The Temporal Track of the International Planning Competition

Amanda Coles and Andrew Coles

King’s College London, UK

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No. 730086 (ERGO).
Temporal Planning

In general, activities have **varying durations**:

- Loading a package onto a truck is much quicker than driving the truck;
- Drinking a cup of tea takes longer than making it;
- Procrastinating tasks takes longer than doing them;
- ...
TGP Durative Actions

- All Preconditions must hold at the start of the action;
- Preconditions that do not appear in effects must hold throughout execution;
- Effects are undefined during execution and only guaranteed to hold at the final time point.

Temporal Graph Plan

- Using the action model described above;
- Modified version of Graphplan;
- Makespan optimal;
- Also capable of reasoning about exogenous events/time windows (TILs).

Durative Actions in PDDL 2.1
First Temporal Track @ Third IPC: 2002

over all

pre       pre       pre
   eff          A          eff

at start       at end

PDDL Example (i)

(: action LOAD-TRUCK
  :parameters
  (?obj - obj ?truck - truck ?loc - location)
  :precondition
  (and (at ?truck ?loc)
       (at ?obj ?loc))
  :effect
  (and (not (at ?obj ?loc))
       (in ?obj ?truck))
PDDL Example (i)

(:durative-action LOAD-TRUCK
 :parameters
 (?obj - obj ?truck - truck ?loc - location)
 :duration (= ?duration 2)
 :condition
  (and  (over all (at ?truck ?loc))
        (at start (at ?obj ?loc)))
 :effect
  (and (at start (not (at ?obj ?loc)))
       (at end (in ?obj ?truck)))

Beware of self-overlapping actions!

"Complexity of concurrent temporal planning“, Rintanen J., ICAPS 2007
Durative Actions?

Classical Planner
Durative Actions?

Classical Planner
Temporal Planners in IPC 2003

<table>
<thead>
<tr>
<th>Planner</th>
<th>Solved</th>
<th>Attempted</th>
<th>Success Ratio</th>
<th>Tracks entered</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>237 (+70)</td>
<td>284 (+76)</td>
<td>83% (85%)</td>
<td>S, N, HN</td>
</tr>
<tr>
<td>LPG</td>
<td>372</td>
<td>428</td>
<td>87%</td>
<td>S, N, HN, ST, T</td>
</tr>
<tr>
<td>MIPS</td>
<td>331</td>
<td>508</td>
<td>65%</td>
<td>S, N, HN, ST, T, C</td>
</tr>
<tr>
<td>SHOP2</td>
<td>899</td>
<td>904</td>
<td>99%</td>
<td>S, N, HN, ST, T, C</td>
</tr>
<tr>
<td>Sapa</td>
<td>80</td>
<td>122</td>
<td>66%</td>
<td>T, C</td>
</tr>
<tr>
<td>SimSyn</td>
<td>11</td>
<td>144</td>
<td>8%</td>
<td>S, N</td>
</tr>
<tr>
<td>Simplanner</td>
<td>91</td>
<td>122</td>
<td>75%</td>
<td>S</td>
</tr>
<tr>
<td>Stella</td>
<td>50</td>
<td>102</td>
<td>49%</td>
<td>S</td>
</tr>
<tr>
<td>TALPlanner</td>
<td>610</td>
<td>610</td>
<td>100%</td>
<td>S, ST, T</td>
</tr>
<tr>
<td>TLPlan</td>
<td>894</td>
<td>894</td>
<td>100%</td>
<td>S, N, HN, ST, T, C</td>
</tr>
<tr>
<td>TP4</td>
<td>26</td>
<td>204</td>
<td>13%</td>
<td>N, ST, T, C</td>
</tr>
<tr>
<td>TPSYS</td>
<td>14</td>
<td>120</td>
<td>12%</td>
<td>ST, T</td>
</tr>
<tr>
<td>VHPOP</td>
<td>122</td>
<td>224</td>
<td>54%</td>
<td>S, ST</td>
</tr>
</tbody>
</table>

Winner, Fully Automated: LPG, solved more problems because it also handled temporal domains.
PDDL Example (ii)

(:durative-action open-barrier
 :parameters
 (?loc – location ?p - person)
 :duration (= ?duration 1)
 :condition
 (and (at start (at ?loc ?p)))
 :effect
 (and (at start (barrier-open ?loc))
    (at end (not (barrier-open ?loc))))
PDDL Example (ii)

(:durative-action open-barrier
 :parameters (?loc – location ?p - person)
 :duration (= ?duration 1)
 :condition (and (at start (at ?loc ?p)))
 :effect (and (at start (barrier-open ?loc))
          (at end (not (barrier-open ?loc)))))
Durative Actions in LPGP

(Fox and Long, ICAPS 2003)
Durative Actions in LPGP

(Fox and Long, ICAPS 2003)
Durative Actions in LPGP

(Fox and Long, ICAPS 2003)
Planning with Snap Actions (i)

Challenge 1: What if $B \not\vdash$ interferes with the goal?

- **PDDL 2.1 semantics:** no actions can be executing in a goal state.

- **Solution:** add $\neg A_s$, $\neg B_s$, $\neg C_s$... to the goal
  - (Or make this implicit in a temporal planner.)
Planning with Snap Actions (ii)

• Challenge 2: what about **over all** conditions?
  - If A is executing, inv_A must hold.

• Solution:
  - In every state where As is true: inv_A must also be true
  - Or: (implies (As) inv_A)
  - Violating an invariant then leads to a **dead-end**.
Planning with Snap Actions (iii)

- Challenge 3: where did the durations go?
  - More generally, what are the temporal constraints?
  - Logically sound ≠ temporally sound.
Option 1: Decision Epoch Planning

- Search with **time-stamped states** and a **priority queue** of pending end snap-actions.
  - See e.g. Temporal Fast Downward (Eyerich, Mattmüller and Röger); Sapa (Do and Kambhampati).

- In a state $S$, at time $t$ and with queue $Q$, either:
  - Apply a start snap-action $A$ (at time $t$)
    - Insert $A$ into $Q$ at time $(t + \text{dur}(A))$
    - $S'.t = S.t + \varepsilon$
  - Remove and apply the first end snap-action from $Q$.
    - $S'.t$ set to the scheduled time of this, plus $\varepsilon$

Running through our example...

Can only choose $A \leftarrow$ - eliminated the temporally inconsistent option ($B \leftarrow$ before $A \leftarrow$)

What does this look like if we do Bstart first?
Decision Epoch Planning: The snag

- Must **fix start- and end-timestamps** at the point when the action is started.
  - Used for the priority queue
- Can we always do this?

```
q
C ⊨

D ⊨
q
C ⊨

D ⊨
¬q
```
Decision Epoch Planning: The snag

• Must **fix start- and end-timestamps** at the point when the action is started.
  
  - Used for the priority queue

• Can we always do this?

\[ \text{dur}(C) = 10 \]
\[ \text{dur}(D) = 1 \]
Decision Epoch Planning: The snag

- Must **fix start- and end-timestamps** at the point when the action is started.
  - Used for the priority queue

- Can we always do this?

```
<table>
<thead>
<tr>
<th>t = 0</th>
<th>t = 0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>C⊥</td>
<td>D⊥</td>
</tr>
</tbody>
</table>
```

```
\begin{align*}
\text{dur}(C) &= 10 \\
\text{dur}(D) &= 1
\end{align*}
```
## IPC 2004 Planners

<table>
<thead>
<tr>
<th></th>
<th>ADL</th>
<th>DP</th>
<th>Numbers</th>
<th>Durations</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRIKEY</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>FAP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FD, FDD</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LPG-TD</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Macro-FF</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Marvin</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Optop</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>P-MEP</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Roadmapper</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SGPlan</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Tilsapa</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>YAHSP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Simple Temporal Networks: VHPOP and CRIKEY!

“Temporal Constraint Networks”, Dechter, Meiri and Pearl, Artificial Intelligence, 1991
Option 2: a Simple Temporal Problem

• All our constraints are of the form:
  • \( \varepsilon \leq t(i+1) - t(i) \) \( (c.f. \ sequence \ constraints) \)
  • \( \text{dur}_{\text{min}}(A) \leq t(A_{\leftarrow}) - t(A_{\rightarrow}) \leq \text{dur}_{\text{max}}(A) \)

• Or, more generally, \( lb \leq t(j) - t(i) \leq ub \)
  - Is a Simple Temporal Problem
  - “Temporal Constraint Networks”, Dechter, Meiri and Pearl, AIJ, 1991

• Good news – is polynomial
  - Bad news – in planning, we need to solve it a lot....
Simple Temporal Networks

- Can map STPs to an equivalent digraph:
  - One vertex per time-point (and one for 'time zero');
  - For $lb \leq t(j) - t(i) \leq ub$:
    - An edge $(i \rightarrow j)$ with weight $ub$.
    - An edge $(j \rightarrow i)$, with weight $-lb$
      - (c.f. $lb \leq t(j) - t(i) \rightarrow t(j) - t(i) \leq -lb$)
STN Example
STN Example

A

B

A

B
STN Example
STN Example
STN Example

\[ 0.00: (A) \quad [3] \]
\[ 0.01: (B) \quad [5] \]
Simple Temporal Networks (ii)

- Solve the shortest path problem (e.g. using Bellman-Ford) from/to zero
  - dist(0,j)=x → maximum timestamp of j = x
  - dist(j,0)=y → minimum timestamp of j = -y

- If we find a **negative cycle** then the temporal constraints are inconsistent:

![Diagram](image)
(Coles, Fox, Long and Smith, AAAI 2008);

(See also Halsey, Fox and Long, ECAI 2004)
Other fiddly details

• The closed list is a headache;

• Classical planning:
  − Discard states that are the same (in terms of facts, or same/worse cost) as states already seen.

• Temporal planning:
  − **Facts don't tell us everything** – due to the temporal constraints, the plan steps matter too.
  − ...as does their order – plans with different **permutations** of actions are interestingly different

## IPC 2004: Results

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>D+TL</th>
<th>D+NV</th>
<th>D+TL+NV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>302</td>
<td>116</td>
<td>272</td>
<td>136</td>
</tr>
<tr>
<td>CRIKEY</td>
<td>47</td>
<td>66</td>
<td>—</td>
<td>98</td>
</tr>
<tr>
<td>FAP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>FD</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>FDD</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>LPG-3</td>
<td>45</td>
<td>62</td>
<td>—</td>
<td>56</td>
</tr>
<tr>
<td>LPG-TD</td>
<td>76</td>
<td>62</td>
<td>63</td>
<td>100</td>
</tr>
<tr>
<td>Macro-FF</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Marvin</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Optop</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>P-MEP</td>
<td>24</td>
<td>45</td>
<td>24</td>
<td>43</td>
</tr>
<tr>
<td>Roadmapper</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>SGPlan</td>
<td>75</td>
<td>90</td>
<td>78</td>
<td>74</td>
</tr>
<tr>
<td>Tilsapa</td>
<td>—</td>
<td>10</td>
<td>69</td>
<td>62</td>
</tr>
<tr>
<td>YAHSP</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Right: % of instances attempted, left % of these solved

D: Durative Actions
NV: Numeric Variables
TL: Timed Initial Literals

**Note:** Change of rules, temporal track now separate. LPG3: last year’s winner.
Metric used: scalability (problems solved)

We will focus on generic techniques

A tuned planner

if domain name begins with “PS” and part after first letter is “SR”:
    use algorithm 100
else if there are 5 actions, all with 3 args, and 12 non-ground predicates:
    use algorithm −1000
else if all predicates ground and 10th/11th domain name letters “PA”:
    use algorithm −1004
else if there are 11 actions and action name lengths range from 5 to 28:
    use algorithm 107
...

IPC 2008 results slides
PDDL 2.2: Timed Initial Literals

- Introduced in PDDL 2.2 (IPC 2004);
- Allow us to model facts that become true, or false, at a specific time.
- Can use them to model deadlines or time windows.
- Cannot be done directly, but we can achieve this by adding more facts to the domain.
Modelling Deadlines using TILs

- Make sure the action achieving the desired fact has a condition that ensures it takes place before the deadline (over all or at start/end).
- Make that fact true in the initial state.
- And a TIL to delete it at the deadline.
- Note that we could have multiple deadlines for different objects.

```
(:durative-action unload-truck
 :parameters (?p - obj ?t- truck ?l- location)
 :duration (= ?duration 2)
 :condition (and (over all (at ?t ?l))
 (at start (in ?p ?t)))
 (at end (can-deliver ?p)))
 :effect (and (at start (not (in ?p ?t)))
 (at end (at ?p ?l))))

Init:
(can-deliver package1)
(at 9 (not (can-deliver package1)))
(can-deliver package2)
(at 11 (not (can-deliver package2)))
```
Modelling Time Windows Using TILs

- Make sure the action achieving the desired fact has a condition that ensures it takes place during the window (over all or at start/end). POPF/OPTIC will generally work better if you use over all where possible.

- Have a TIL to add that fact at the starting point for the window.

- And one to delete it when the window ends.

- Note that we could have multiple windows for the same fact by adding further TILs to the initial state.

```
(:durative-action bus-route
   ?from ?to – loc)
 :duration (= ?duration (route-duration ?r))
 :condition (and (at start (route ?r ?from ?to))
   (at start (at ?d ?from))
   (at start (at ?b ?from))
   (over all (working ?d))
   (at end (due ?r)))
 :effect (and (at start (not (at ?d ?from)))
   (at start (not (at ?b ?from)))
   (at end (at ?d ?to))
   (at end (at ?b ?to))
   (at end (done ?r))
)
init:
(at 3.75 (due route2))
(at 4 (not (due route2)))
```
Reasoning with TILs

- **TIL Sapa**
  - Before search starts add all TILs to the event queue at the time they must occur.

- **CRIKEY! (3)**
  - Consider TILs as actions that can be applied in search, check temporal consistency as applied.

- **LPG**
  - Local search approach: when a TIL precondition is not satisfied either:
    - Remove the action;
    - Delay the action until after the TIL is true;
    - Remove earlier actions so that the action can occur sooner.
q is an invariant condition of all ‘real’ actions in the domain, $g_n$ becomes a goal.

Introduces required concurrency, making temporally interesting domains;

Cannot be handled by planners using action compression (although the original TIL models can).

Compilation makes problems much harder to solve.
Focus on Metrics measuring Plan Quality, not just coverage/speed: tracks again merged together (no separate temporal track), overall satisfying track winner SGPlan.

<table>
<thead>
<tr>
<th>Plan</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPPlan</td>
<td>STRIPS</td>
</tr>
<tr>
<td>Fast Downward</td>
<td>STRIPS</td>
</tr>
<tr>
<td>Hplan-P</td>
<td>STRIPS, Simple Preferences, Qualitative Preferences</td>
</tr>
<tr>
<td>MIPS-XXL</td>
<td>STRIPS, Simple Preferences, Qualitative Preferences, Time</td>
</tr>
<tr>
<td>Yochan-PS</td>
<td>STRIPS, Simple Preferences, Time</td>
</tr>
<tr>
<td>SGPlan</td>
<td>STRIPS, Simple Preferences, Qualitative Preferences, Time</td>
</tr>
</tbody>
</table>

First (makespan) Optimal Temporal Planner in Competition: Winner CPT (Vidal & Tabary) works by compilation to constraint programming. No other competitors, subsequent years cancelled due to only having one participant.

Temporal Preferences introduced, handled by MIPS-XXL (and SGPlan). Preference tracks also did not run after 2006.

No required concurrency.
‘Baseline’ performed best – throw time away, run a classical planner. No temporally interesting domains, so this worked very well.

SGPlan 6 was the best competitor – also ignored time

TFD – Decision Epoch Planner

DAE – decomposed by learning a goal agenda

CPT – optimal temporal planning using CP

TLP-GP – temporally expressive planner, based on regression in planning graphs
IPC 2011

• Return of some temporally interesting domains:
  - TMS (required concurrency bake during fire kiln)
  - Turn and Open (turn handle and open door)
  - Match Cellar (mend fuse whilst match is lit).
Winner: DAE, now with YAHSP – a forward-search planner with lookahead. Not temporally expressive, so no problems solved in matchcellar, turn-and-open and TMS.

Joint runners up: YAHSP without DAE; and POPF – the only competitive planner to solve temporally expressive problems

LMTD: prototype landmark heuristic with TFD

Sharaabi: extension of SAPA to increase temporal expressivity
10 domains, incl. 3 temporally interesting ones (from 2011).

5 Participants:
- ITSAT: SAT-Based Temporally Expressive Planner.
- tBURTON: Uses sub-goals and calls a sub-planner (TFD). Temporally Expressive if sub-planner is.
- Temporal Fast Downward.
- YAHSP3 and YAHSP3-MT (MT = multi-threaded)
- DAE-YAHSP.

<table>
<thead>
<tr>
<th></th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>YAHSP3-MT</td>
<td>86.5/200</td>
<td>1st</td>
</tr>
<tr>
<td>Temporal-FD</td>
<td>79.2/200</td>
<td>2nd</td>
</tr>
<tr>
<td>YAHSP3</td>
<td>66.6/200</td>
<td>3rd</td>
</tr>
</tbody>
</table>

First portfolios in the temporal track: TemPorAl and CP4TP. The former did not use a temporally expressive planner; the latter did (ITSAT), so could solve problems in the ‘Cushing’ domain.

TFLAP – forward partial-order planner, with landmark and relaxed-plan heuristics. Competitive with CP4TP – a portfolio!

PopCorn – a planner for domains with control parameters (not tested in the competition)

<table>
<thead>
<tr>
<th>Domain/Planner</th>
<th>PopCorn</th>
<th>TemPorAl</th>
<th>TFLAP</th>
<th>CP4TP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road-traffic</td>
<td>0</td>
<td>8.04</td>
<td>0</td>
<td>6.67</td>
</tr>
<tr>
<td>trucks</td>
<td>4.86</td>
<td><strong>8.99</strong></td>
<td>7.32</td>
<td>4.5</td>
</tr>
<tr>
<td>Map-analyzer</td>
<td>0</td>
<td><strong>9.05</strong></td>
<td>6.29</td>
<td>5.12</td>
</tr>
<tr>
<td>sokoban</td>
<td>0.76</td>
<td>5.28</td>
<td>4</td>
<td>4.05</td>
</tr>
<tr>
<td>airport</td>
<td>3</td>
<td><strong>9.98</strong></td>
<td>8.99</td>
<td>8.9</td>
</tr>
<tr>
<td>parking</td>
<td>4</td>
<td>6.17</td>
<td>3.65</td>
<td>2.85</td>
</tr>
<tr>
<td>quantum</td>
<td>4.38</td>
<td><strong>8.74</strong></td>
<td>7</td>
<td>7.82</td>
</tr>
<tr>
<td>cushing</td>
<td>1</td>
<td>0</td>
<td>2.98</td>
<td>2.22</td>
</tr>
<tr>
<td>floortile</td>
<td>0</td>
<td><strong>9.55</strong></td>
<td>2.9</td>
<td>1.59</td>
</tr>
<tr>
<td>Total</td>
<td>18</td>
<td><strong>65.8</strong></td>
<td>43.13</td>
<td>43.72</td>
</tr>
</tbody>
</table>
Recent Work/Challenges in Temporal Planning

- Much work in temporal planning is outwith PDDL2.1, e.g. timeline-based approaches (Frank, Chen, Smith, Cesta, Oddi, Fratini, ....)
- Reasoning efficiently with more interesting temporal constraints:
  - Relaxation heuristics for time windows (Allard et al); MTP (To et al); FAPE (Bit Monnot & Smith); Temporal Landmarks (Marzal et al; Wang et al); effective memoisation and metastates (Coles et al)
- RoboCup Logistics League Competition (robocup.org/leagues/17)
- Plan execution, including with temporal uncertainty (Chen et al)
- Hybrid Planning (e.g. PDDL+), interaction of time and numbers:
  - UPMurphi (Della Penna et al), DiNo (Piotrowski et al), ENHSP (Scala et al), PluReal (Bryce), OPTIC+ (Coles²), SMTPlan+ (Cashmore et al), Kongming (Li & Williams).