The Temporal Track of the International Planning Competition

Amanda Coles and Andrew Coles

KING'S College LONDON

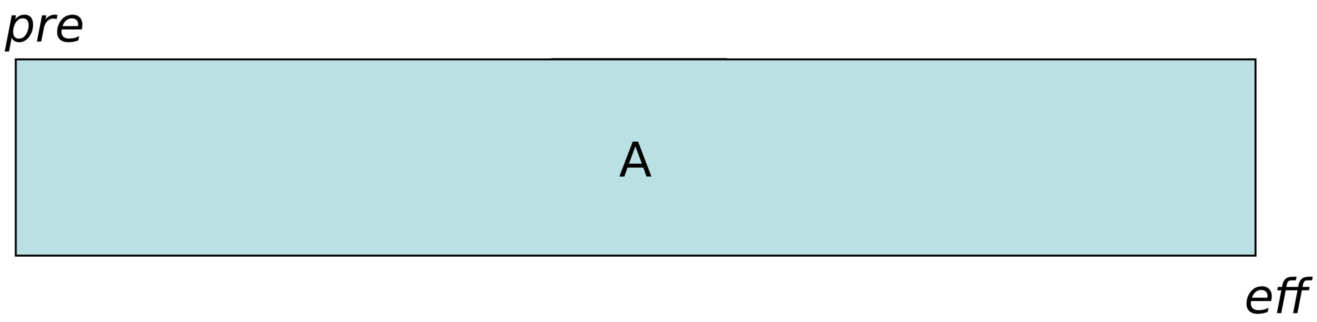
King's College London, UK

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.

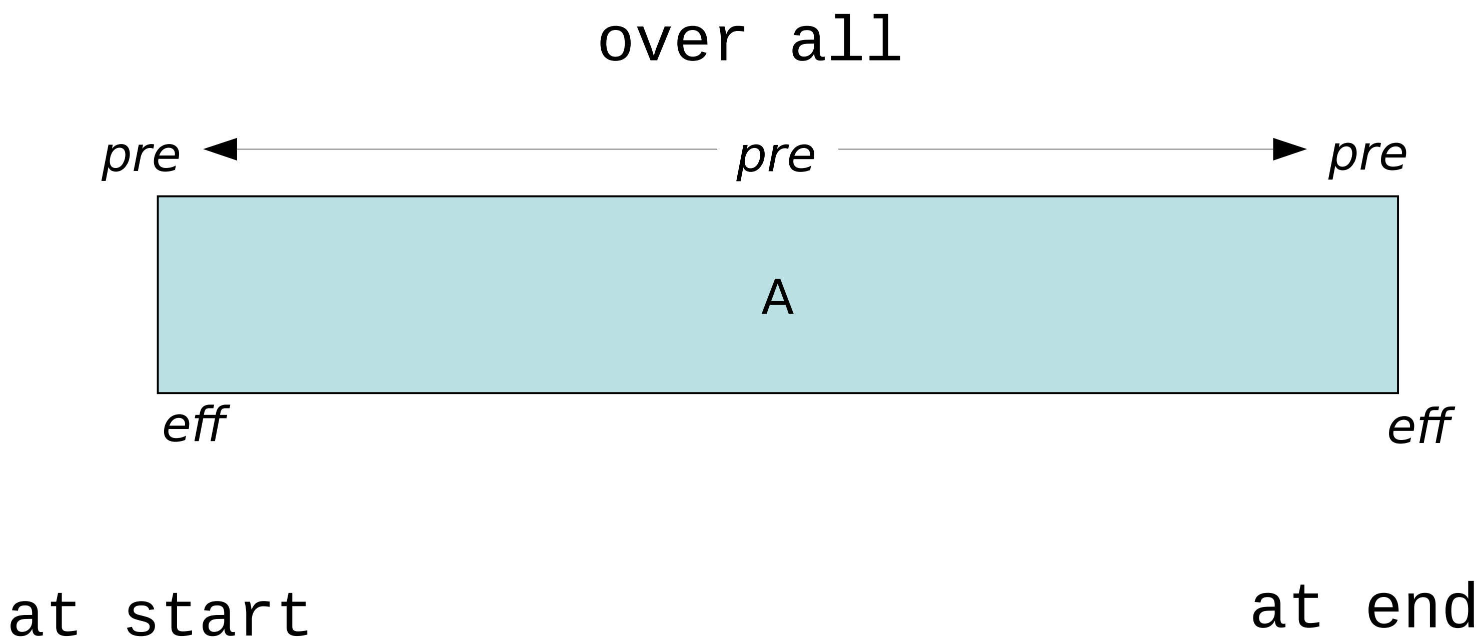
Temporal Graph Plan

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

"Temporal Planning with Mutual Exclusion Reasoning" D. Smith & D. Weld, IJCAI 1999.

Durative Actions in PDDL 2.1

First Temporal Track @ Third IPC: 2002



"PDDL2.1: an extension to PDDL for expressing temporal planning domains", Fox M. and Long D., JAIR Vol. 20, 2003.

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)))
```

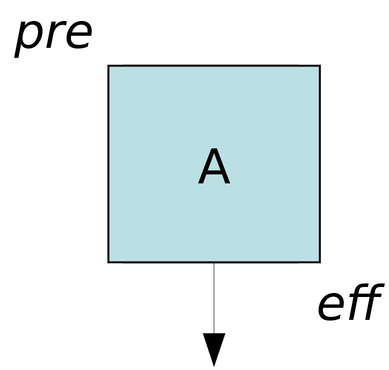
PDDL Example (i)

```
(: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)))
```

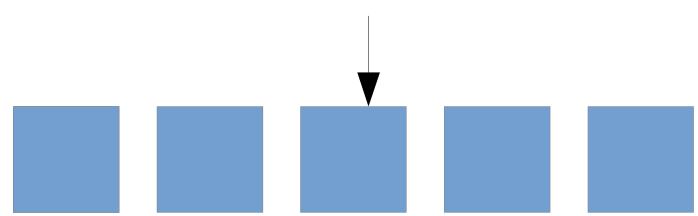
Beware of self-overlapping actions!

“Complexity of concurrent temporal planning“, Rintanen J., ICAPS 2007

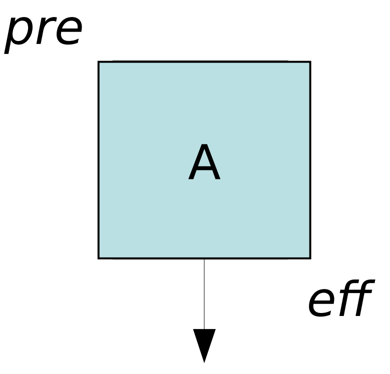
Durative Actions?



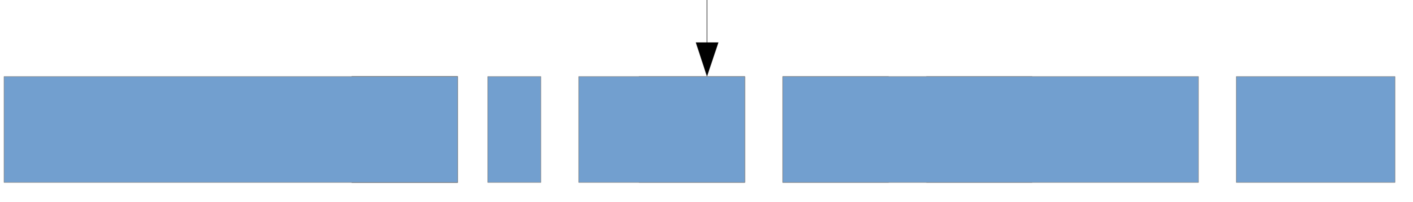
Classical Planner



Durative Actions?



Classical Planner



Temporal Planners in IPC 2003

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

Winner, Fully Automated: LPG, solved more problems because it also handled temporal domains.

PDDL Example (ii)

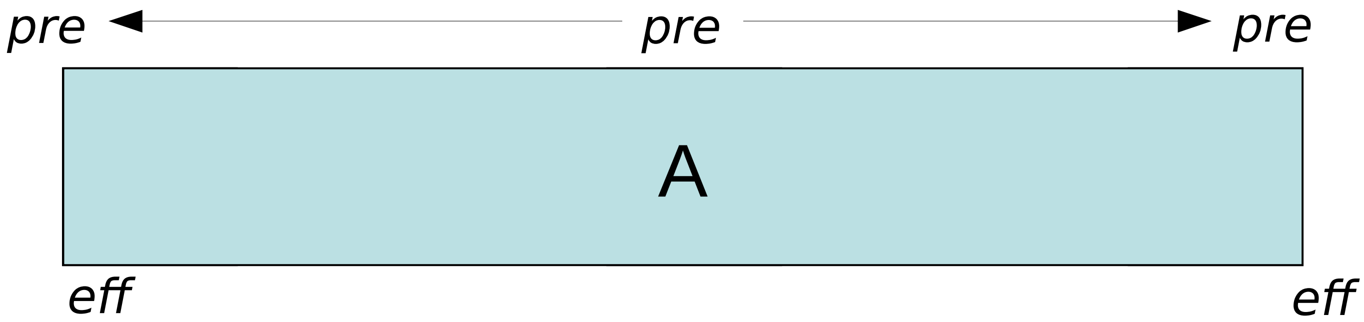
```
(:durative-action open-barrier
   :parameters
  (?loc – location ?p - person)
   :duration (= ?duration 1)
   :condition
    (and   (at start (at ?loc ?p)))
   :effect
    (and (at start (barrier-open ?loc))
         (at end (not (barrier-open ?loc))))
```

PDDL Example (ii)


```
(:durative-action op
  :parameters
 (?loc – location ?p
  :duration (= ?dura
  :condition
   (and   (at start
  :effect
   (and (at start (barrier-open ?loc))
        (at end (not (barrier-open ?loc))))
```

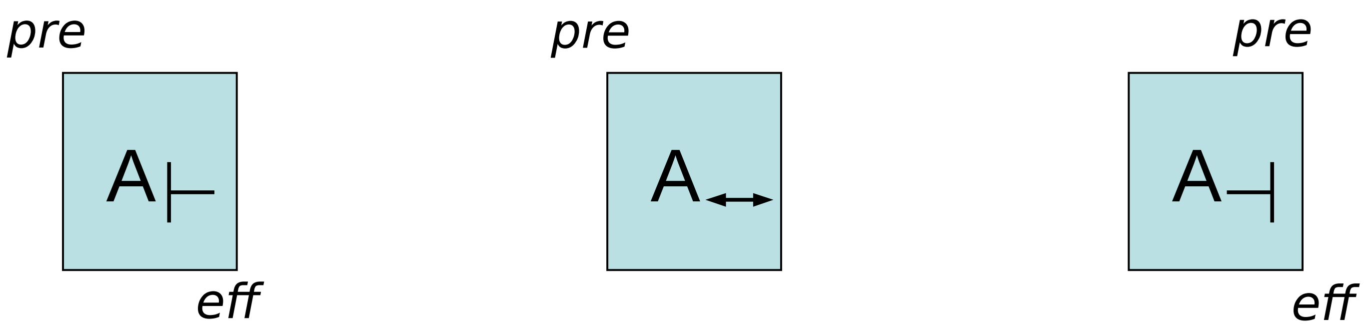
Durative Actions in LPGP

(Fox and Long, ICAPS 2003)



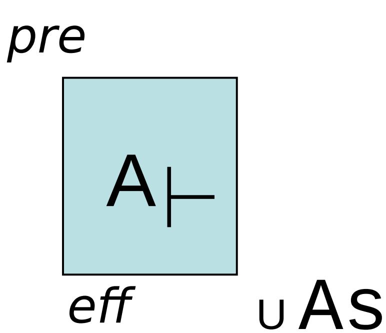
Durative Actions in LPGP

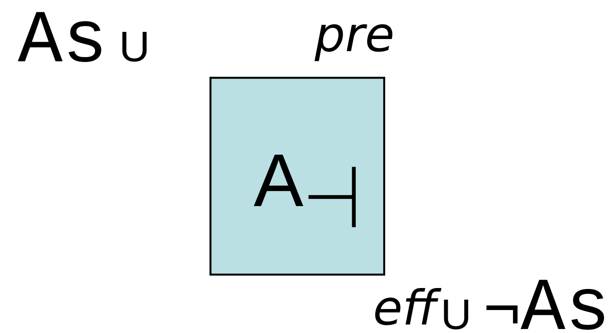
(Fox and Long, ICAPS 2003)



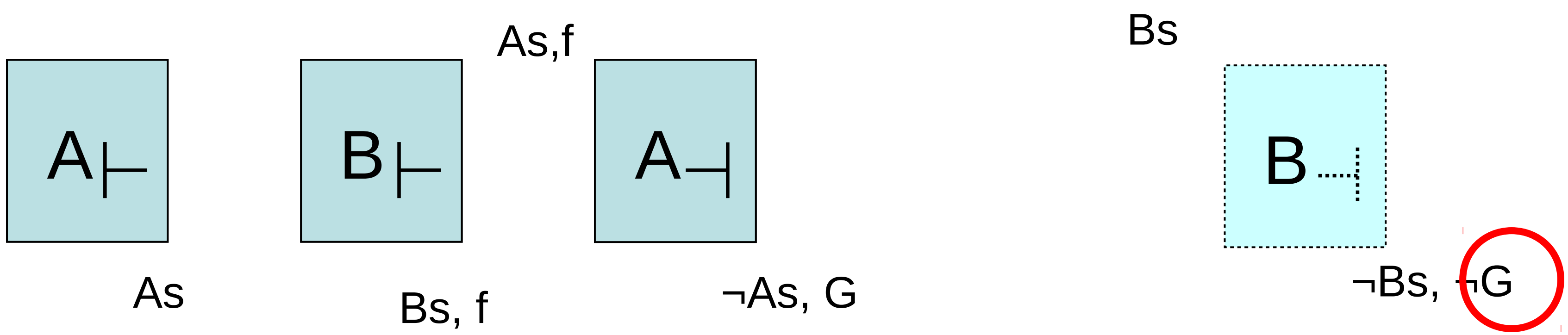
Durative Actions in LPGP

(Fox and Long, ICAPS 2003)





Planning with Snap Actions (i)

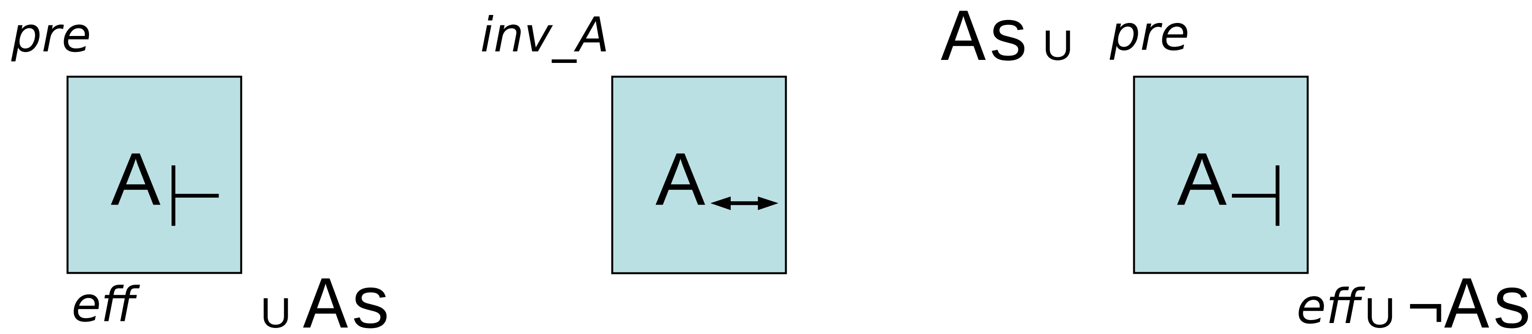


Challenge 1: What if $B_{\dashv}$ interferes with the goal?

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

- **Solution:** add ¬As, ¬Bs, ¬Cs.... 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: `(imply (As) inv_A)`
  - Violating an invariant then leads to a **dead-end**.

Planning with Snap Actions (iii)

$A_{\vdash}$ $B_{\vdash}$ $B_{\dashv}$ $A_{\dashv}$

- 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 $A_{\vdash}$ (at time t)
    - Insert $A_{\dashv}$ into Q at time $(t + dur(A))$
    - $S'.t = S.t + \varepsilon$
  - Remove and apply the first end snap-action from Q.
    - $S'.t$ set to the scheduled time of this, plus ε

"Using the Context-enhanced Additive Heuristic for Temporal and Numeric Planning." Eyerich P., Mattmüller R. and Röger G., ICAPS 2009
"Sapa: A Scalable Multi-Objective Metric Temporal Planner", Do M. and Kambhampati S., JAIR 2003.

Running through our example...

$A\vdash$ $B\vdash$

t=0 *t=0.01*

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

t=3 *t=5.01*

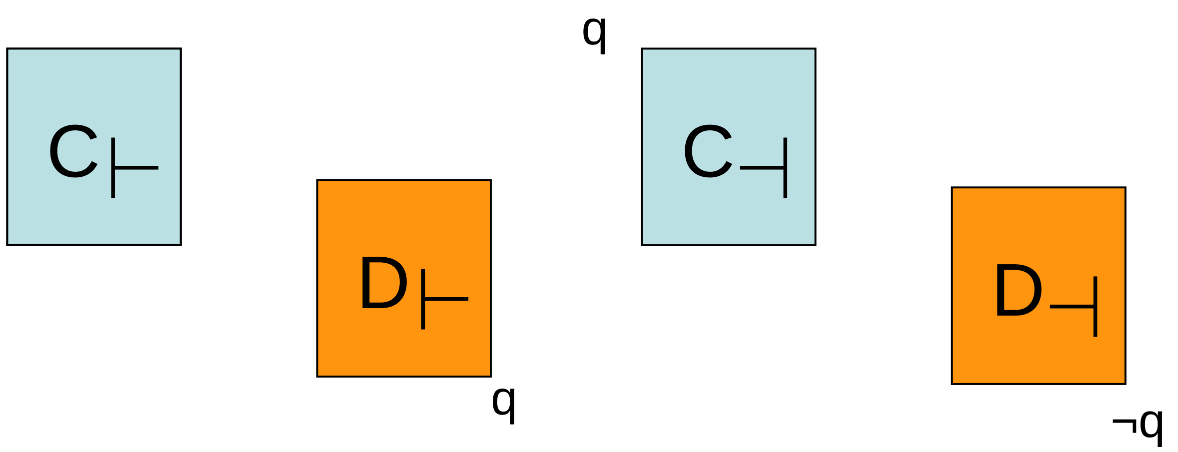
$A\dashv$ $B\dashv$

Q

What does this look like if we do Bstart first?

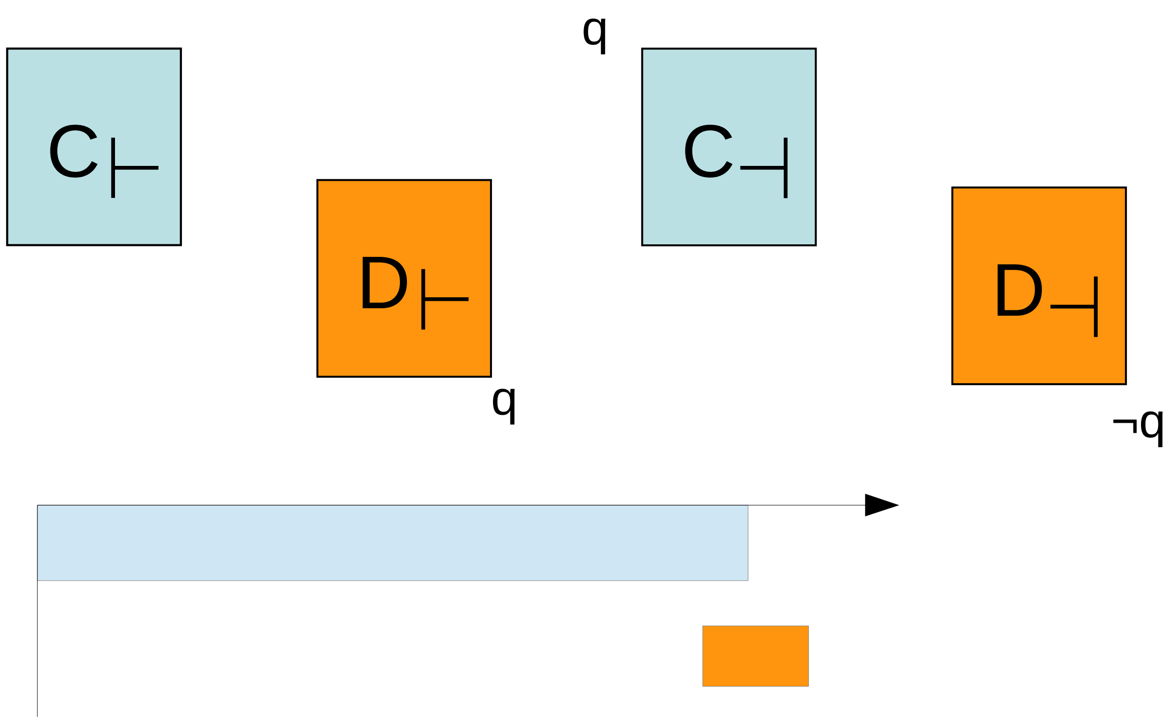
Decision Epoch Planning: The snag

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



Decision Epoch Planning: The snag

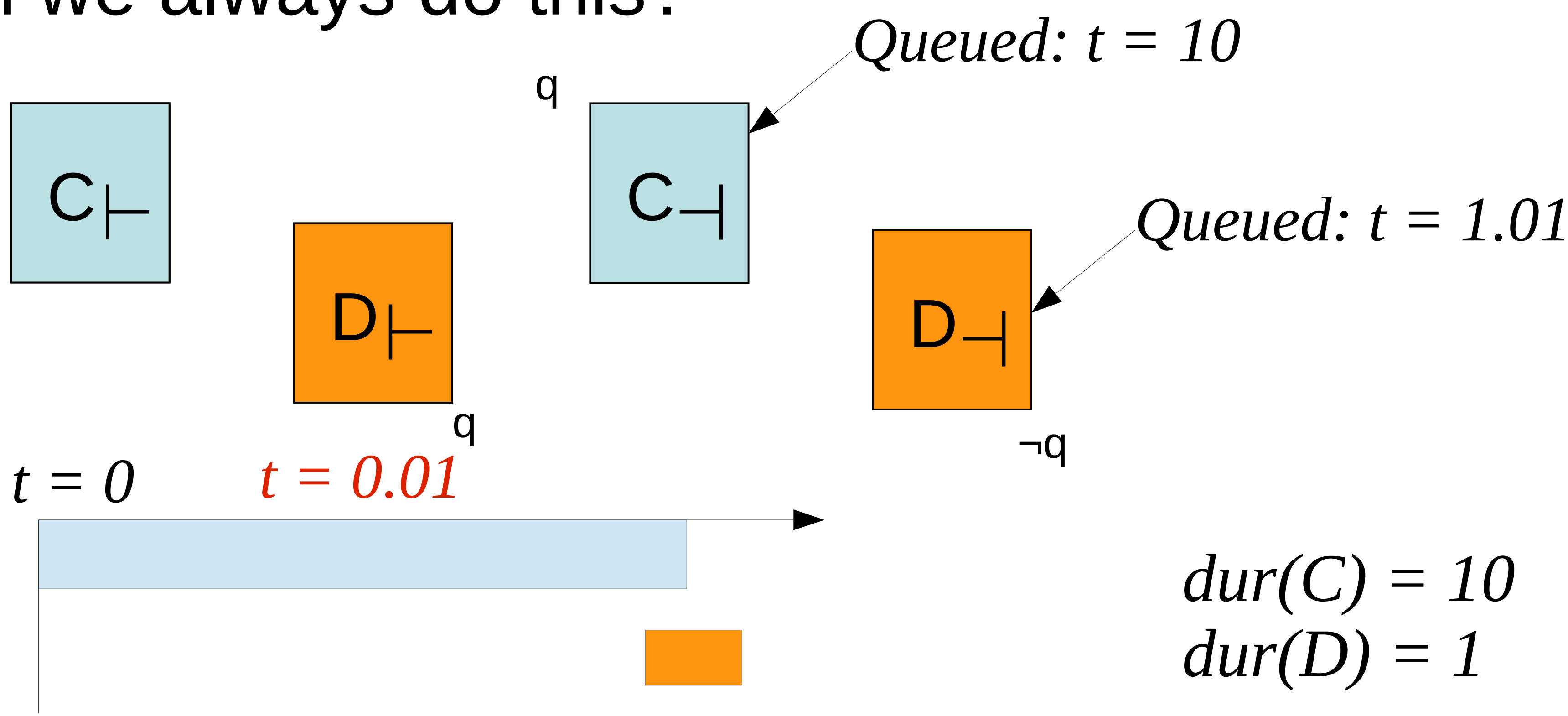
- Must **fix start- and end-timestamps** at the point when the action is started.
  - Used for the priority queue
- Can we always do this?



dur(C) = 10
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?



IPC 2004 Planners

	ADL	DP	Numbers	Durations	TL
CRIKEY	-	-	+	+	-
FAP	-	-	-	-	-
FD, FDD	+	+	-	-	-
LPG-TD	+	+	+	+	+
Macro-FF	+	-	-	-	-
Marvin	+	+	-	-	-
Optop	+	+	+	+	+
P-MEP	+	-	+	+	+
Roadmapper	-	-	-	-	-
SGPlan	+	+	+	+	+
Tilsapa	-	-	+	+	+
YAHSP	-	-	-	-	-

"The Deterministic Part of IPC-4: An Overview" Hoffmann J. and Edelkamp S., JAIR Special Issue on the 4th International Planning Competition 2005.

Simple Temporal Networks: VHPOP and CRIKEY!

“Temporal Constraint Networks”, Dechter, Meiri and Pearl, Artificial Intelligence, 1991
“VHPOP: Versatile heuristic partial order planner” Younes H. and Simmons R., JAIR Vol 20, 2003.
"Planning with Problems Requiring Temporal Coordination." A. I. Coles, M. Fox, D. Long, and A. J. Smith. AAAI 08.
"Managing concurrency in temporal planning using planner-scheduler interaction." A. I. Coles, M. Fox, K. Halsey, D. Long, and A. J. Smith. Artificial Intelligence, 173 (1), 2009.

Option 2: a Simple Temporal Problem

- All our constraints are of the form:
  - $\varepsilon \leq t(i+1) - t(i)$ *(c.f. sequence constraints)*
  - $dur_{min}(A) \leq t(A_{\dashv}) - t(A_{\vdash}) \leq dur_{max}(A)$
- 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....

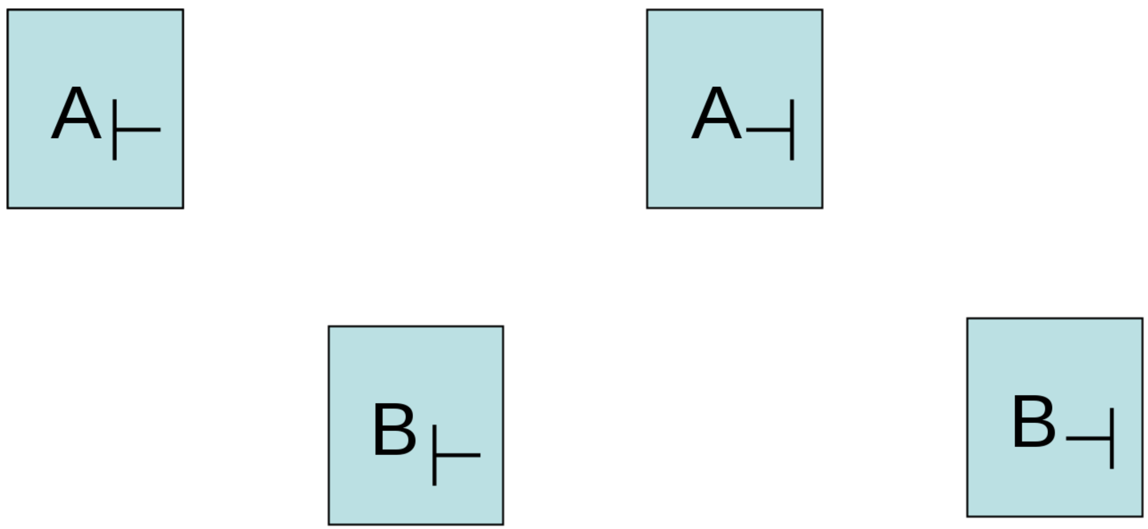
Simple Temporal Networks

- 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 → j) with weight *ub*.
    - An edge (j → i), with weight *-lb*
      - (c.f. $lb \leq t(j) - t(i) \quad \rightarrow \quad t(j) - t(i) \leq -lb$)

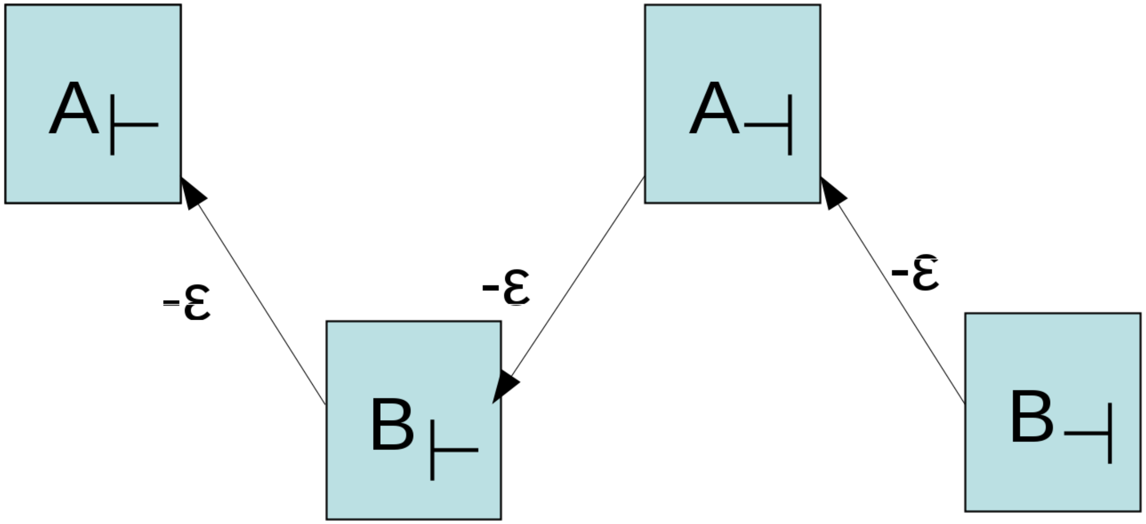
STN Example

$A\vdash$ $B\vdash$ $A\dashv$ $B\dashv$

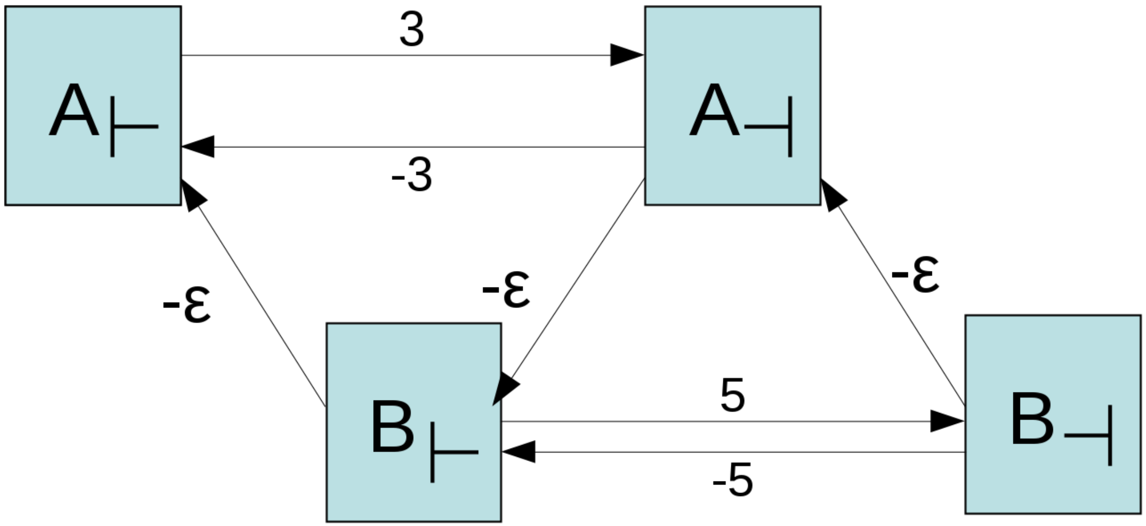
STN Example



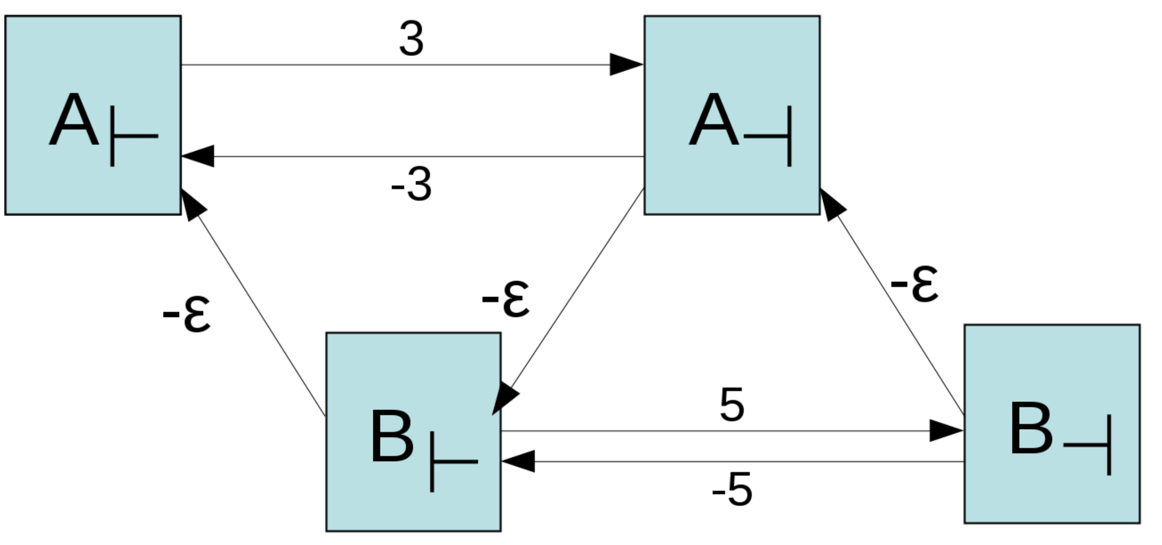
STN Example



STN Example



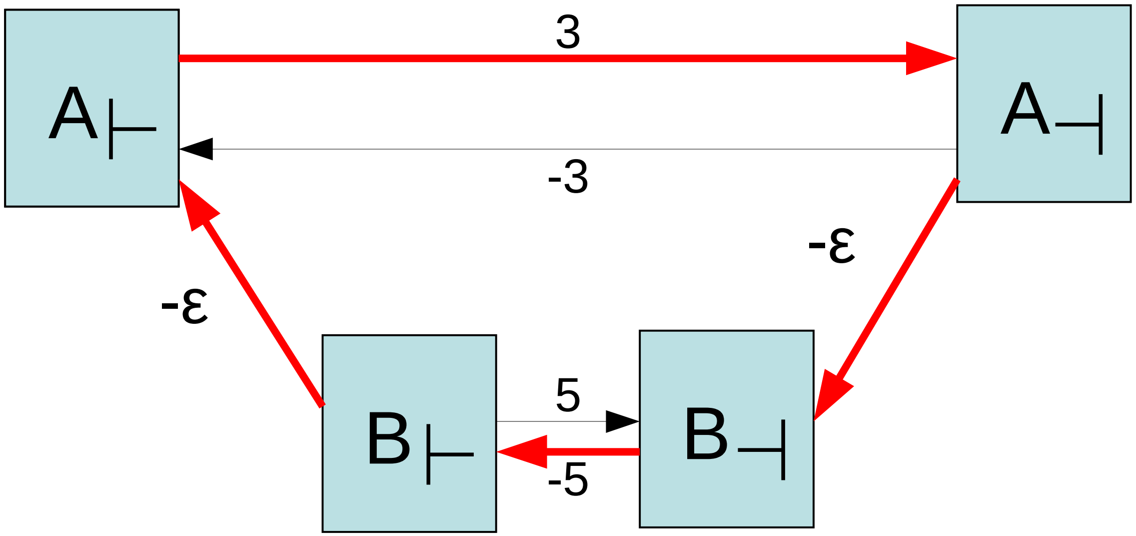
STN Example




```
0.00: (A) [3]
0.01: (B) [5]
```

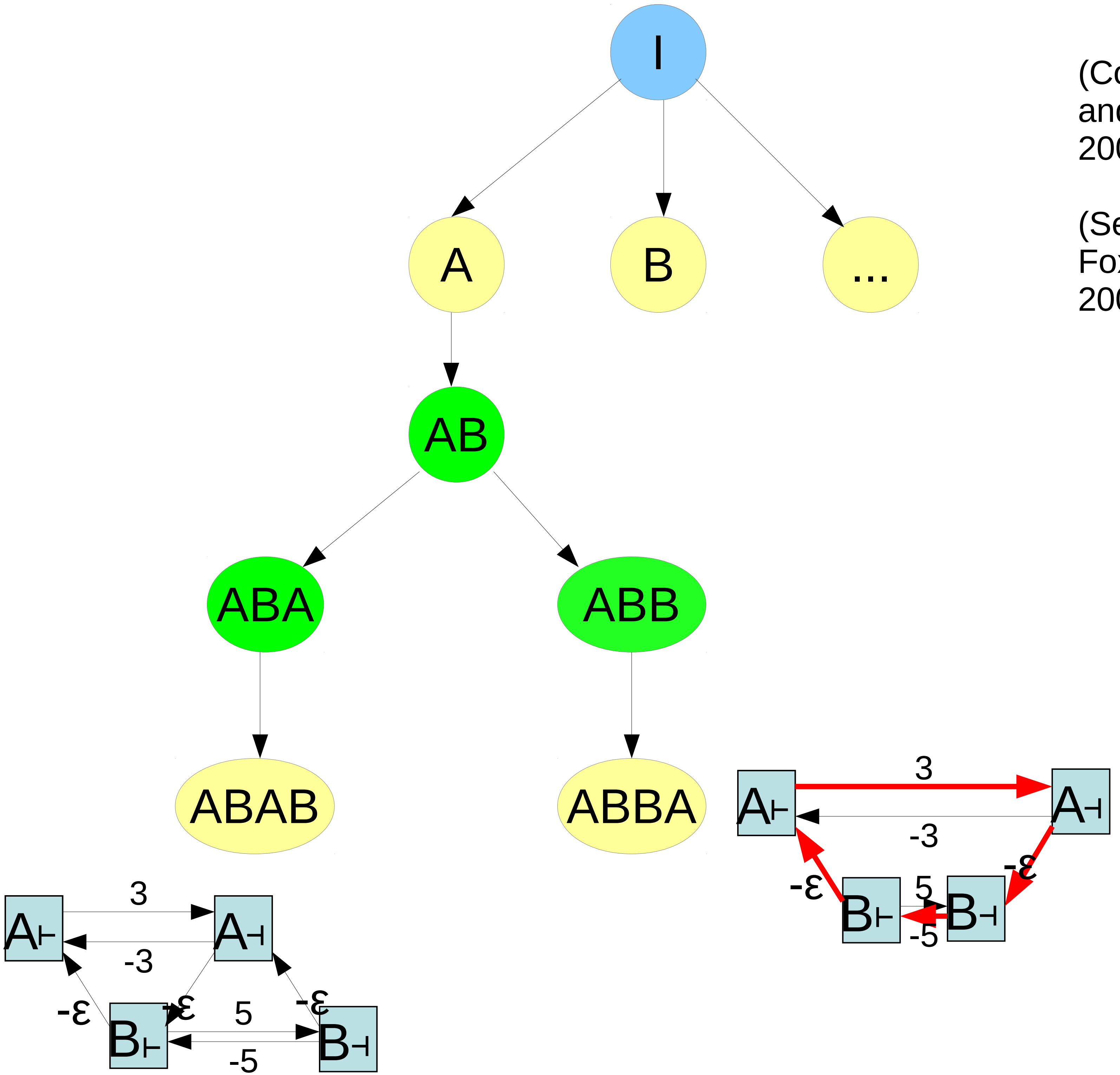
Simple Temporal Networks (ii)

- 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:



CRIKEY! (3)

"Planning with Problems Requiring Temporal Coordination." A. I. Coles, M. Fox, D. Long, and A. J. Smith. AAAI 08.
"Managing concurrency in temporal planning using planner-scheduler interaction." A. I. Coles, M. Fox, K. Halsey, D. Long, and A. J. Smith. Artificial Intelligence. 173 (1). 2009.



(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

"Have I Been Here Before? State Memoization in Temporal Planning", A.J. Coles & A.I. Coles ICAPS 2016.
Paper: https://www.aaai.org/ocs/index.php/ICAPS/ICAPS16/paper/view/13187
Talk Video: https://youtu.be/AwL1A25tjYo?list=PLj-ZdQ5rfSEpnsOfJeG7UfheAuZ42tEOM&t=928

IPC 2004: Results

	D		D+TL		D+NV		D+TL+NV	
Number	302		116		272		136	
CRIKEY	47	66	—		98	55	—	
FAP	—		—		—		—	
FD	—		—		—		—	
FDD	—		—		—		—	
LPG-3	45	62	—		56	50	—	
LPG-TD	76	62	63	100	96	50	87	100
Macro-FF	—		—		—		—	
Marvin	—		—		—		—	
Optop			8	43	—		—	
P-MEP	24	45	24	43	13	32		
Roadmapper	—		—		—		—	
SGPlan	75	90	78	74	85	100	74	100
Tilsapa			10	69			62	63
YAHSP	—		—		—		—	

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)

"The Deterministic Part of IPC-4: An Overview" Hoffmann J. and Edelkamp S., JAIR Special Issue on the 4th International Planning Competition 2005.

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
...

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

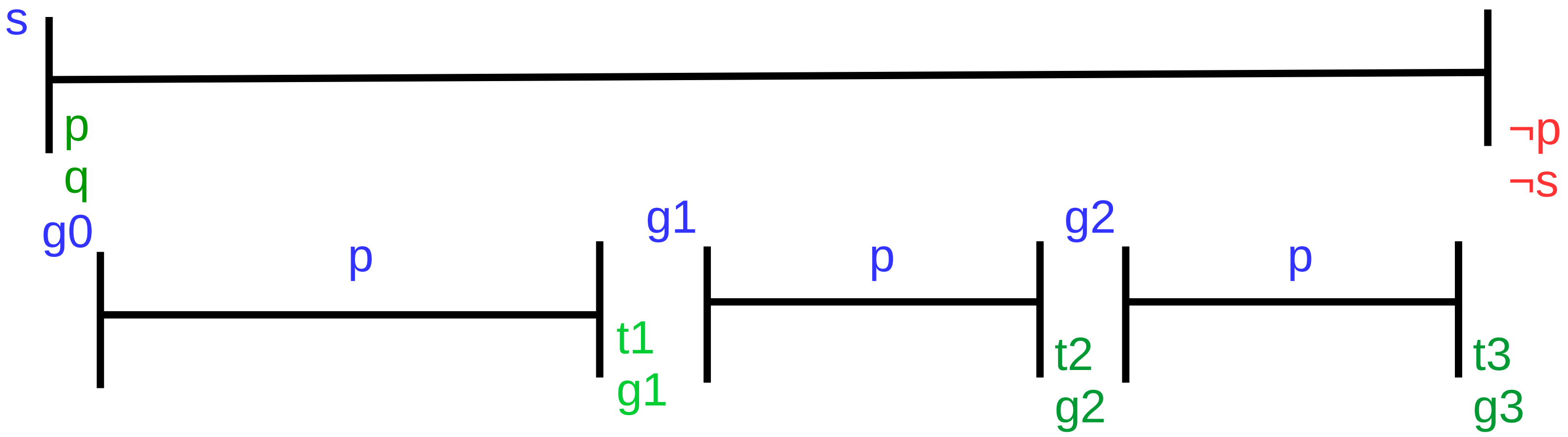
- Make sure the action achieving the desired fact has a condition that ensures it takes place during the window (over all or at start/end). POPF/OPTIC will generally work better if you use over all where possible.
- Have a TIL to add that fact at the starting point for the window.
- And one to delete 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
  :parameters (?d – driver ?r – route ?b – bus
                    ?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.

Compiled TIL Domains
Pipes, Airport, Satellite, UMTS



- q is an invariant condition of all ‘real’ actions in the domain, gn 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.

IPC 2006

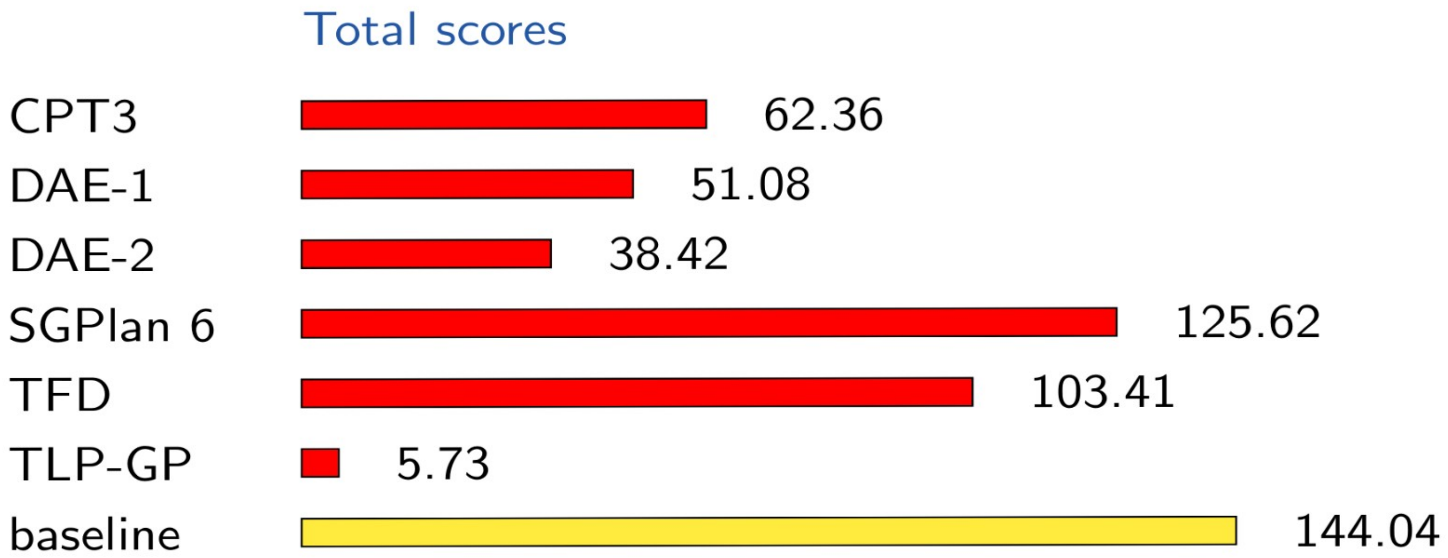
Gerevini, Dimopolous, Haslum and Saetti

- Focus on Metrics measuring Plan Quality, not just coverage/speed: tracks again merged together (no separate temporal track), overall satisfycing track winner SGPlan.

IPPlan	STRIPS
Fast Downward	STRIPS
Hplan-P	STRIPS, Simple Preferences, Qualitative Preferences
MIPS-XXL	STRIPS, Simple Preferences, Qualitative Preferences, Time
Yochan-PS	STRIPS, Simple Preferences, Time
SGPlan	STRIPS, Simple Preferences, Qualitative Preferences, Time

- 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.

IPC 2008

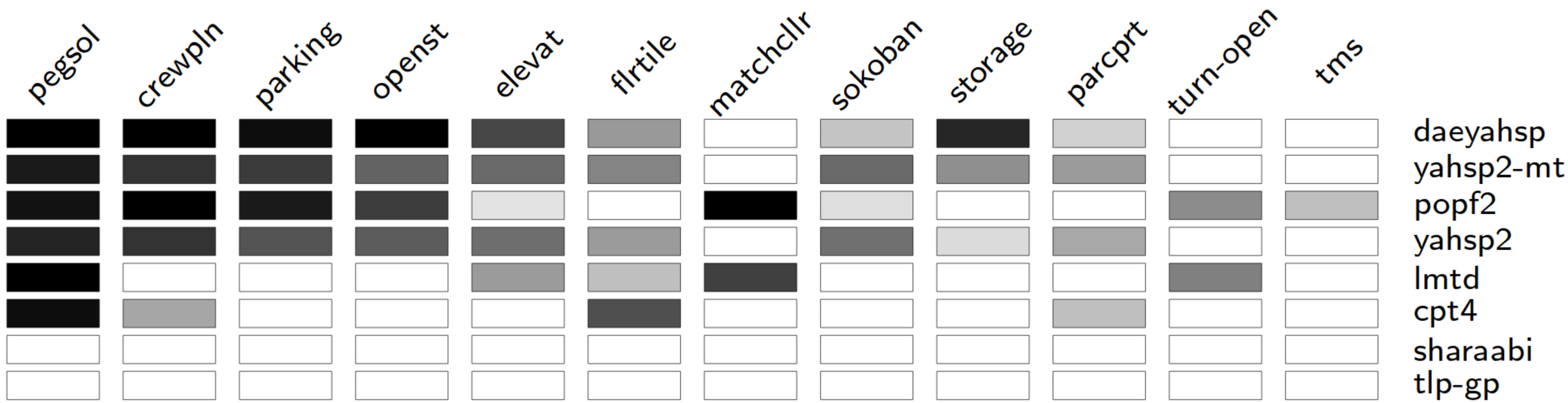


- 'Baseline' performed best – throw time away, run a classical planner. No temporally interesting domains, so this worked very well.
- SGPlan 6 was the best competitor – also ignored time
- TFD – Decision Epoch Planner
- DAE – decomposed by learning a goal agenda
- CPT – optimal temporal planning using 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).

IPC 2011



- 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

IPC 2014

- 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.

YAHSP3-MT	86.5/200	**1st**
Temporal-FD	79.2/200	**2nd**
YAHSP3	66.6/200	3rd

“The 2014 International Planning Competition: Progress and Trends” Vallati, M. and Chrpa, L. and Grzes, M. and McCluskey, T.L. and Roberts, M. and Sanner, S. AI Magazine, 2015.

IPC 2018

Domain/Planner	PopCorn	TemPorAl	TFLAP	CP4TP
Road-traffic	0	8.04	0	6.67
trucks	4.86	**8.99**	7.32	4.5
Map-analyzer	0	**9.05**	6.29	5.12
sokoban	0.76	**5.28**	4	4.05
airport	3	**9.98**	8.99	8.9
parking	4	6.17	3.65	2.85
quantum	4.38	**8.74**	7	7.82
cushing	1	0	2.98	2.22
floortile	0	**9.55**	2.9	1.59
Total	18	**65.8**	43.13	43.72

- 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)

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+ (Coles^2), SMTPlan+ (Cashmore et al), Kongming (Li & Williams).
- Applications work: Retirement Home Assistance, Space, Liner Shipping, Aerial Surveillance, Mining.