AI Planning

Mathijs de Weerdt

Cees Witteveen

04/01/2007

Faculty of Electrical Engineering, Mathematics and Computer Science,
Department of Software Technology

CWI, Amsterdam, Sen-4, Han la Poutré

Delft University of Technology

Aim of AI

- “Intelligent” systems, decide themselves
  - what to do, and
  - how to do it.
- Planning is…
  - given what to do (goals),
  - determine how (and when) to do it (plan).
- Currently planning research community very active
  - Significant scale-up
  - Bi-annual planning competition

Brief tutorial on AI Planning

(there was a whole summer school on AI Planning in June)

- Some basic background on AI Planning
- Relation with multi-agent planning

Contents

- Introduction
  - Applications
  - Problem
- Classical planning
  - Model
  - STRIPS
- Refinement planning framework
  - Partial plans
  - Plan space refinement
  - HTN planning
- Complexity of planning

Applications

- Action choice + resource handling
  - for transportation of goods
  - at schools, hospitals
  - Hubble Space Telescope scheduler
- Interactive decision making
  - for military operations
  - for astronomic observations
  - Plan-based interfaces (plan recognition)




Planning problem

- How to get from the current state to your goal state?
- Planning involves
  - Action selection
  - Action sequencing
  - Resource handling
- Plans can be
  - Action sequences
  - Policies/strategies (action trees)

Additional complexities

Because the world is …

- Dynamic
- Stochastic
- Partially observable

And because actions

- take time
- have continuous effects

AI Planning background

- Focus on classical planning; assume none of the above
- Deterministic, static, fully observable
  - “Basic”
  - Most of the recent progress
  - Ideas often also useful for more complex problems

Classical planning model

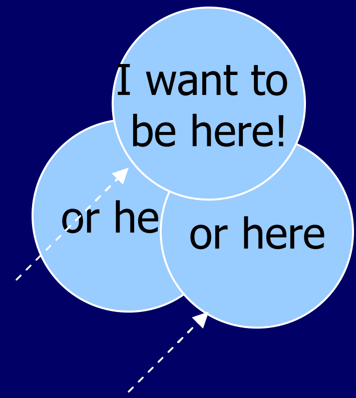
- Origins:
  - STanford Research Institute Problem Solver ('71)
  - derived from GPS = human problem solving ('61)
- States described by propositions currently true
- Actions: general state transformations described by sets of pre- and post-conditions
- Represents a state-transition system (but more compact)

Planning is searching

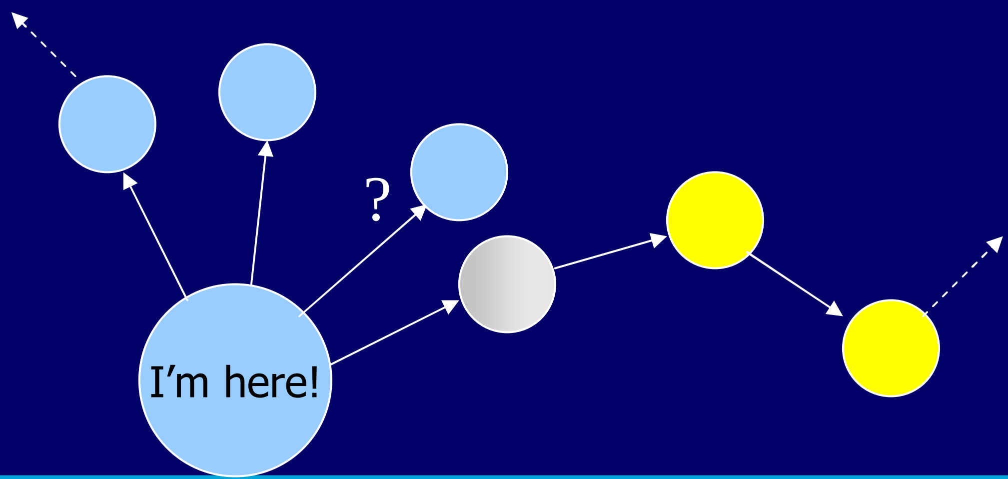
…in state space (or…)

Choose between possible actions

- Depth-first
- Breadth-first



goal states

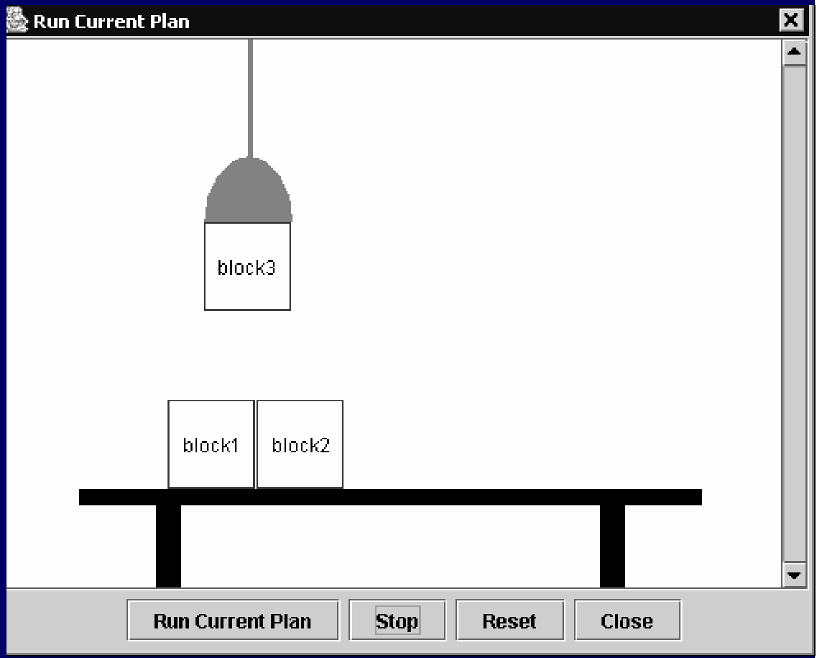


a plan is a route in state space

initial state

STRIPS Formalism (now in PDDL)

- action: preconditions, add, delete effects
- pickup(B1, B2)
  - precondition: empty & clear(B1) & on(B1, B2)
  - add-effect: holding(B1), clear(B2)
  - delete-effect: empty, on(B1, B2), clear(B1)
- problem: initial state, actions/operators, goal description
- objects and variables



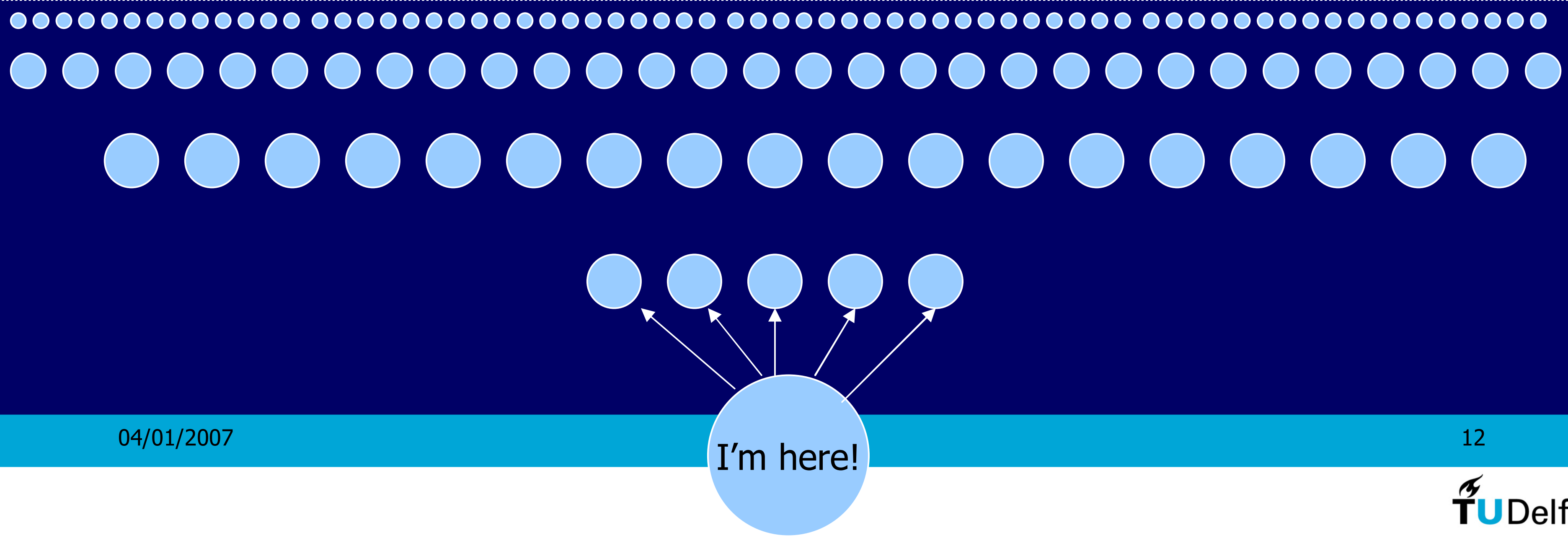
TUDelft

Searching for a plan

start from initial state, try all possible actions

→ large search space

STRIPS: regression: look at goal state!

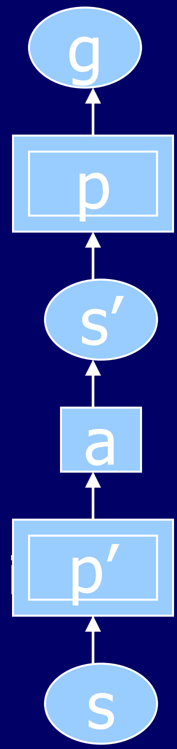


STRIPS algorithm

STRIPS(*s*, *g*)

returns: a sequence of actions that transforms *s* into *g*

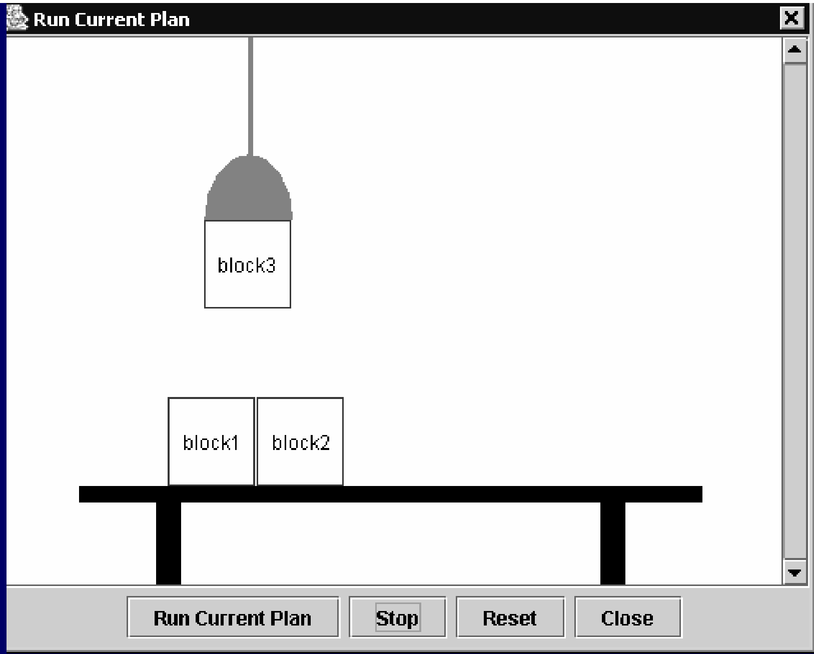
1. Calculate the difference set *d=g-s*.
    1. If *d* is empty, return an empty plan
2. Choose action *a* whose add-list has most formulas contained
3. *p′* = STRIPS(*s*, precondition of *a*)
4. Compute the new state *s′* by applying *p′* and *a* to *s*.
5. *p* = STRIPS(*s′*, *g*)
6. return *p′;a;p*



TUDelft

STRIPS demo

(by CI – space)



Action	Preconditions	Add List	Delete List
pickup(B1, B2)	empty & clear(B1) & on(B1, B2)	holding(B1), clear(B2)	empty, on(B1, B2), clear(B1)
pickuptable(B)	empty & clear(B) & ontable(B)	holding(B)	empty, ontable(B), clear(B)
putdown(B1, B2)	holding(B1) & clear(B2)	empty, on(B1, B2), clear(B1)	clear(B2), holding(B1)
putdowntable(B)	holding(B)	empty, ontable(B), clear(B)	holding(B)

Classical planning in a multiagent setting

- When an agent is unable to do a task
  - Give task to other
  - Other: adapt current plan $\rightarrow$ plan repair

Van der Krogt (2005)

Refinement planning framework

- Framework to capture all planning algorithms
- Idea: Narrow set *P* of potential action sequences
- Subcontents:
  - Specify action sequences by partial plans
  - Refinement strategies
  - Generic template

Refinement planning template

Refineplan(*P* : Plan set)

1. If *P* is empty, Fail.
2. If a minimal candidate of *P* is a solution, return it. End
3. Select a refinement strategy *R*
4. Apply *R* to *P* to get a new plan set *P′*
5. Call Refineplan(*P′*)

Termination ensured if *R* complete and monotonic.

TUDelft

Existing Refinement Strategies

- State space refinement: e.g. STRIPS
- Plan space refinement: e.g. Least commitment planning
- Task refinement: e.g. HTN

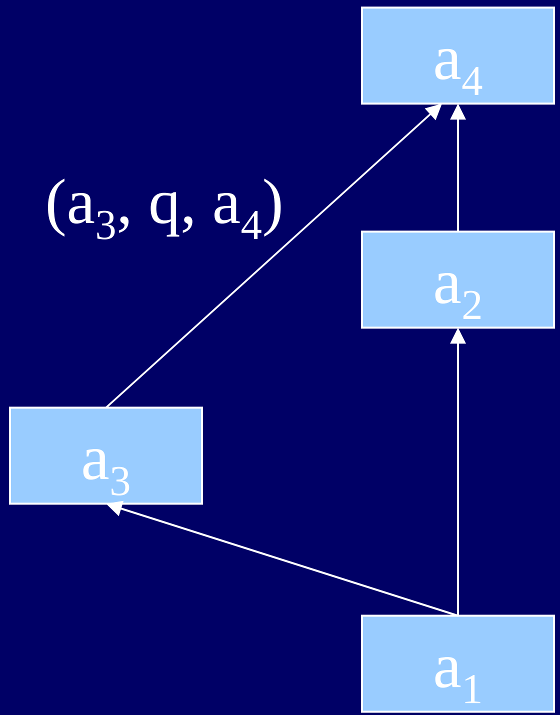
Plan space refinement (I)

- Least commitment planning (Weld, 94)
  - search in plan space instead of state space
  - represent plans more flexible: not a sequence, but a partially ordered set
  - keep track of decisions and the reasons for these decisions

Partial Plans: Syntax

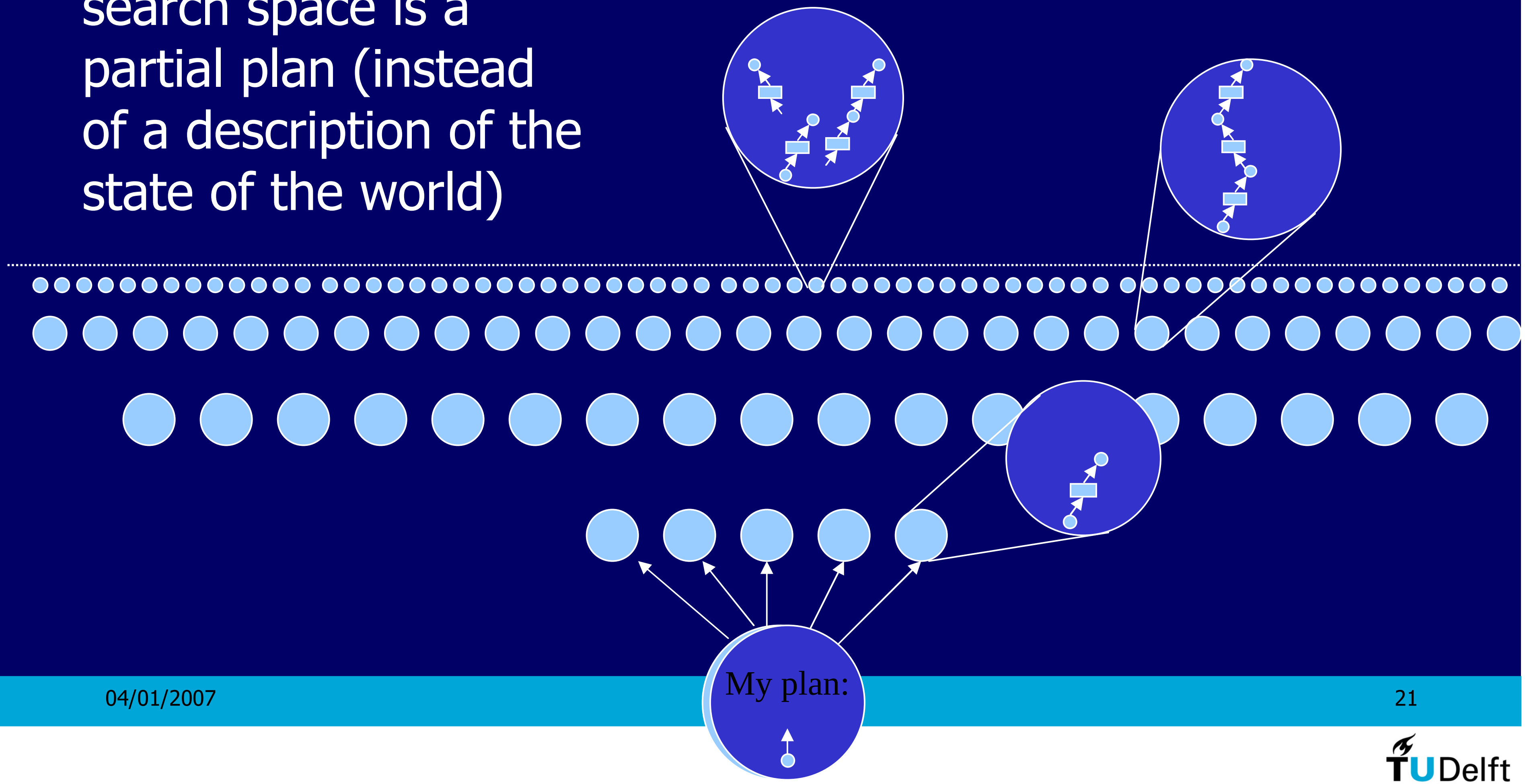
Partial plan = (Actions, partial Ordering, causal Links)

- causal links = Interval preservation constraint (IPC) (a_1 , p , a_2)
  - p must be preserved between a_1 and a_2



Plan space

A state in the (plan) search space is a partial plan (instead of a description of the state of the world)



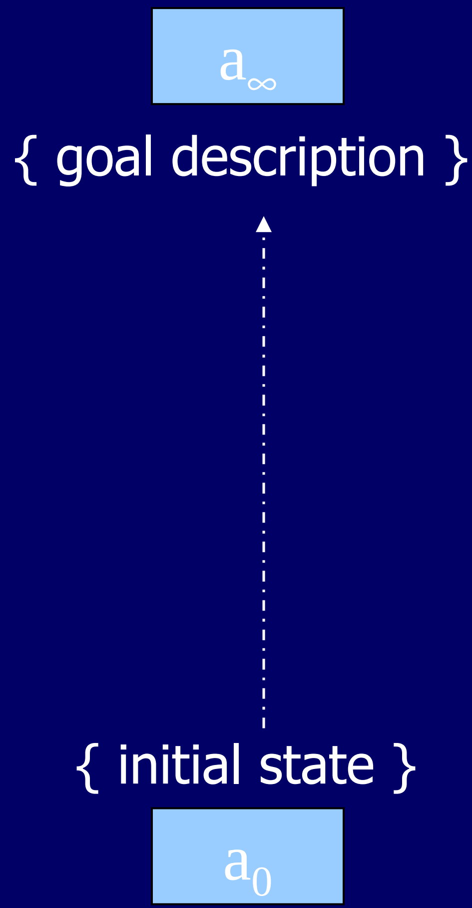
TUDelft

POP

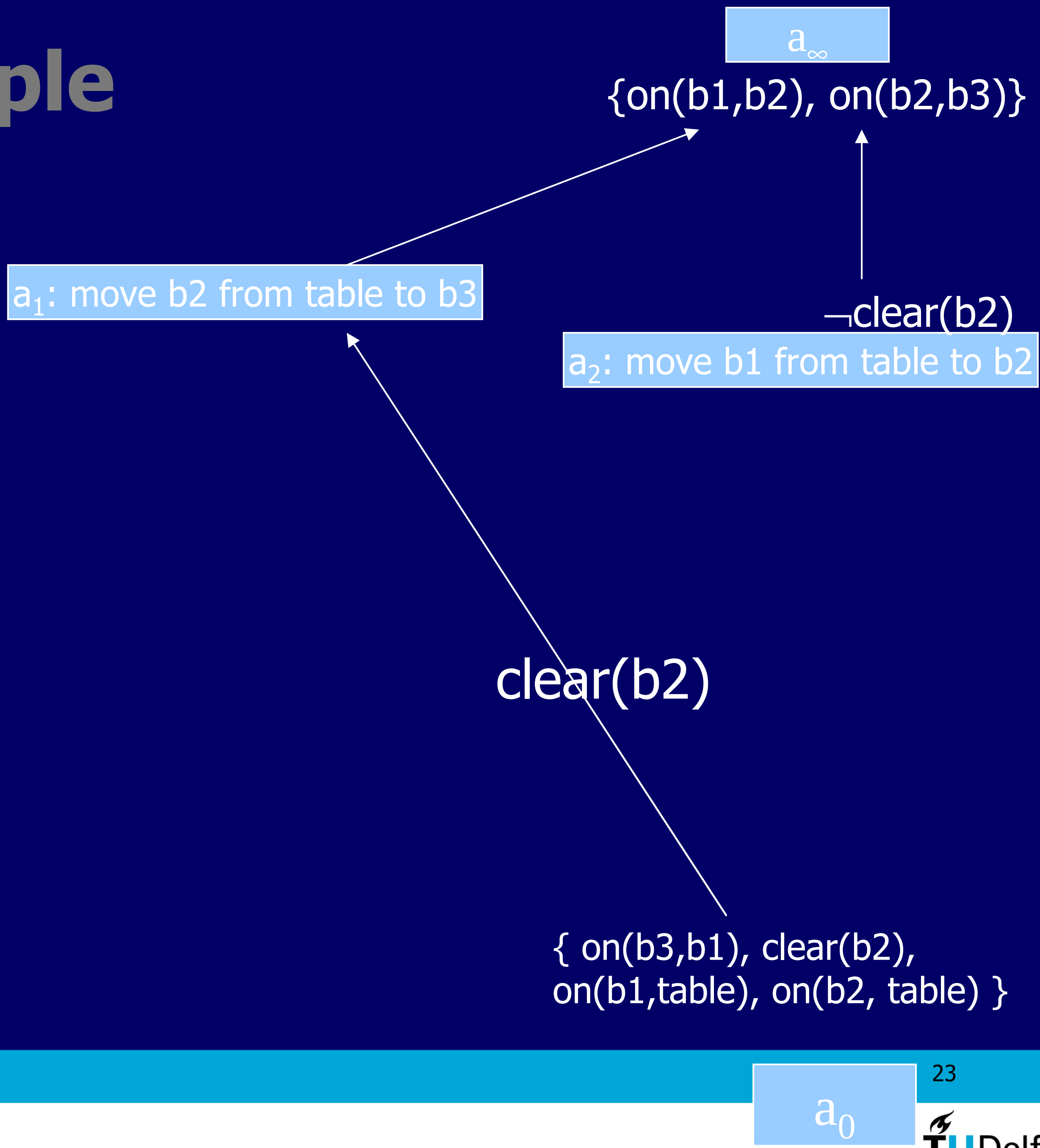
- Start with
  - null (empty) plan
  - agenda

    = list of (precondition, action) goals

    = $\{(g_1, a_\infty), (g_2, a_\infty), (g_3, a_\infty), \ldots\}$
- Deal with one (g,a) at a time



POP example



a∞
{on(b1,b2), on(b2,b3)}
a1: move b2 from table to b3
¬clear(b2)
a2: move b1 from table to b2
clear(b2)
{ on(b3,b1), clear(b2),
on(b1,table), on(b2, table) }
a0

POP((A,O,L), agenda,)

1. **Termination:** if agenda is empty return (A,O,L)
2. **Goal selection:** select a (g,a_{need}) from the agenda
3. **Action selection:** choose an action a_{add} that adds g. Update L, O, and A
4. **Update goal set:** remove (g,a_{need}) from agenda, and add its preconditions
5. **Causal link protection:** for every action a_t that might threaten a link, add an ordering constraint

Tradeoffs among Refinements

State space refinement:

- commit to both order and relevance of actions
- include state information (easier plan validation)
- lead to premature commitment
- too many states when actions have durations

Plan-space refinement:

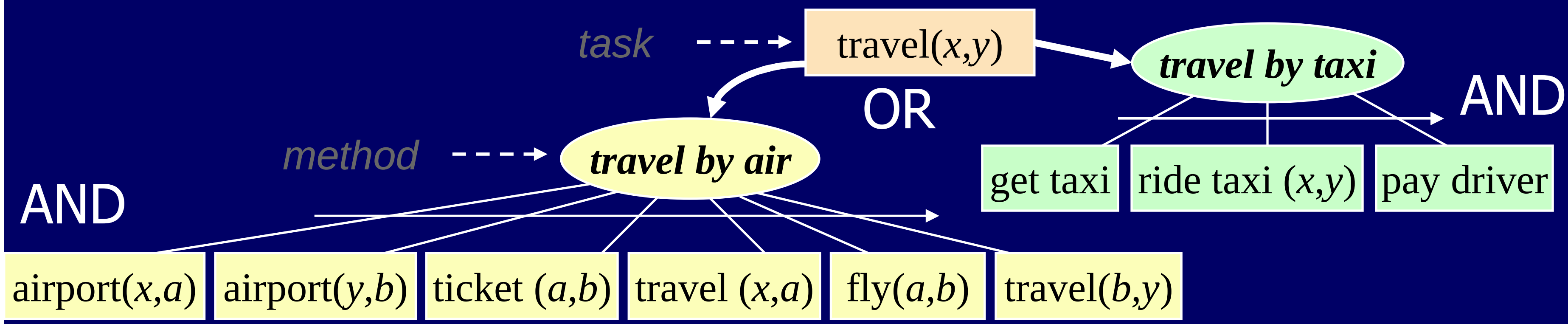
- commit to actions, avoid constraining order
- increase plan-validation costs
- reduce commitment (large candidate set /branch)
- easily extendible to actions with duration

Partial plans in a multiagent setting

- Broadcast (abstraction of) part of your plans relevant for others → "partial global plan"
- Keep updating this global plan until nothing changes

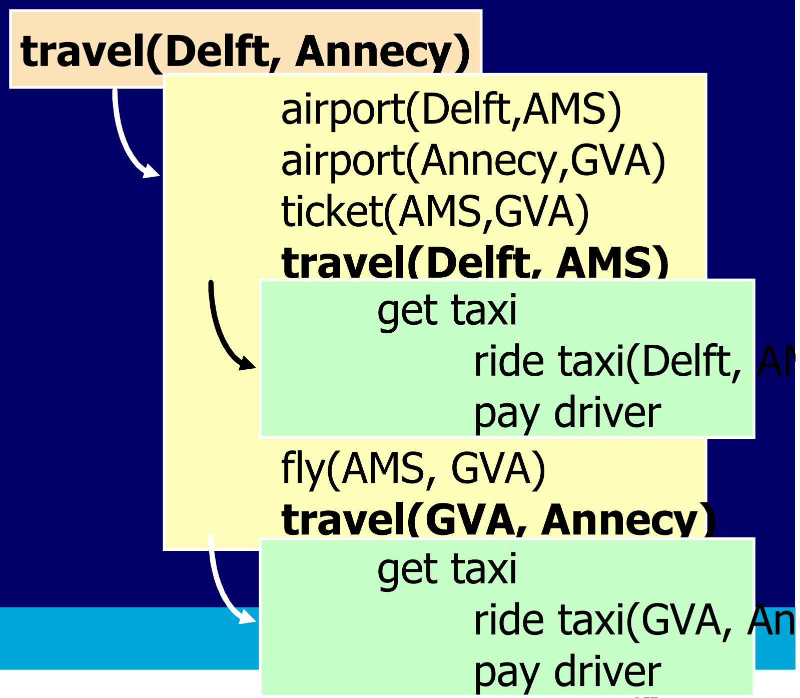
Generalized Partial Global Planning (Durfee, Decker, Lesser, et al.)

HTN Planning



Problem reduction

- Decompose tasks into subtasks
- Handle constraints (e.g., taxi not good for long distances)
- Resolve interactions (e.g., take taxi early enough to catch plane)
- If necessary, backtrack and try other decompositions



Hierarchical multiagent planning

- Task structure for all involved agents
  - QAFs
    - SUM (~ AND), SyncSUM
    - MAX (~ OR), MIN
  - Non-local effects
    - Enables, Disables
    - Facilitates, Hinders
- TAEMS (Decker), c_TAEMS (Boddy et al.)
- Zlot and Stentz (2006)

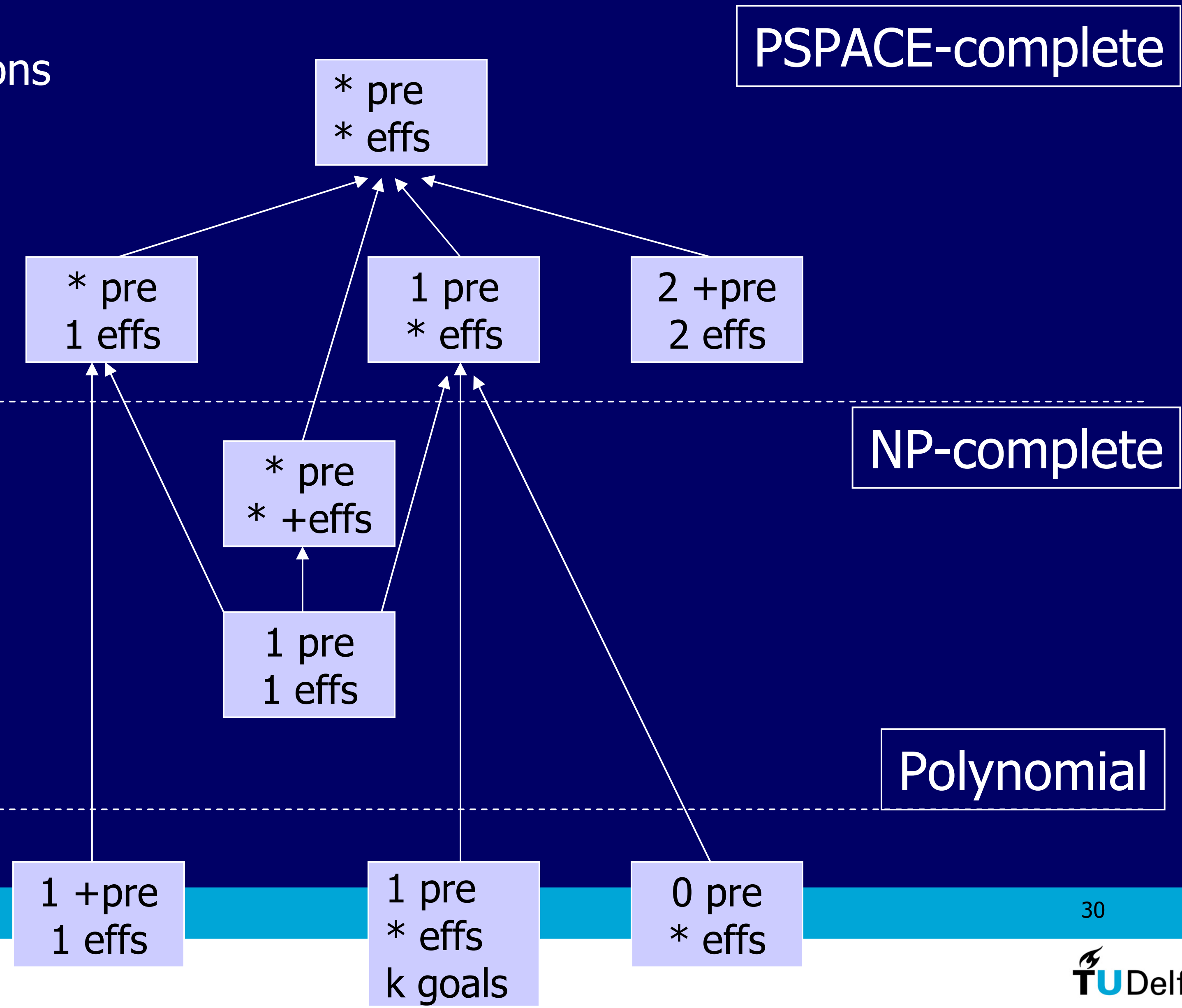
Complexity of planning

- PSPACE-completeness (NP$\subseteq$PSPACE)
- PLANEX: plan existence problem is PSPACE-complete for propositional STRIPS
- Restrictions on preconditions allowed and on effects, helps to reduce complexity

Complexity of planning

pre: preconditions

effs: effects



Future work

Apply other planning techniques to multiagent planning:

- Non-deterministic actions (surprises)
- Partially observable world (exact costs unknown)
- Durative actions (move, travel) and continuous variables (capacity, fuel, distance, time) $\rightarrow$ optimality

Recommended reading

- Kambhampati, S. (1997). Refinement planning as a unifying framework for plan synthesis. *AI Magazine*, 18(2):67-97.
- Ghallab, M., Nau, D., Traverso, P. (2004). *Automated Planning: Theory and Practice*, Elsevier.
- Vlahavas I., Vrakas, D. (2005). *Intelligent Techniques for Planning*, Idea Group.

Other references

- Boddy, M., Horling, B., Phelps, J., Goldman, R. Vincent, C. Long and B. Kohout, C_TAEMS Language Specification, Version 1.06.
- Decker, K. S. and Lesser, V. R. (1992). Generalizing the partial global planning algorithm. *International Journal of Intelligent and Cooperative Information Systems*, 1(2):319-346.
- Decker, K. (1996). TAEMS: A Framework for Environment Centered Analysis & Design of Coordination Mechanisms. *Foundations of Distributed Artificial Intelligence*, Chapter 16, G. O'Hare and N. Jennings (eds.), Wiley Inter-Science, pp. 429-448.
- Fikes, R. E. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189-208.
- Newell, A. and Simon, H. (1963). Gps: A program that simulates human thought. In Feigenbaum, E. and Feldman, J., editors, *Computers and Thought*, pages 279-296.
- Van der Krogt, R. and de Weerdt, M. (2005). Coordination through Plan Repair. In Gelbukh and de Albornoz and Terashima-Marin (Eds.). *MICAI 2005: Advances in Artificial Intelligence*, pages 264-274. LNAI 3789.
- Weld, D. S. (1994). An introduction to least-commitment planning. *AI Magazine*, 15(4):27-61.
- Zlot, R. M. and Stentz, A. (2006). Market-based multirobot coordination for complex tasks. *International Journal of Robotics Research*, Special Issue on the 4th International Conference on Field and Service Robotics, 25(1):73-101.