporal uncertainty.

FLEXIBLE PLANS

Temporal uncertainty is currently handled by **flexible plans**, which represent a set of possible solutions through flexibility intervals:



To be sure they are executable, flexible plans are then checked for **weak/strong/dynamic controllability**, similarly to STNUs.

LIMITATIONS OF FLEXIBLE PLANS

The focus on temporal uncertainty means flexible plans cannot represent strategies involving non-temporal choices.

- flexible plans are inherently sequential;
- the control strategy can only choose the timings of the already fixed sequence of tokens;
- if the expected non-temporal behavior of external variables mismatches during the execution, re-planning is needed.

We want to extended the approach to handle general nondeterminism.

GAME-THEORETIC APPROACH

We propose to approach timeline-based planning with uncertainty in game-theoretic terms.

- We define the timeline-based planning game as a two-player game;
- the controller tries to satisfy the given set of synchronization rules;
- the environment plays arbitrarily.

A timeline-based game is a tuple $G = (SV_C, SV_E, S, D)$.

- Two players, Charlie (the controller) and Eve (the environment);
- players play by starting and ending tokens, building a plan;
- **Charlie** can start tokens for variables in SV_C, **Eve** those for variable in SV_E;
- **Charlie** decides when to stop **controllable** tokens, while **Eve** decides when to stop **uncontrollable** ones;
- **Charlie** tries to satisfy the set S of **system rules**, whatever the behavior of Eve;
- **both** players are assumed to play as to satisfy the set D of **domain rules**.

STRATEGIES

We want to guarantee the existence of a winning strategy for Charlie.

- a strategy is a function σ that given a partial plan gives the next move of the player (i.e. which token to start/end, if any).
- a strategy σ is **admissible** if any play played according to σ will eventually satisfy D.
- a strategy σ_C for Charlie is **winning** if, for any admissible strategy σ_E for Eve, any play played according to σ_C and σ_E is going to satisfy $S \cup D$.

RESULT

Theorem

Winning strategies for timeline-based games are strictly more general than dynamically controllable flexible plans

In particular:

- given a dynamically controllable flexible plan, a winning strategy for the corresponding game exists
- there are solvable problems that admit a winning strategy but not a dynamically controllable flexible plan

ADVANTAGES

Charlie has a winning strategy if he can play to satisfy the rules no matter what Eve does, supposing rules in D are satisfied.

- a general form of nondeterminism is handled in this way, not only temporal uncertainty;
- no need for re-planning, as the winning strategy can already handle any behavior of Eve;
- greater modeling flexibility: domain rules allow to describe complex interactions between the agent and the environment;
- provably subsumes the approach based on dynamically controllable flexible plans;
- but how hard is it to find such a strategy?

RESULT

Theorem

Deciding whether a given timeline-based game admits a winning strategy for Charlie is **2EXPTIME-complete**

Proof ideas:

- problem solved by encoding it into an **ATL* model-checking** problem
- hardness proved by reduction from **domino tiling games**

CONCLUSIONS

A new game-theoretic formulation of timeline-based planning problems with uncertainty:

- uniform treatment of general nondeterminism and temporal uncertainty
- stricty more general than the current approach based on flexible plans
- finding winning strategies is 2EXPTIME-complete.

Coming soon:

- Controller synthesis!

Thank you

Questions?

nicola.gigante@unibz.it

[1] Javier Barreiro, Matthew Boyce, Minh Do, Jeremy Frank, Michael Iatauro, Tatiana Kichkaylo, Paul Morris, James Ong, Emilio Remolina, Tristan Smith, and David Smith. “EUROPA: A Platform for AI Planning, Scheduling, Constraint Programming, and Optimization.” In: *Prof. of the 4th International Competition on Knowledge Engineering for Planning and Scheduling*. 2012.

[2] A. Cesta, G. Cortellessa, S. Fratini, and A. Oddi. “Developing an End-to-End Planning Application from a Timeline Representation Framework.” In: *Proc. of the 21st Conference on Innovative Applications of Artificial Intelligence (IAAI-09)*. 2009, pp. 66–71.

[3] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, and D. Tran. “ASPEN - Automated Planning and Scheduling for Space Mission Operations.” In: *Proc. of the 8th International Conference on Space Operations*. 2000.

[4] Marta Cialdea Mayer, Andrea Orlandini, and Alessandro Umbrico. “Planning and execution with flexible timelines: a formal account.” In: *Acta Informatica* 53.6-8 (2016), pp. 649–680. DOI: 10.1007/s00236-015-0252-z.

[5] Simone Fratini, Amedeo Cesta, Riccardo De Benedictis, Andrea Orlandini, and Riccardo Rasconi. “APSI-Based Deliberation in Goal Oriente Autonomous Controllers.” In: *Proceedings of the 11th Workshop on Advanced Space Technologies for Robotics and Automation*. 2011.

[6] Nicola Muscettola. “HSTS: Integrating Planning and Scheduling.” In: *Intelligent Scheduling*. Ed. by Monte Zweben and Mark S. Fox. Morgan Kaufmann, 1994. Chap. 6, pp. 169–212.