# BACK TO THE FUTURE of PLANNING

Subbarao Kambhampati
Arizona State University

# Acknowledgements

[Home Page](#)
[Lecturers & Course Materials](#)
[Call for Participation](#)
[Application Form](#)
[Registration Form](#)
[School Location](#)
[School Programme](#)
[List of the participants](#)
[PLANET](#)
[Sponsors](#)

PLANET

# International Summer School on AI Planning

September 28 - October 1, 2000
Coral Beach Hotel, Cyprus

## A Unifying and Brand-Name-Free Introduction to Planning

Participants included
- Carmel Domshlak
- Jorg Hoffmann
- Julie Porteous
- Malte Helmert
- Michael Brenner

planning a realistically usable tool for complex problem-solving.

The school is aimed at PhD students and young academic researchers.

### Lecturers & Topics of the School

Dr. Hector Geffner (University Simón Bolívar, Venezuela)
Heuristic Search Planning: Models, Heuristics, and Algorithms

Dr. Malik Ghallab (LAAS-CNRS, France)
Planning with time and resources

Prof. Subbarao Kambhampati (Arizona State University, USA)
A Unifying and Brand-Name-Free Introduction to Planning

Dr. Derek Long (University of Durham, UK)
Pre-processing and Domain Analysis

Prof. Dana Nau (University of Maryland, USA)
Ordered Task Decomposition: Theory and Applications

Prof. Bernhard Nebel (University of Freiburg, Germany)
Computational Complexity of Planning and Expressiveness of I

Dr. Paolo Traverso (IRST-ITC, Italy)
Planning as model checking

### Location of the School

The Summer School will be held in the beautiful seaside environment of t
http://www.coral.com.cy

Although planning is one of the oldest research areas of AI, recent years have brought many dramatic advances in both its theory and practice. On the theory side, we now understand the deep connections between AI planning, heuristic search, constraint satisfaction, logic and operations research. On the practical side, we have effective ways of capturing and using domain-specific control knowledge, and have planners that are capable of synthesizing plans with hundred or more actions in minutes. These are undoubtedly exciting times for planning research. For newcomers to the field, however, all this excitement does present special problems of trying to figure out foundational ideas scattered among a welter of brand-name algorithms.

The aim of my lecture(s) will be to provide a comprehensive overview of the field, placing both the traditional ideas and the recent advances in a unified perspective. I will isolate and present a brandname-free collection of foundational ideas underlying the old and new crops of planning algorithms. I will then discuss how these can be mixed and matched to develop planning algorithms offering a broad spectrum of tradeoffs.

While my initial emphasis will be on planning algorithms for deterministic domains, I will also briefly discuss the extensions of the essential ideas to domains with metric and temporal constraints, partially observable states as well as stochastic dynamics. The lectures should be accessible to anyone with basic computer science and AI background.

**Preliminary material** for the course will be available at URL
rakaposhi.eas.asu.edu/planning-tutorial.

CSE 571: Artificial Intelligence (Fall 10)

http://rakaposhi.eas.asu.edu/cse571/

Apple Yahoo! Google Maps YouTube Wikipedia News (54) Popular

Course Overview

Lecture Notes

Course Wiki

Mail Archive

AI@ASU

CSE 571(F09) Notes

CSE 571(F09) Wiki

CSE 571(F09) Mail

CSE 471

Related Courses

Rao Kambhampati

Unable to read strip image!

# Lecture Notes for CSE571 (F10)

http://rakaposhi.eas.asu.edu/cse571

Lecture notes:

1. Introduction
   - L1 Audio of [Aug 23, 2010] ( NEW NEW Video of the lecture video (4gb) Introduction to the course expectations; plus bulk of the time reviewing CSE471.
2. (Deterministic) Planning (Here is a tutorial on landmark heuristics)
   - L2 Audio of [Aug 25, 2010] ( NEW NEW Video of the lecture video (4gb) Trends in AI; Start of Planning; different kinds of planning; atomic account of classical planning and its limitations; propositional account--STRIPS representation; Progression.
   - L3 Audio of [Aug 30, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) STRIPS representation-->ADL representation; conditional and quantified effects and compiling them into canonical representation. Issues on handling multi-valued fluents (state variables); Progression; Regression; blind-search tradeoffs.
   - L4 Audio of [Sep 1st, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Different ways of proving the correctness of plans. Causal proof and plan-space (partial order) planning. Discussion of the partial order planning algorithm. Observations on flaw selection heuristics etc. Discussion on handling conditional effects in regression and plan-space planning. Discussion on handling lifted (partially specified) actions in regression and partial order planning.
   - L5 Audio of [Sep 8th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Reachability analysis and planning graph heuristics. Understanding planning graph as an optimistic projection of reachability. h_level, h_sum, h_max and h_{relaxed-plan} heuristics. Relaxed plan extraction (and how it becomes hard with mutual exclusions).
   - L6 Audio of [Sep 13th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Heuristics vs. search strategies; PG heuristics for progression vs. regression; progression vs. regression--can the balance have something to do with ergodicity of the benchmark domains? backchaining as a meta-idea with multiple realizations.
   - L7 Audio of [Sep 15th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Negative interactions and the idea of capturing them with level-specific n-ary mutexes; static mutex identification rules and which of them are minimally required; mutex propagation rules for binary mutexes. mutexes, memos and graphplan completeness theorem. Converting plan extraction from planning graph into a SAT problem.
   - L8 Audio of [Sep 20th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Majority of the class on compiling bounded length planning into SAT and CSP. Planning graph compilation first, followed by the more general view of encodings coming from lines of proof of correctness. State-based vs. causal-proof encodings. Planning graph encoding as explanatory frame-axiom encoding with mutex propagation. Use of negative interactions in planning graph heuristics (and the adjusted sum heuristic); using planning graph heuristics in partial-order planning.
   - L9 Audio of [Sep 22nd, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Majority of the class on landmark heuristics (using Richter/Karpas ICAPS 2010 tutorial), with digressions into causal-graph heuristic (used in Fast Downward), and cost-propagation on planning graphs (used in cost-based landmark analysis). Final 10 minutes are devoted to motivating the atomic model for stochastic worlds and general reward structures.
3. MDPs
   - L10 Audio of [Sep 27th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Markov Decision Processes; background, terminology, motivations
   - L11 Audio of [Sep 29th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Computing the value of a policy. Optimal policy construction for finite-horizon MDPs. Relations between finite-horizon MDPs and bounded length planning. Brief discussion of indefinite horizon problems--and their need for sink (absorbing) states. Infinite horizon problems--and how discount factor affects the convergence rate. The idea of infinite horizon MDP value iteration as just a "repeat until" version of the finite horizon MDP value iteration (where the until condition checks that the max-norm difference between two iterations is less than epsilon)
   - L12 Audio of [Oct 4th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Infinite horizon MDP value iteration; understanding bellman update as a contraction operator, greedy policy for a given value

5. Planning in Belief-space
   - L16 Audio of [Oct 18th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Handling state uncertainty--the road map; atomic model with and without stochastic uncertainty; belief states; applying actions to belief states (and why action preconditions can get in the way); why stochastic uncertainty--which is additional knowledge-- seems to increase the difficulty of the problem by exploding the belief space; two ways of reducing state uncertainty--with causal actions and with observational ones (and realizing that in general we can have actions that have both casual and observational effects); observation model; how observations partition the state space--and how the number of partitions corresponds to the degree of observability (the notion of idf of the observation). State estimation and planning problems in belief-space. For planning, the difference between conformant and contingent planning; the difference between full vs. limited contingency planning.
   - L17 Audio of [Oct 18th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Discussion of factored approaches for belief-space planning; discussion on BDDs and BDD-based planning; progression and regression for conformant planning; sensing actions; heuristics for conformant planning--all-states determinization; labelled uncertainty graphs.
   - L18 Audio of [Oct 25th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Part 1: Discussion of heuristics for belief-space planning; the idea of state interactions (in addition to action interactions). The merged; unioned and LUG planning graphs. The notion of cross-world mutexes and how that leads to CGP (conformant graphplan); a little on heuristics for sensing actions.
6. POMDPs
   - L18 Contd: Part 2: POMDPS start. The model. The non-markovian nature of decisions based on observations and the need for observation history. Two ways of compactly representing the observation history--as belief-space and as a policy represented by a finite-state controller. The depressing complexity results on POMDPS.
   - L19 Audio of [Oct 27th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) POMDP discussion continued. Formally showing that POMDP is an MDP in the belief space. Discussion of the value iteration for finite horizon POMDP. Ideas for improving the complexity of value iteration.
   - L20 Audio of [Nov 1st, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Approximating POMDP value function (with FOMDP one as the upper bound and NOMDP one as the lower bound). Online approaches for POMDP. (Comparing POMDP online search to non-deterministic belief space search with observations).
7. Reinforcement Learning
   - L21 Audio of [Nov 3rd, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Reinforcement learning--the problem, the dimentions of RL algorithms. Passive RL with montecarlo. Passive RL with ADP. Generalization in RL. Model correctness/completeness considerations and notions of robustness.
   - L22 Audio of [Nov 8th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Active learning, exploration policies; GLIE policies; model-free learning--Temporal difference learning; Q-learning; SARSA and on- vs. off- policy learning.
   - L23 Audio of [Nov 10th, 2010] ( NEW NEW Video of the lecture part 1 (4gb) part 2 (1gb) Monte-carlo vs. Temporal difference learning; and the idea of TD(lambda). Generalizationi in RL; basics of feature functions

REPORT

## Reflections on the Planning Summer School

Autho

I am a cognitive scientist, so when I received an e-mail advertising an AI Summer School in Planning, my first thought was "I know AI is still kicking and screaming, but I thought planning at least had disappeared quietly." Well, if they're disappearing to Cyprus, they're doing something right, I then thought, and the lecture titles gave that impression as well. "A Unifying and Brand-name-free Introduction" is something which hardly any field can promise, and "Complexity" has always been a dirty word in most of AI. With these courses, two on heuristics (my area), and another by the Bridge programming genius Dana Nau, the School looked well attractive.

**Figure 1:** In a lectu

The courses delivered c barao Kambhampati ( sented a unifying view of planning as successively refining a search space, quickly introducing and absorbing forward-, backward-, method-level-, and a host of other planning techniques. Hector Geffner ("Hector heuristic") spoke on the $h^m$ heuristic which generalises much of GraphPlan and slays dragons in polynomial time. The courses then ventured out of blocks-world, relating other subjects and solving real problems. Bernhard Nebel ("complexity Nebel") explained what "polynomial time" means in a crash course on complexity theory (complete with notes occupying a large chunk of NPSPACE). Derek Long

("domain Derek") reinforced the plexity by giving an easy, familiar lem in an abstact vocabulary (w nearly intractable to do by han he explored ways of pre-process domain knowledge. Dana Nau ( Dana") continued the gospel of planning, taking the theory of O composition and applying it to pr thing called the real-world. Ma turned us to the world of theo entirely so, and reviewed techn ning with common things like res Finally, Paolo Traverso encoded

Lots of us, although coming from areas as diverse as natural language generation and knowledge

tunately, Derek didn't say that anyone has already done it (though I sat through that course

ven better, I nd who want

fference were h is very re- in this field reat working to come, just four days in he art in this leas and con- bring back which tops na and Sub- editerranean, sing Ordered ment world- ng sun.

**Heneveld** Institute for Representation and Reasoning, Division of

Do not go gentle into that good night
Old age should burn and rave at close of day
Rage rage against the dying of the light
--Dylan Thomas

as and con- bring back to Edinburgh, but there is one image which tops them all, the ideal of academia: Dana and Subbarao standing ankle-deep in the Mediterranean, in the same spot for an hour, discussing Ordered Task Decomposition and the Refinement world-view against the backdrop of a setting sun.

PROCEEDINGS OF THE 1986 WORKSHOP

TIMBERLINE OREGON

# REASONING ABOUT ACTIONS & PLANS

EDITED BY
MICHAEL P. GEORGEFF
AMY L. LANSKY



# REASONING ABOUT ACTIONS & PLANS

PROCEEDINGS OF THE 1986 WORKSHOP

EDITED BY
MICHAEL P. GEORGEFF & AMY L. LANSKY
SRI INTERNATIONAL

JUNE 30—JULY 2, 1986
TIMBERLINE, OREGON

SPONSORED BY
THE AMERICAN ASSOCIATION FOR ARTIFICIAL INTELLIGENCE
and the
CENTER FOR THE STUDY OF LANGUAGE AND INFORMATION

MORGAN KAUFMANN PUBLISHERS, INC.
95 FIRST STREET
LOS ALTOS, CALIFORNIA 94022

# CONTENTS

# Rao's Complaints: Then

**Then**

- What good are expressive and ambitious planning paradigms when we have so little scalability?
  - Need to work on search control
  - Need benchmarks to measure progress

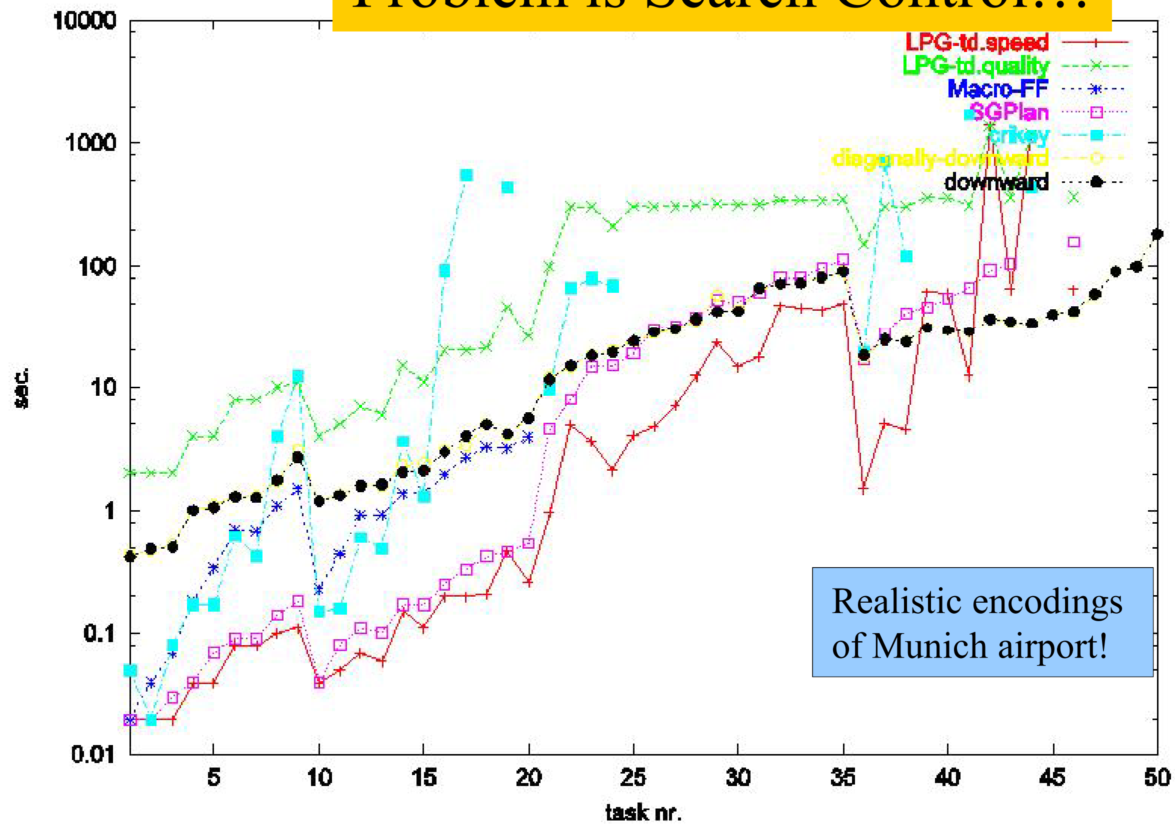
[. . .] *Search is usually given little attention in this field, relegated to a footnote about how "Backtracking was used when the heuristics didn't work."*

Drew McDermott 1991

# Scalability was the big bottle-neck…
# We have figured out how to scale synthesis..

Problem is Search Control!!!

- Before, planning algorithms could synthesize about 6 – 10 action plans in minutes
- Significant scale-up in the last decade
  - Now, we can synthesize 100 action plans in seconds.




Realistic encodings of Munich airport!

**The primary revolution** in planning in the recent years has been **methods to** scale up plan synthesis

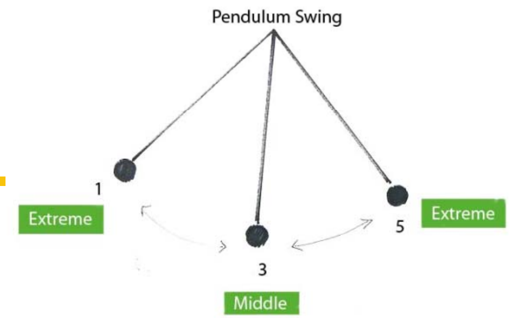
# ..and we have done our fair bit...

So, Rao should be happy..

Right?

Wrong!

# Rao's Complaints: Then & Now




**Then**

- What good are expressive and ambitious planning paradigms when we have so little scalability?
  - Need to work on search control
  - Need benchmarks to measure progress

**Now**

- What good are scalable planners if all they want to do is stack blocks all the way to the moon?
  - Streetlight effect
- There should be more to planning than combinatorial search!

[. . .] *Search is usually given little attention in this field, relegated to a footnote about how "Backtracking was used when the heuristics didn't work."*

Drew McDermott [26, p. 413]

I love planning man. It is just search!

A graduate student in a Taverna in Thessaloniki during ICAPS 2009

I'd rather learn from one bird how to sing
than teach ten thousand stars
    how not to dance

*ee cummings*

ARIZONA STATE UNIVERSITY

# Lecture Overview…

I'd rather learn from one bird how to sing
than teach ten thousand stars
how not to dance

*ee cummings*

- How to use our hammers *wisely*
- How to be skeptical of our benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning

# Lecture Overview…

I'd rather learn from one bird how to sing
than teach ten thousand stars
how not to dance

*ee cummings*

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

# On Using Our Hammers Wisely

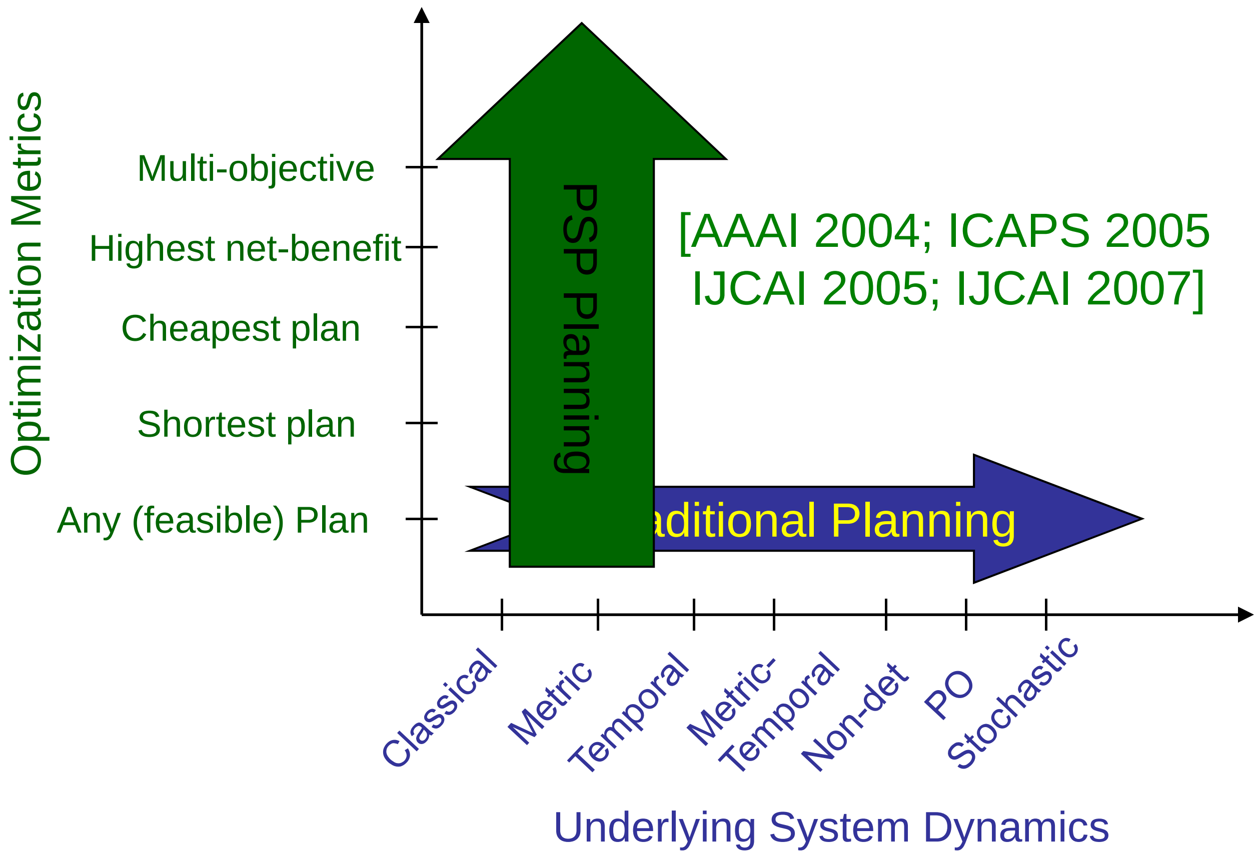
Make things as simple as possible,
but not simpler
-Attributed to Einstein

- Classical Planners have justifiably become our hammers… This is mostly GOOD NEWS
  - We want to coax all other planning problems into formats that will allow us to maximally utilize the progress made in scaling up classical planning
  - ..But, we need to be careful, lest we lose the essence of the expressive planning problems during the coaxing (compilation)
  - Some examples..
    - Cost-based Planning (ε-cost trap)
    - Temporal Planning  (Required Concurrency)
    - Stochastic Planning (Biased Determinizations)

# Lecture Overview…

> I'd rather learn from one bird how to sing
> than teach ten thousand stars
> how not to dance
>
> *ee cummings*

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)



Optimization Metrics
Multi-objective
Highest net-benefit
Cheapest plan
Shortest plan
Any (feasible) Plan
PSP Planning
[AAAI 2004; ICAPS 2005
IJCAI 2005; IJCAI 2007]
aditional Planning
Classical
Metric
Temporal
Metric-
Temporal
Non-det
PO
Stochastic
Underlying System Dynamics

# Partial Satisfaction/Over-Subscription Planning

- Traditional planning problems
  - Find the (lowest cost) plan that satisfies *all* the given goals
- PSP Planning
  - Find the highest utility plan given the resource constraints
    - Goals have utilities and actions have costs
- …arises naturally in many real world planning scenarios
  - MARS rovers attempting to maximize scientific return, given resource constraints
  - UAVs attempting to maximize reconnaisance returns, given fuel etc constraints
  - Logistics problems resource constraints
- … due to a variety of reasons
  - Constraints on agent's resources
  - Conflicting goals
    - With complex inter-dependencies between goal utilities
  - Soft constraints
  - Limited time

[AAAI 2004; ICAPS 2005; IJCAI 2005; IJCAI 2007; ICAPS 2007; CP 2007]

# Classical vs. Partial Satisfaction Planning (PSP)

Classical Planning

- Initial state
- Set of goals
- Actions

Find a plan that achieves *all* goals

(prefer plans with fewer actions)

Partial Satisfaction Planning

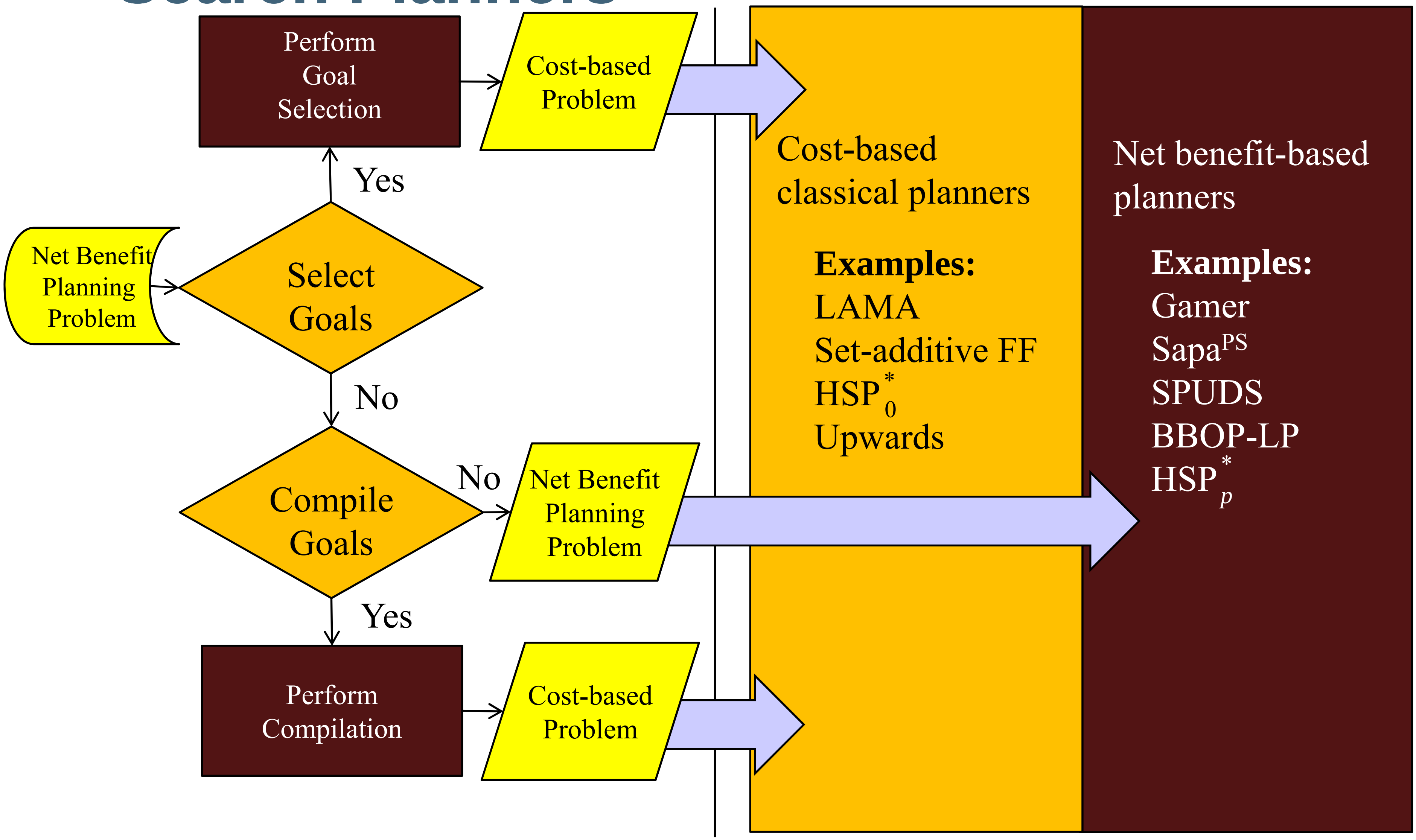
- Initial state
- Goals *with differing utilities*
- Actions *with differing costs*

Find a plan with highest *net benefit*

(cumulative utility – cumulative cost)

(best plan may not achieve all the goals)

# How to Leverage Modern Heuristic Search Planners



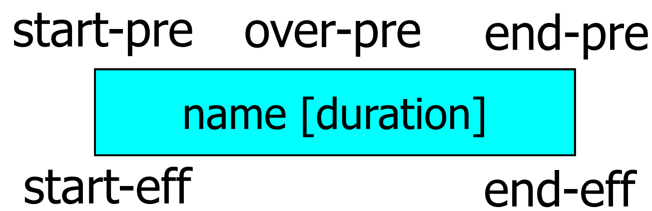
# Surrogate Search to avoid $\varepsilon$-cost traps

- Most planners use A* search variants
- A* is susceptible to $\varepsilon$-cost traps
  - $\varepsilon$ is the ratio of the lowest to highest cost action
  - Would be small if there is large cost variance (which is usually the case in planning domains—e.g. cost of boarding vs. flying)
  - In such cases, A*'s propensity to conflate discovery and optimality proof proves to be its undoing
    - Consider an optimal solution at depth 10 and the second best at depth 1000
  - This pathology has been noticed [e.g. LAMA], but the cause ($\varepsilon$-cost trap wasn't) leading to *ad hoc* stop gaps
- Solution to $\varepsilon$-cost trap is to guide A* search with a *surrogate* evaluation function that:
  - has a significantly higher $\varepsilon$
  - ..and is cost (objective) focused
- One idea is to go with *size-based* evaluation function as the surrogate
  - This one has $\varepsilon=1$ but is not particularly well-focused on the objective
    - Surprisingly, surrogate search with it it does significantly better than direct cost-based search
- A better alternative is to consider cost sensitive size-based evaluation function (which estimates the size of the cheapest path through the current state)

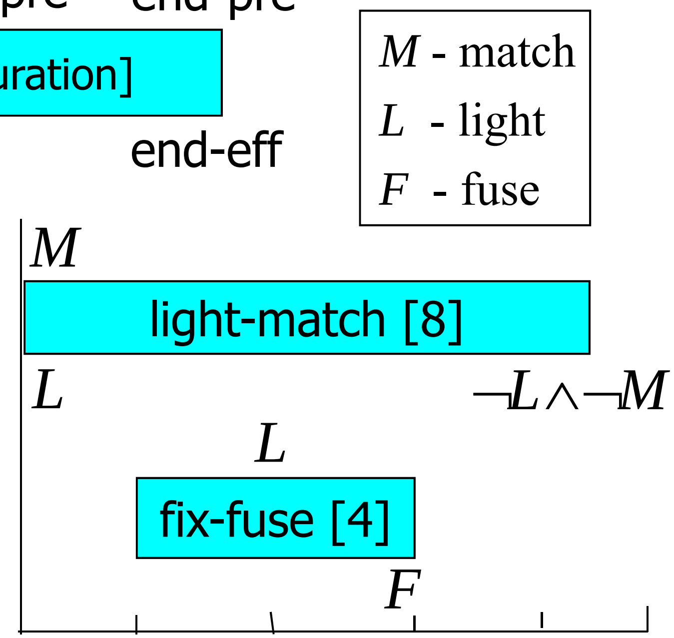
# Lecture Overview…

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

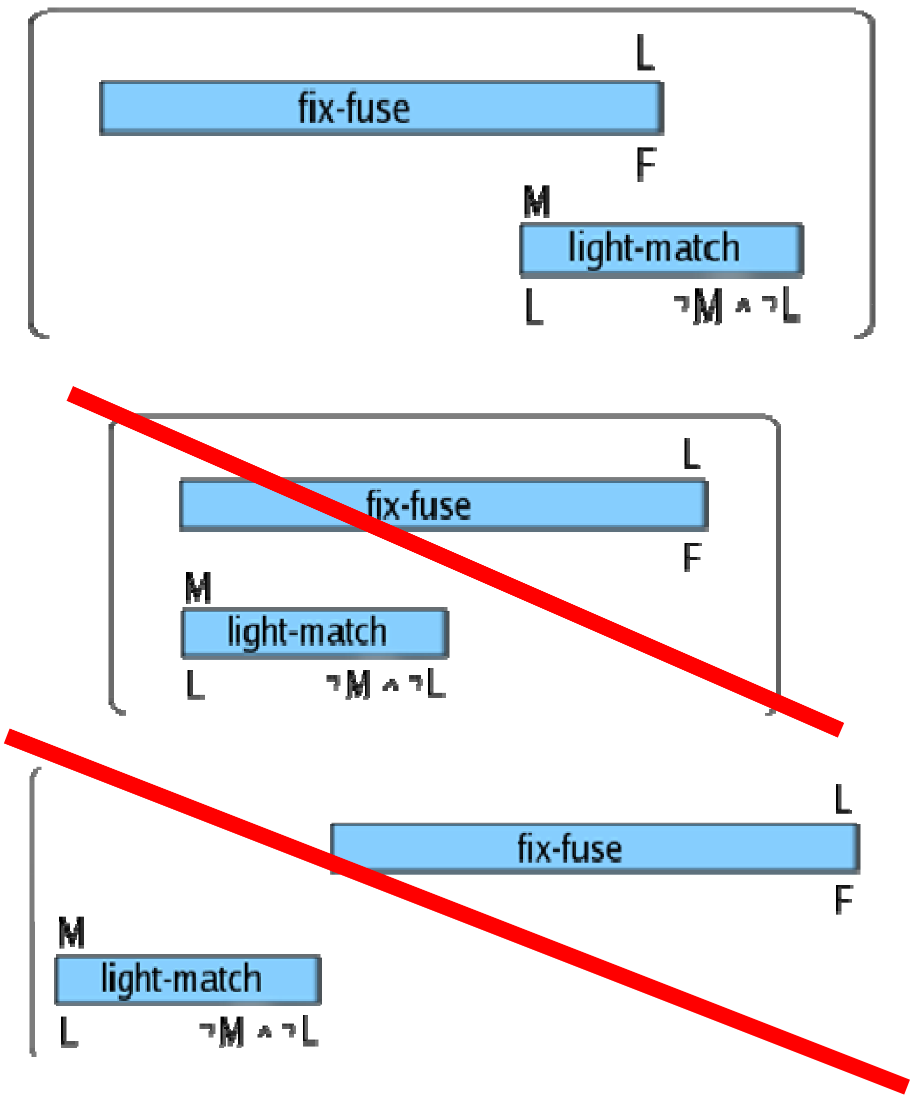
# Temporal Planning



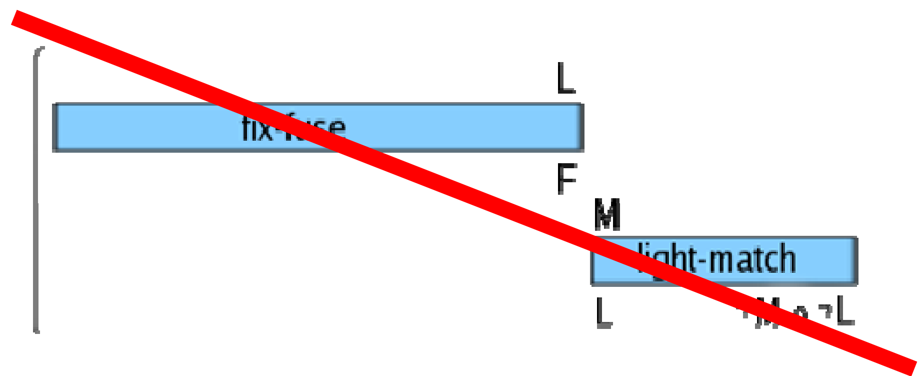

- Plan-space is natural
  - Zeno, IxTET etc.
- Desire to exploit classical planning progress
  - Extended planning graph [TGP]
  - ***State-space?***
    - Problem: Infinite number of time points
- Decision Epochs
  - Restrict start times to events
    - *Competition winners*
    - *Reachability heuristics*



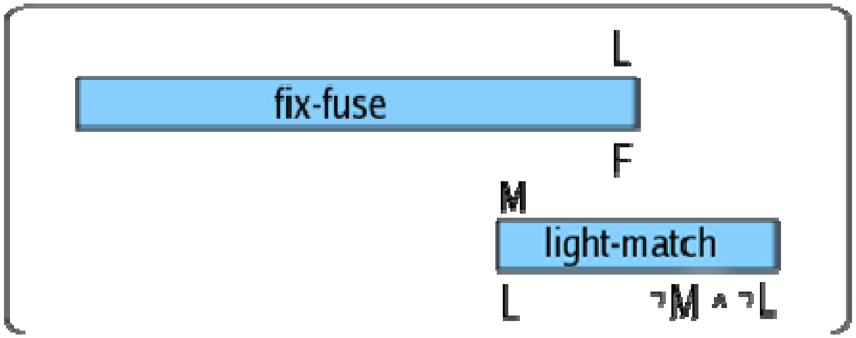
# Short matches




- No epoch available
  - “middle of nowhere”
- Decision Epoch Planning is incomplete!



# Short matches




Wow!

# Troubling Questions

- What do/should the IPCs measure?

*Essence of Temporal Planning*

- Required Concurrency
- Temporally Simple ≈ Classical
- Temporally Expressive ≈ **Harder**

- Can Decision Epoch Planning be fixed?

- No.
- But!
- DEP+
  - "Less" incomplete
- TEMPO
  - Reachability heuristics

# Required Concurrency

- Temporally Simple Languages
  - Concurrency ***never*** necessary
  - …but can be exploited for quality
- Temporally Expressive Languages
  - Can specify problems such that concurrency is ***needed***

# Temporal Action Languages

**S**tart-pre **O**ver-pre **E**nd-pre

name [duration] $\mapsto L^{s,\,o,\,e}_{s,\,e}$

**S**tart-eff **E**nd-eff

---

**O**ver-pre

name [duration] $\mapsto L^{o}_{e}$

**E**nd-eff

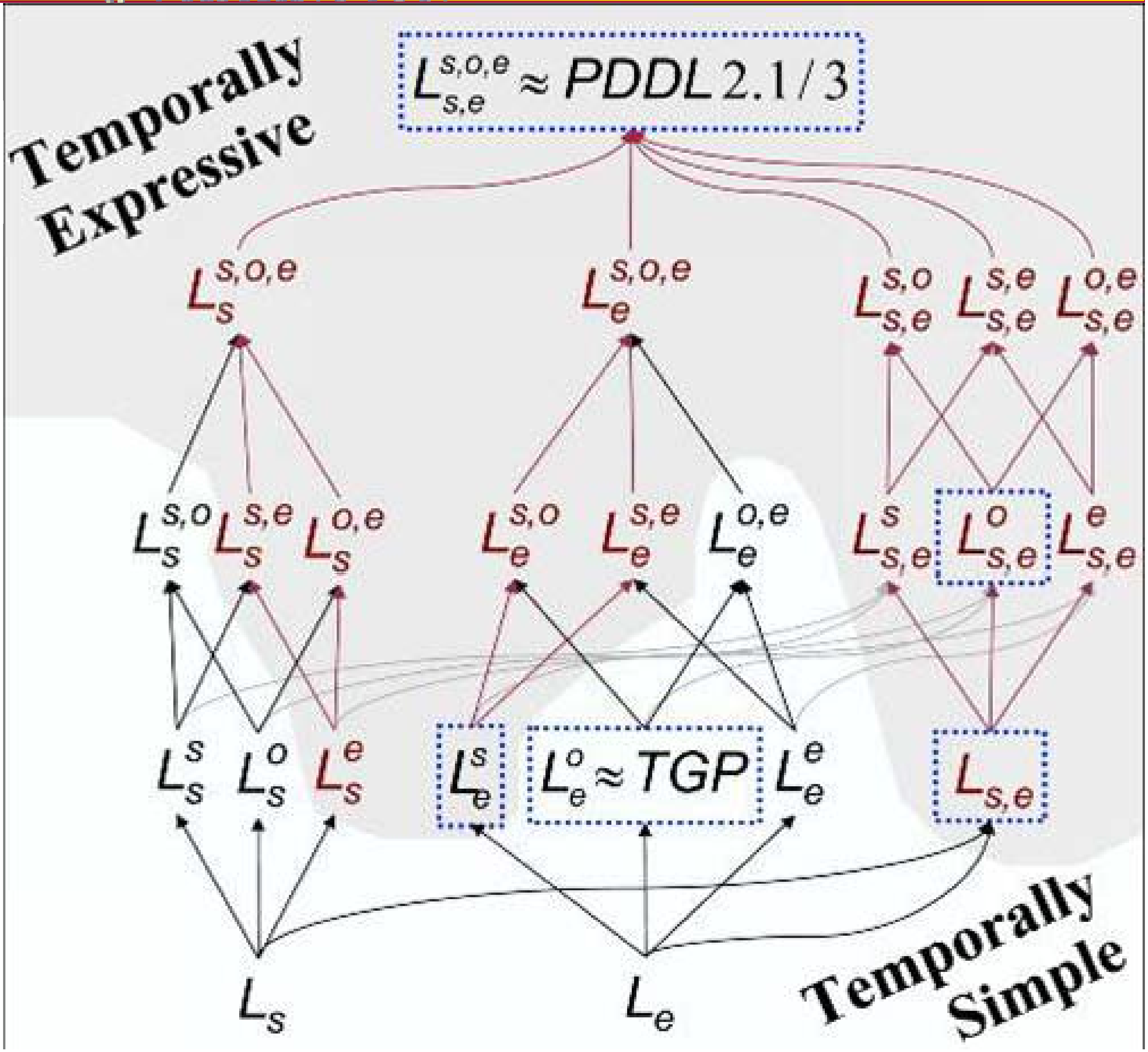
# Essence of Temporal Planning

**s** **o** **e**

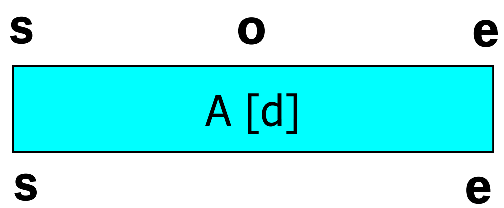
A [d]

**s** **e**

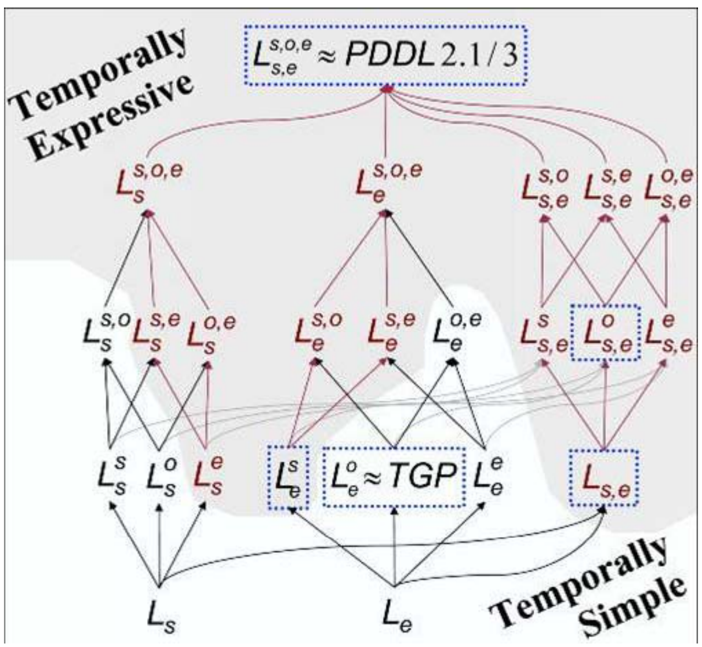
$$L^{s,o,e}_{s,e}$$



# Temporal Action Languages

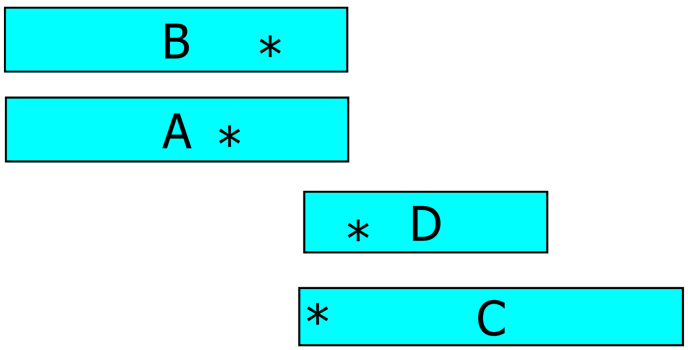



$$L^{s,o,e}_{s,e}$$

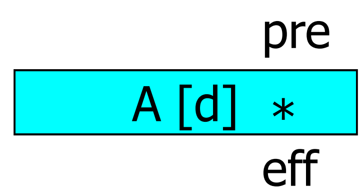



- Temporally Simple
  - Rescheduling is possible
    - MIPS, SGPlan, LPG, …
  - Sequential planning is complete – "optimal" ?
    - TGP, yes
    - In general, yes
- Temporally Expressive
  - $L_{s,e}$ $L^{s}_{e}$ $L^{e}_{s}$
    - Temporal Gap
      - Before-condition and effect
      - After-condition and effect
      - Two effects
    - Temporally Simple $\Rightarrow$ No Temporal Gap

# No Temporal Gap ⇒ Classical + Scheduling




- Forbidding temporal gap implies
  - All effects at one time
  - Before-conditions meet effects
  - After-conditions meet effects
- Unique transition per action



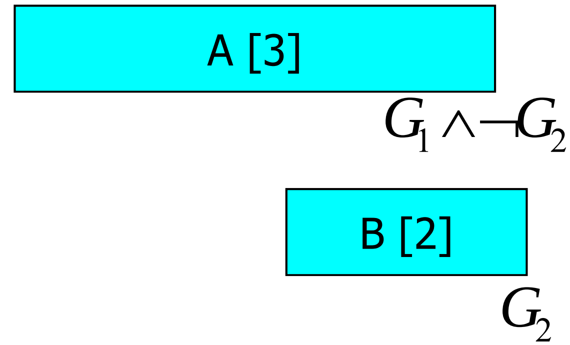

- Theorem: Every concurrent plan is an O(n) rescheduling of a sequential plan
  - And vice versa

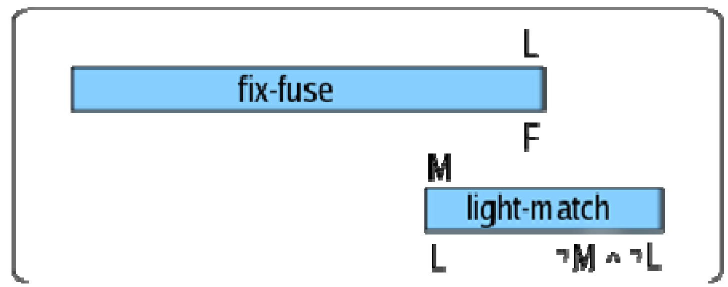
# Wow!

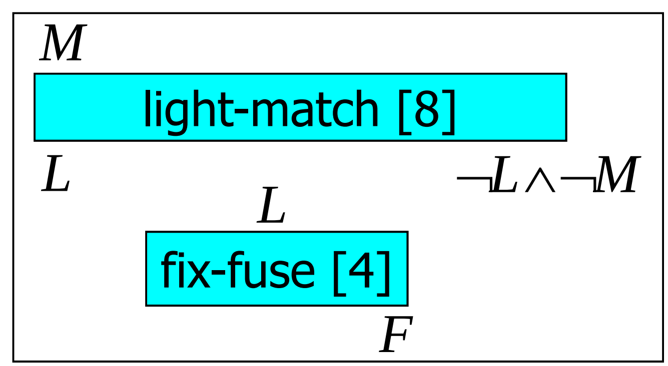
- Temporally Simple ⇒
  Classical + Scheduling
- Winners incomplete for all
  Temporally Expressive Languages
- Most/all benchmarks are classical!

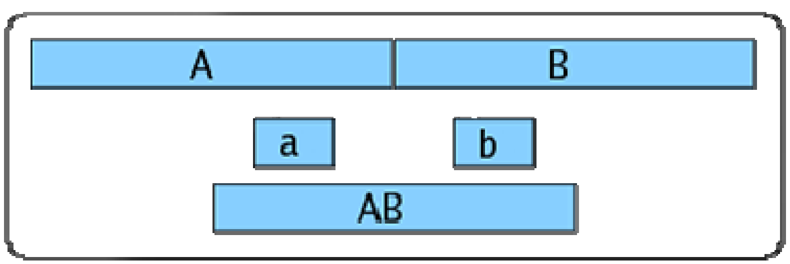
# Decision Epoch Planning: DEP

- Only start actions after events
- Choose
  - Start an action
  - Advance epoch
- Temporally Simple
  - Complete, suboptimal
- Temporally Expressive
  - Incomplete, suboptimal



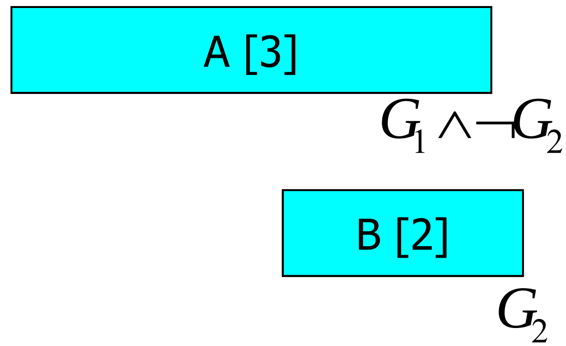



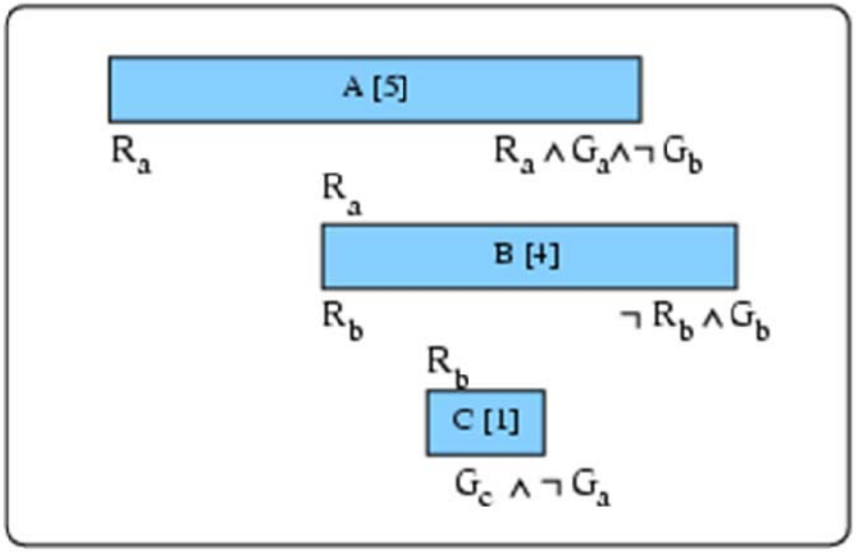



# Generalized DEP: DEP+

- Also end actions after events
- Choose
  - Start an action
  - End an action
  - Advance epoch




- Temporally Simple
  - Complete, optimal
- Temporally Expressive
  - Incomplete, suboptimal

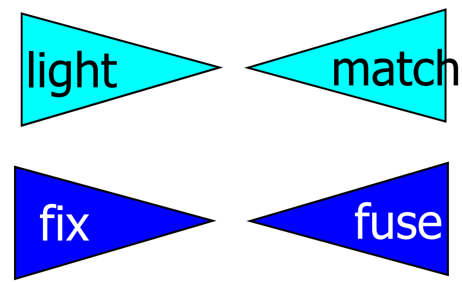


# State of the Art: Incomplete or Slow

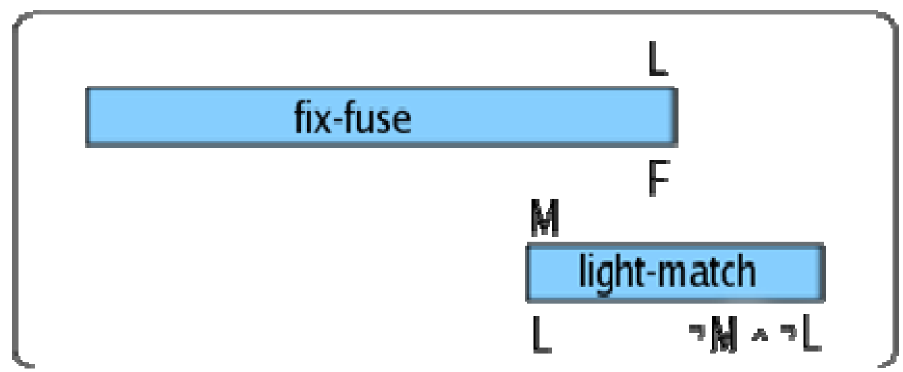
- Metric-FF, MIPS, SGPlan, SAPA, TP4, TPG, HSP*, ...
  - Guarantees only for temporally simple languages
  - Can solve some concurrent problems
    - Light-match, but not short-match
    - Difficult to detect
- ZENO, IxTeT, VHPOP, LPGP, ...
  - Complete
  - Slow

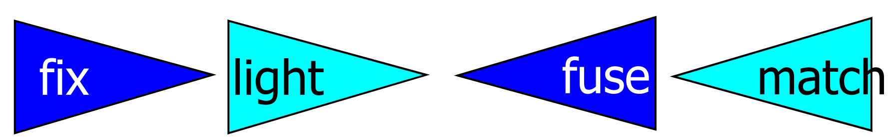
# Interleaving-Space: TEMPO

- Delay dispatch decisions until afterwards
- Choose
  - Start an action
  - End an action
  - ~~Make a scheduling decision~~
- Solve temporal constraints




- Temporally Simple
  - Complete, Optimal
- Temporally Expressive
  - Complete, Optimal




[Colin planner]

# Conclusions

- Required concurrency is *the* essence of temporal planning
  - Otherwise classical planner + O(n) scheduling suffices
  - Simple test for required concurrency: Temporal gap
- Decision epoch planning is fundamentally incomplete
  - But DEP+ may solve most real-world problems
- Complete state-space temporal planning: TEMPO
  - Allows leveraging of state-based reachability heuristics
- !!!!!

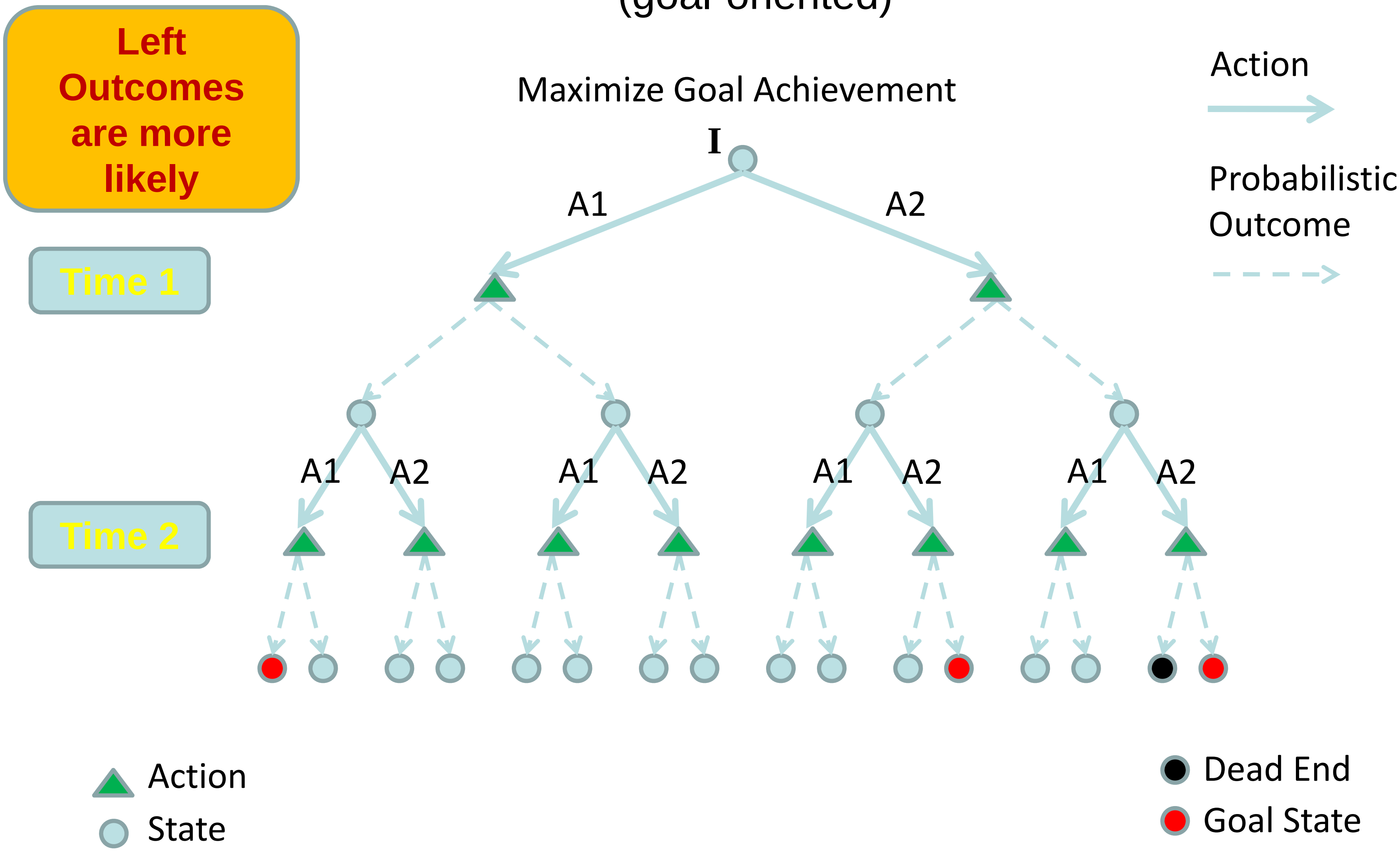
Lesson: Be wary of the temptation of efficiency at the expense of essence of expressive planning

# Lecture Overview…

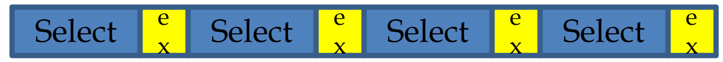
- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

# Probabilistic Planning

(goal-oriented)



# How to compete?


## Off-line policy generation

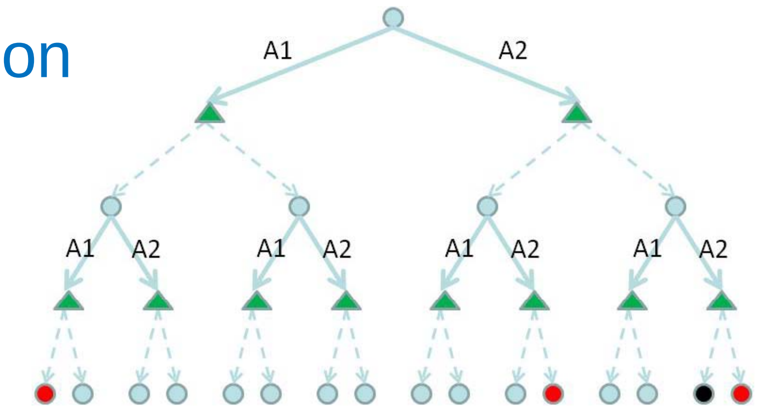
- First compute the whole policy
  - Get the initial state
  - Compute the optimal policy given the initial state and the goals
- Then just execute the policy
  - Loop
    - Do action recommended by the policy
    - Get the next state
      - Until reaching goal state
- Pros: Can anticipate all problems;
- Cons: May take too much time to start executing

## Online action selection

- Loop
  - Compute the best action for the current state
  - execute it
  - get the new state
- Pros: Provides fast first response
- Cons: May paint itself into a corner..

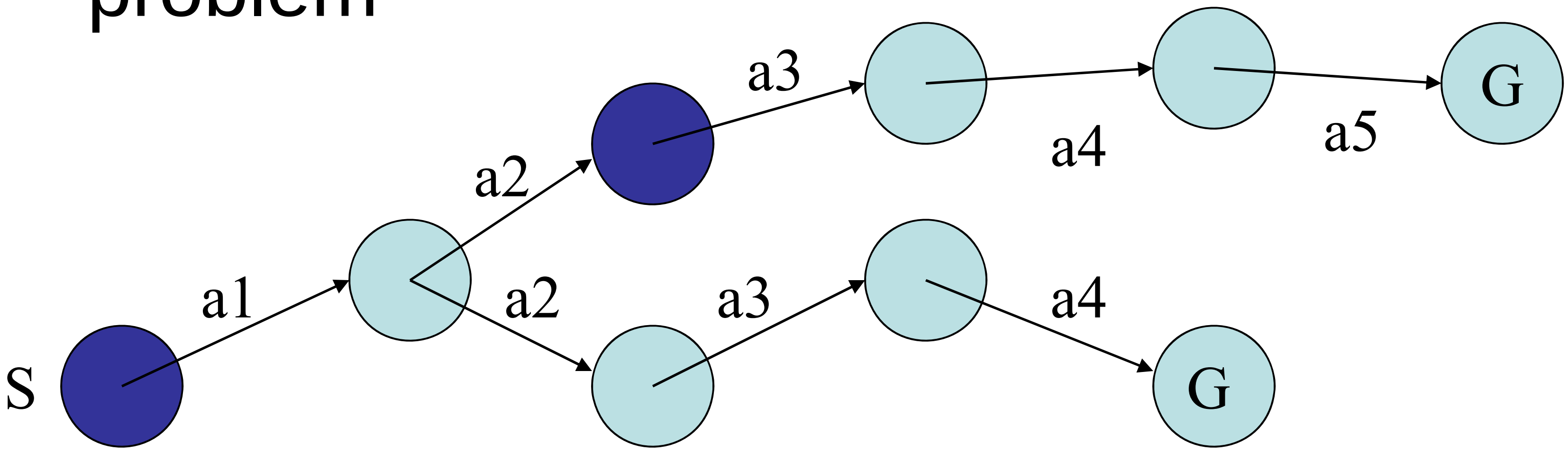
# Determinizations

- Determinizations allow us a way to exploit classical planning technology
  - Most-likely outcome determinization
    - Inadmissible
    - e.g. if only path to goal relies on less likely outcome of an action
  - All outcomes determinization
    - Admissible, but not very informed
    - e.g. Very unlikely action leads you straight to goal
  - Hindsight Optimization
    - *Sample* determinizations..
      - *The sampling (rather than a static process) determines what effects an action has at each time step*



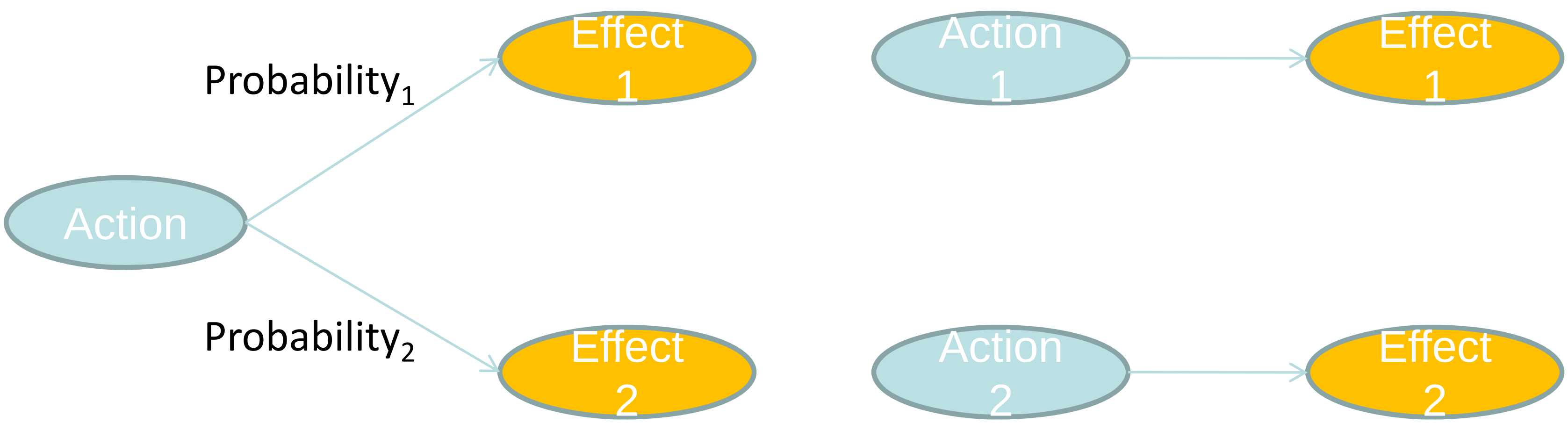
# FF-Replan

- Simple replanner
- Determinizes the probabilistic problem
- Solves for a plan in the determinized problem



# All Outcome Replanning ($FFR_A$)

ICAPS-07



# $1^{st}$ IPPC & Post-Mortem..

## IPPC Competitors

- Most IPPC competitors used different approaches for offline policy generation.
- One group implemented a simple online "replanning" approach in addition to offline policy generation
  - Determinize the probabilistic problem
    - Most-likely vs. All-outcomes
  - Loop
    - Get the state S; Call a classical planner (e.g. FF) with [S,G] as the problem
    - Execute the first action of the plan
- Umpteen reasons why such an approach should do quite badly..

## Results and Post-mortem

- To everyone's surprise, the replanning approach wound up winning the competition.
- Lots of hand-wringing ensued..
  - May be we should require that the planners really really use probabilities?
  - May be the domains should somehow be made "probabilistically interesting"?
- Current understanding:
  - The "replanning" approach is just a degenerate case of hind-sight optimization

# Hindsight Optimization (Online Computation of $V^{HS}$ )

- Pick action a with highest Q(s,a,H) where
  - $Q(s,a,H) = R(s,a) + \Sigma T(s,a,s')V^*(s',H-1)$
- Compute V* by sampling
  - H-horizon future $F^H$ for M = [S,A,T,R]
    - Mapping of state, action and time (h<H) to a state
  - $S \times A \times h \rightarrow S$
    - Common-random number (correlated) vs. independent futures..
    - Time-independent vs. Time-dependent futures
- Value of a policy π for $F^H$
  - $R(s,F^H, \pi)$
- $V^*(s,H) = \max_\pi E_F{}^H [ R(s,F^H,\pi) ]$
  - But this is still too hard to compute..
  - Let's swap max and expectation
- $V^{HS}(s,H) = E_F{}^H [\max_\pi R(s,F^H,\pi)]$
  - $\max_\pi R(s,F^{H-1},\pi)$ is approximated by FF plan

- **$V^{HS}$ overestimates V***
- **Why?**
  - **Intuitively, because $V^{HS}$ can assume that it can use different policies in different futures; while V* needs to pick one policy that works best (in expectation) in all futures.**
- **But then, $V^{FFRa}$ overestimates $V^{HS}$**
  - **Viewed in terms of J*, $V^{HS}$ is a more informed admissible heuristic..**

# Relaxations for Stochastic Planning

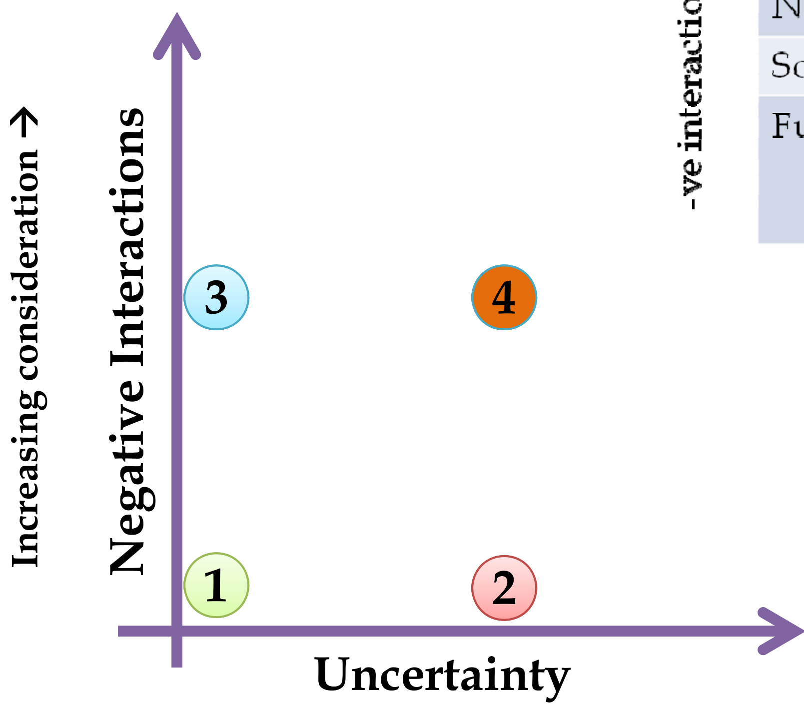
- Determinizations can also be used as a basis for heuristics to initialize the V for value iteration [mGPT; GOTH etc]
- Heuristics come from relaxation
- We can relax along two separate dimensions:
  - Relax –ve interactions
    - Consider +ve interactions alone using relaxed planning graphs
  - Relax uncertainty
    - Consider determinizations
  - Or a combination of both!

# Dimensions of Relaxation

Uncertainty

-ve interactions

| | None | Some | Full |
|---|---|---|---|
| None | Relaxed Plan | McLUG | |
| Some | SAS RP | | |
| Full | FF/LPG | Limited width Stoch Planning | |



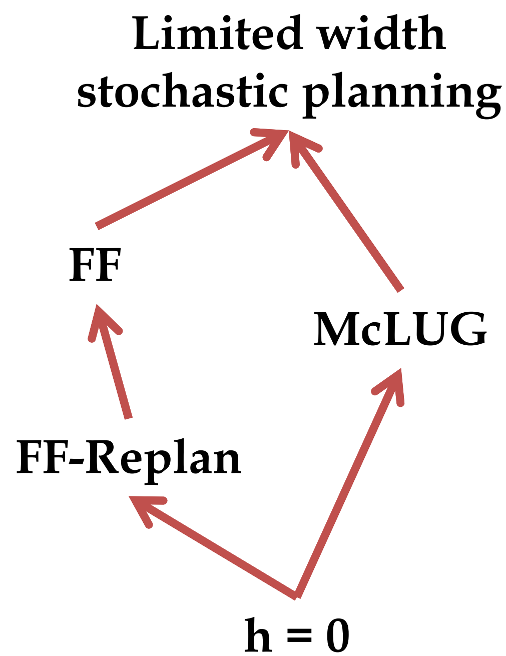

1 **Relaxed Plan Heuristic**

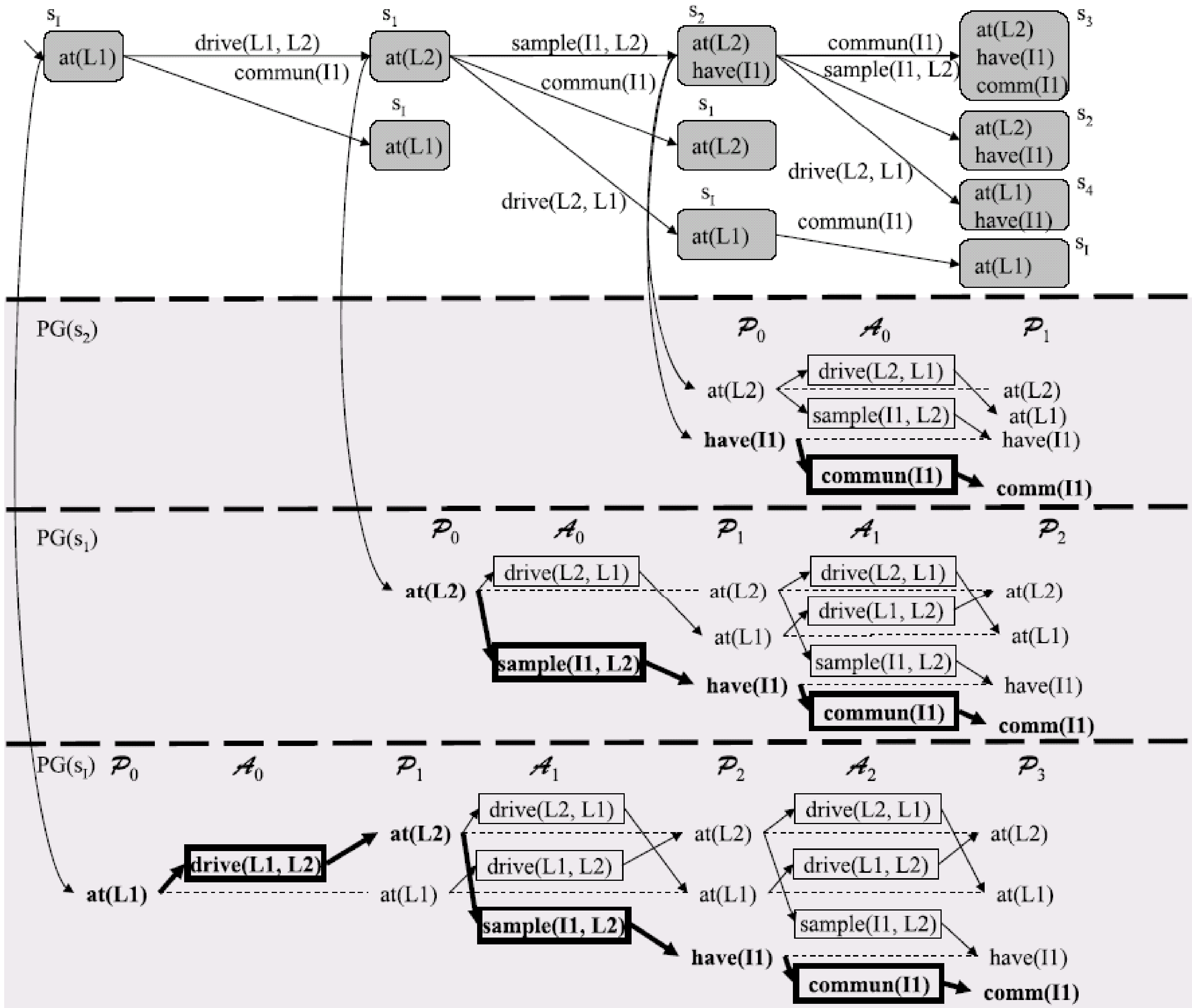
2 **McLUG**

3 **FF/LPG**

4 **Limited width stochastic planning?**

**Reducing Uncertainty**
Bound the number of stochastic outcomes → Stochastic "width"





s_I
at(L1)
drive(L1, L2)
commun(I1)
s_1
at(L2)
s_I
at(L1)
sample(I1, L2)
commun(I1)
drive(L2, L1)
s_2
at(L2)
have(I1)
s_1
at(L2)
s_I
at(L1)
commun(I1)
sample(I1, L2)
drive(L2, L1)
commun(I1)
s_3
at(L2)
have(I1)
comm(I1)
s_2
at(L2)
have(I1)
s_4
at(L1)
have(I1)
s_I
at(L1)
PG(s_2)
P_0
A_0
P_1
at(L2)
drive(L2, L1)
sample(I1, L2)
at(L2)
at(L1)
have(I1)
have(I1)
commun(I1)
comm(I1)
PG(s_1)
P_0
A_0
P_1
A_1
P_2
at(L2)
drive(L2, L1)
sample(I1, L2)
at(L2)
at(L1)
have(I1)
drive(L2, L1)
drive(L1, L2)
sample(I1, L2)
commun(I1)
at(L2)
at(L1)
have(I1)
comm(I1)
PG(s_I)
P_0
A_0
P_1
A_1
P_2
A_2
P_3
at(L1)
drive(L1, L2)
at(L2)
at(L1)
drive(L2, L1)
drive(L1, L2)
sample(I1, L2)
at(L2)
at(L1)
have(I1)
drive(L2, L1)
drive(L1, L2)
sample(I1, L2)
commun(I1)
at(L2)
at(L1)
have(I1)
comm(I1)

# Lecture Overview…

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

# On Being Skeptical About our Benchmarks

- Progress in planning in the old days was hampered by lack of common benchmarks
  - The arguments of expressiveness with no guarantees of comparative efficiency..
- Thanks to IPC competitions, we have a huge chest of benchmarks.. But they pose their own problems
  - Arguments of efficiency with little heed to expressiveness. Undivided benchmarks can themselves inhibit progress
- Examples
  - Temporal Planning benchmarks indirectly inhibited work on expressive temporal planners
  - Most benchmarks inhibited work on lifted planners

# Lecture Overview…

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

# Temporal Benchmarks in IPC

- We saw that Required Concurrency is a hallmark of temporal planning
- We saw that DEP planners are incomplete for problems needing RC
- But, DEP planners "won" temporal planning track…
- Benchmarks must not require (much) concurrency
- How much?
  - None at all
- How do we show it?
  - Use temporal gap?
- Problem: "every" action has temporal gap

# Solution: Decompile temporal gap

- (navigate ?rover ?alpha ?omega)
  - Pre: (at start (at ?rover ?alpha))
  - Eff: (and
    - (at start (not (at ?rover ?alpha)))
    - (at end (at ?rover ?omega)))
- (navigate ?rover ?alpha ?omega)
  - (over all (=> (at ?rover) ?alpha ?omega))

Then, we can show that benchmarks never require concurrency!

# Benchmarks never require concurrency (except due to modeling bugs)

```
(:durative-action navigate
:parameters (?x - rover ?y - waypoint ?z - waypoint)
:duration (= ?duration 5)
:condition (and
    ;;(at start (at ?x ?y)) ;; MV Fluent
    ;;(at start (>= (energy ?x) 8)) ;; Resource Consumption
    (over all (can_traverse ?x ?y ?z))
    (at start (available ?x))
    (over all (visible ?y ?z)) )
:effect (and
    ;;(at start (decrease (energy ?x) 8)) ;; Resource Consumption
     (over all (consume (energy ?x) 8)) ;; Resource Consumption
    ;;(at start (not (at ?x ?y))) ;; MV Fluent
    ;;(at end (at ?x ?z)))) ;; MV Fluent
    (over all (-> (at ?x) ?y ?z)) ;; MV Fluent
))
```

```
;;(at ?x - rover ?y - waypoint)
(at ?x - rover ) - waypoint
```

# Real world required concurrency

- (and
  (lifted bowl-left)
  (lifted bowl-right))
- Spray-oil (during milling)
- Heat-beaker (while adding chemicals)
- Ventilate-room (while drying glue)
- …

In other words, benchmarks inhibited progress on temporal planning…

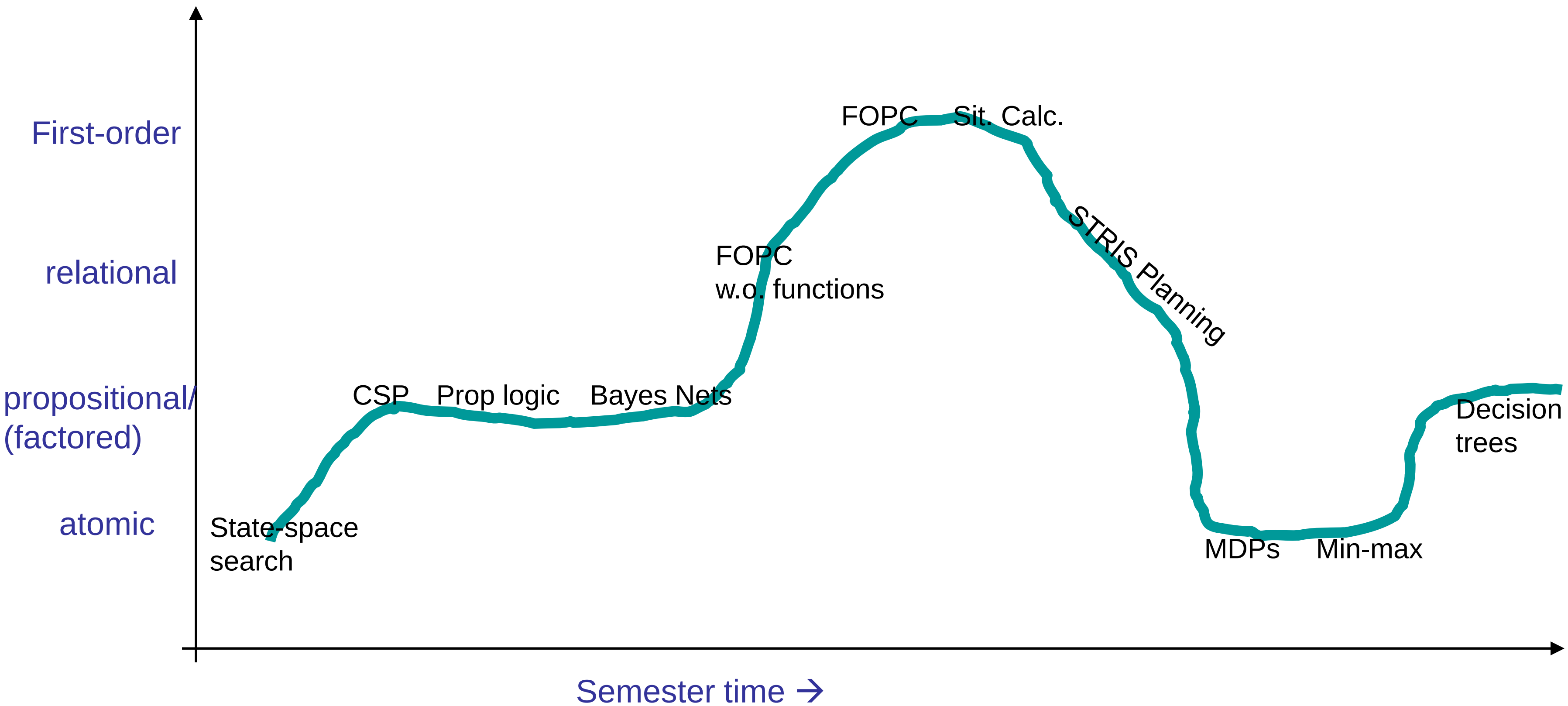
# Lessons for the Competition

- Competitors tune for the benchmarks
  - Most of the competitors simplify to TGP
- Either required concurrency is important
  - Benchmarks should test it
- Or it isn't
  - Language should be inherently sequential
- PDDL spec. highlights light-match
- RC occurs in the real world
  - Might need processes, continuous effects

# Lecture Overview…

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

# The representational roller-coaster in CSE 471




The plot shows the various topics we discussed this semester, and the representational level at which we discussed them. At the minimum we need to understand every task at the atomic representation level. Once we figure out how to do something at atomic level, we always strive to do it at higher (propositional, relational, first-order) levels for efficiency and compactness.

During the course we may not discuss certain tasks at higher representation levels either because of lack of time, or because there simply doesn't yet exist undergraduate level understanding of that topic at higher levels of representation..

# (Lack) of Relational Benchmarks

- Pre-1995, most planners were "relational"
  - That is, they would search in the space of partially instantiated plans
- Post-Graphplan, all planners search in the space of ground plans (propositional level)

# Plan Space Search


Then it was cruelly UnPOPped

The good times return with Re(vived)POP


In the beginning it was all POP.

# (Lack) of Relational Benchmarks

- Pre-1995, most planners were "relational"
  - That is, they would search in the space of partially instantiated plans
- Post-Graphplan, all planners search in the space of ground plans (propositional level)
- So what?
  - Planners can be easily defeated by a profusion of irrelevant objects and actions

- Solution: Develop effective solutions for "lifted planning"
  - Regression and Partial Order Planners can be easily lifted (and **were** lifted)
    - But they are currently slower than ground state search
- What is the resistance?
  - I am doing fine on benchmarks!
    - Why fix what is not broken?
  - But what if benchmarks are not realistic?

The parallel to temporal planning is not a coincidence
Effective temporal planning requires lifting time (precedence constraints)
Effective relational planning will require lifting binding (partial instantiation)

# Example

A has effects

$A^{\neg Q(y)}_{P(x)}$ with preconditions $L(x)$, $M(y)$

Over A

$L(x)$
$Q(a)$
$M(y)$

$x = a$
$y \neq a$

$\xrightarrow{\;A\;}$

$P(a)$
$Q(a)$

Notice that the state now has variable "Codesignation" and non-codesignation constraints

$t_0 \rightarrow A' \xrightarrow{Q(a)} t_\infty$

$P(a)@\infty$
$Q(a)@\infty$

$A$ — Establishing $P(a)@\infty$

$L(a)@A$
$M(y)@A$
$x = a$

Resolving unsafe link $(A' \xrightarrow{Q(a)} t_\infty, A)$

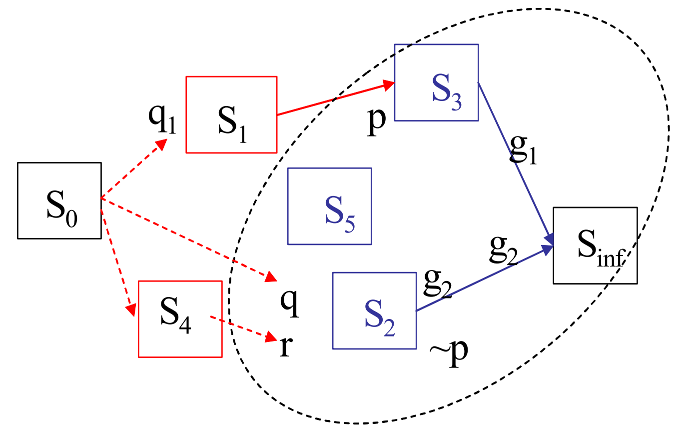
Promotion: $A < A'$

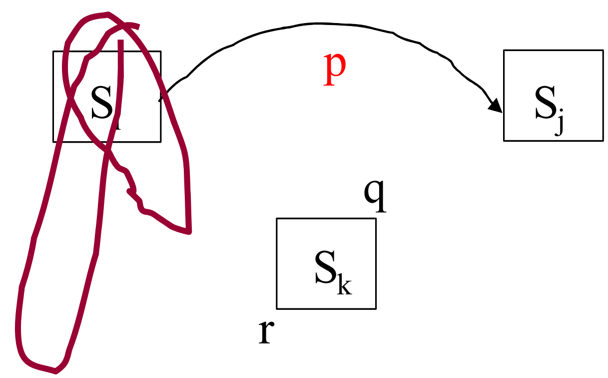
demotion: $t_\infty < A$

Separation: $y \neq a$

# PG Heuristics for Partial Order Planning

- Distance heuristics to estimate cost of partially ordered plans (and to select flaws)
  - If we ignore negative interactions, then the set of open conditions can be seen as a regression state
- Mutexes used to detect indirect conflicts in partial plans
  - A step threatens a link if there is a mutex between the link condition and the steps' effect or precondition
  - Post disjunctive precedences and use propagation to simplify




$$if\ mutex(p,q)\ or\ mutex(p,r)$$

$$S_k \prec S_i \vee S_j \prec S_k$$

# Lecture Overview…

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

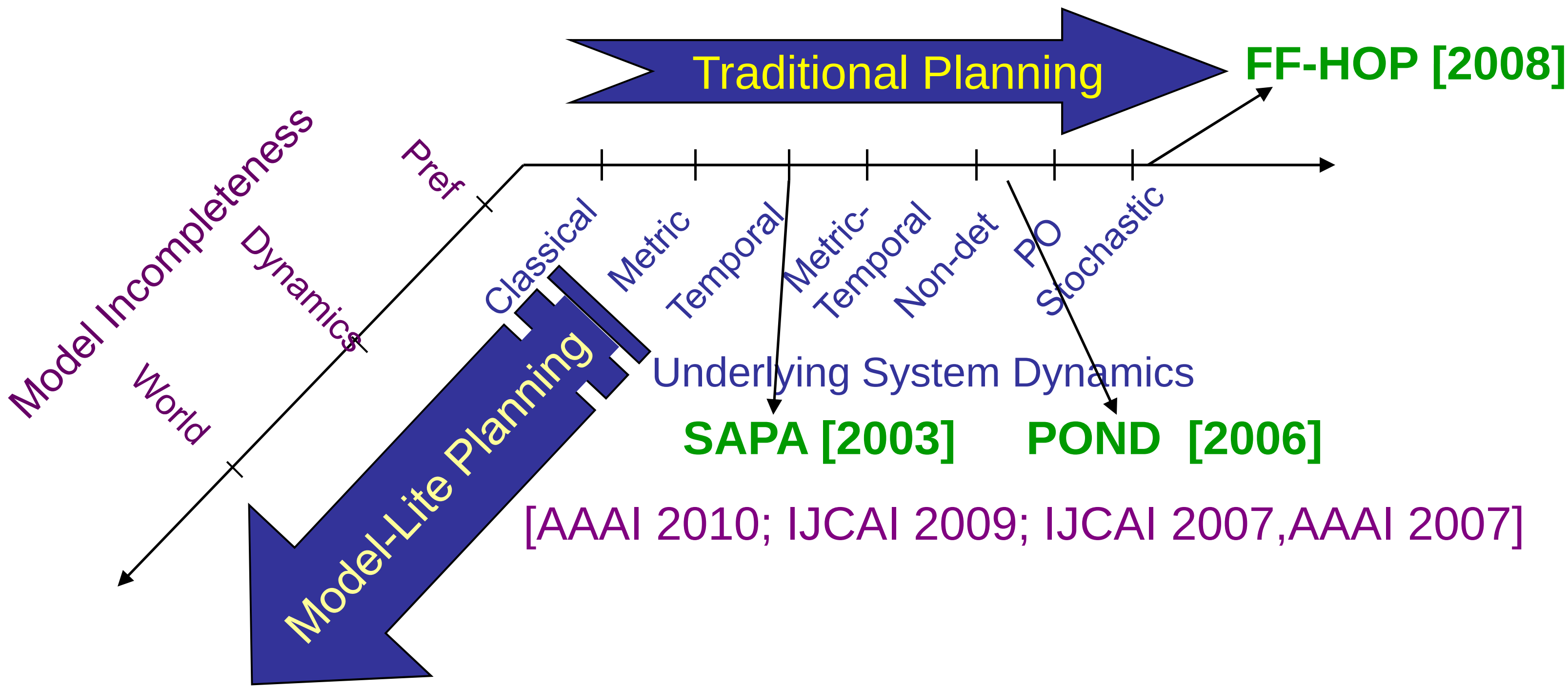
On Going Beyond
Pure Inference
Over Complete Models

Assumption: Complete Models

- →~~Complete~~ Action Descriptions (fallible domain writers)
- →~~Fully Specified~~ Preferences (indecisive users)
- →All ~~objects~~ in the world known up front (open worlds)
- →~~One-shot~~ planning (continual revision)

**Planning is no longer a pure inference problem ☹**

☹ But humans in the loop can ruin a really a perfect day ☹




Effective ways to handle the more expressive planning problems by exploiting the deterministic planning technology

# Model-lite Planning

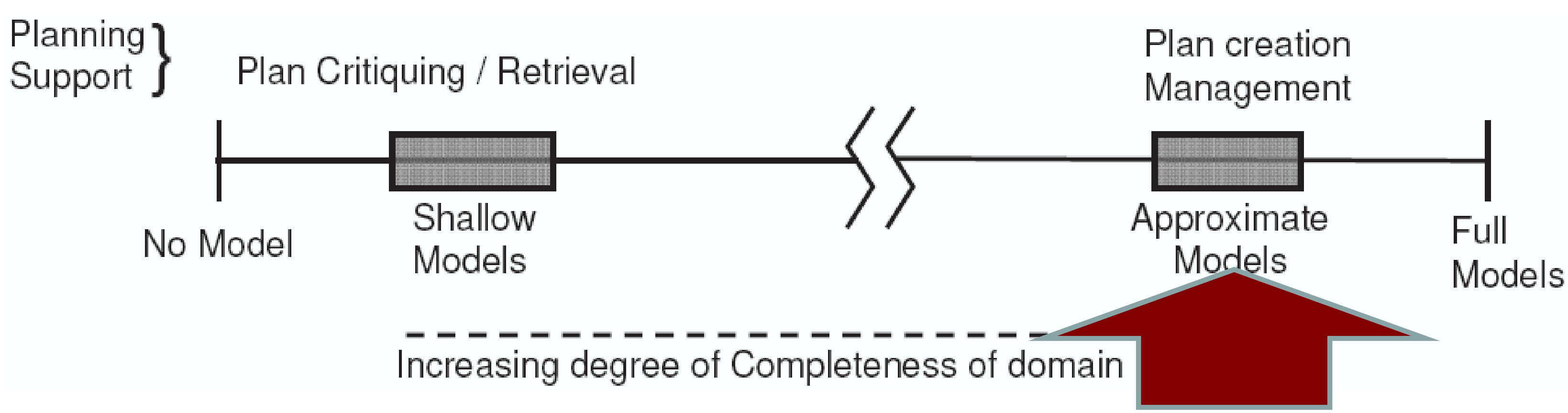
- We need (frame)work for planning that can get by with *incomplete* and *evolving* domain models.
  – I want to convince you that there are interesting research challenges in doing this.
- Disclaimers
  – I am not arguing against model-intensive planning
    - We won’t push NASA to send a Rover up to Mars without doing our best to get as good a model as possible

# Model-lite is Back to the Future

- Interest in model-lite planning is quite old (but has been subverted..)
  - Originally, HTN planning (a la NOAH) was supposed to allow incomplete models of lower-level actions..
  - Originally, Case-based planning was supposed to be a theory of slapping together plans without knowing their full causal models

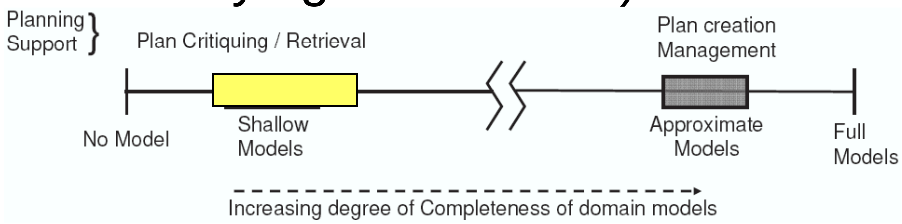
# Model-Lite Planning is
# Planning with incomplete models

- .."incomplete" ➔ "not enough domain knowledge to verify correctness/optimality"
- How *incomplete* is incomplete?
  - Knowing no more than I/O types?
  - Missing a couple of preconditions/effects?



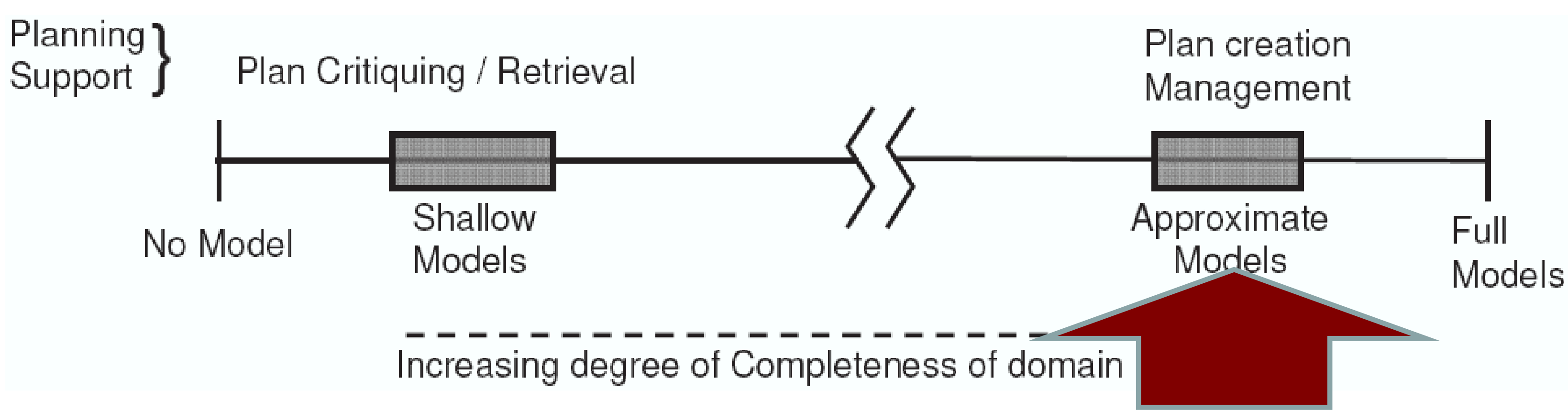
# Challenge: Planning Support for Shallow Domain Models

- Provide planning support that exploits the shallow model available
- Idea: Explore wider variety of domain knowledge that can either be easily specified interactively or learned/mined. E.g.
    - I/O type specifications (e.g. Woogle)
    - Task Dependencies (e.g. workflow specifications)
  - Qn: Can these be compiled down to a common substrate?
- Types of planning support that can be provided with such knowledge
  - Critiquing plans in mixed-initiative scenarios
  - Detecting incorrectness (as against verifying correctness)



# Model-Lite Planning is
# Planning with incomplete models

- ..“incomplete” ➔ “not enough domain knowledge to verify correctness/optimality”
- How *incomplete* is incomplete?
  - Knowing no more than I/O types?
  - Missing a couple of preconditions/effects?



# Challenges of Model-Lite Planning (Approximate Domain Models)

1. Circumscribing the incompleteness

2. Developing the appropriate solution concepts

3. Developing planners capable of synthesizing them

4. Life-long Planning/Learning to reduce incompleteness
   – Commitment-sensitive Replanning

There are known knowns; there are things we know that we know. There are known unknowns; that is to say, there are things that we now know we don't know. But there are also unknown unknowns; there are things we do not know we don't know.

# Challenges of Model-Lite Planning

1. Circumscribing the incompleteness

2. Developing the appropriate solution concepts

3. Developing planners capable of synthesizing them

4. Life-long Planning/Learning to reduce incompleteness
   - Commitment-sensitive Replanning

Tough Problems

# Our Contributions

- Preference incompleteness
  - Unknown Preferences [IJCAI 2007]
  - Partially known Preferences [IJCAI 2009]
- Model incompleteness
  - Robust plan generation [ICAPS Wkshp 2010]
- World/Object incompleteness
  - OWQG [IROS 2009; BTAMP 2009; AAAI 2010]

Model-Lite Planning

# Preferences in Planning – Traditional View

- Classical Model: "Closed world" assumption about user preferences.
  - All preferences assumed to be fully specified/available

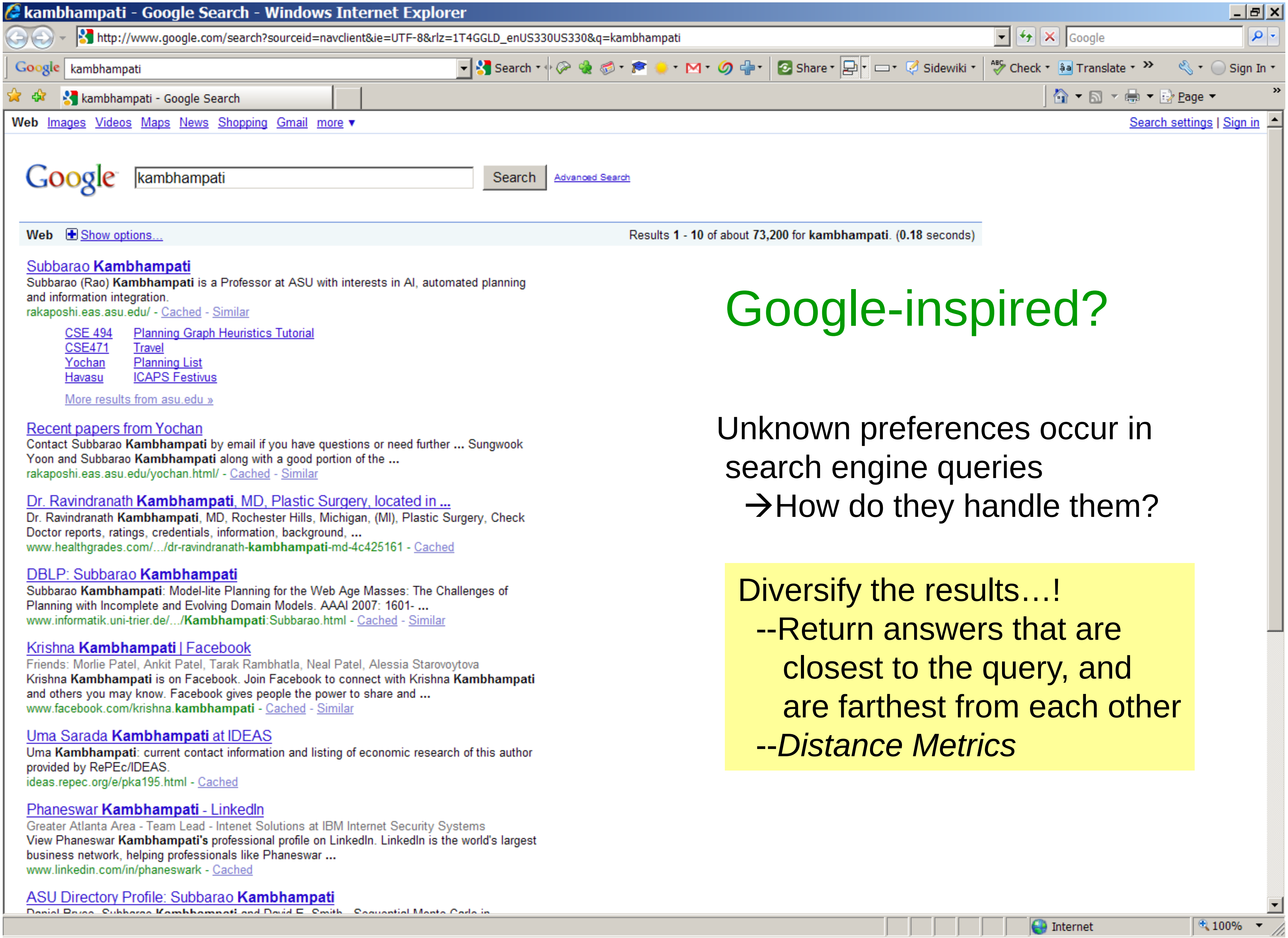
Full Knowledge of Preferences

**Two possibilities**

- If no preferences specified —then user is assumed to be *indifferent*. Any single feasible plan considered acceptable.
- If preferences/objectives are specified, find a plan that is optimal w.r.t. specified objectives.

**Either way, solution is a *single* plan**

# Human in the Loop: Unknown & Partially Known Preferences




# Google-inspired?

Unknown preferences occur in search engine queries
→How do they handle them?

Diversify the results…!
--Return answers that are closest to the query, and are farthest from each other
--*Distance Metrics*

# Handling Unknown & Partially Known Preferences

- **Unknown preferences**
  - For all we know, user may care about every thing -- the flight carrier, the arrival and departure times, the type of flight, the airport, time of travel and cost of travel…
  - Best choice is to return a *diverse* set of plans [IJCAI 2007]
    - Distance measures between plans

# Generating Diverse Plans

- Formalized notions of bases for plan distance measures
- Proposed adaptation to existing representative, state-of-the-art, planning algorithms to search for diverse plans
  - Showed that using action-based distance results in plans that are likely to be also diverse with respect to behavior and causal structure
  - LPG can scale-up well to large problems with the proposed changes

[IJCAI 2007]

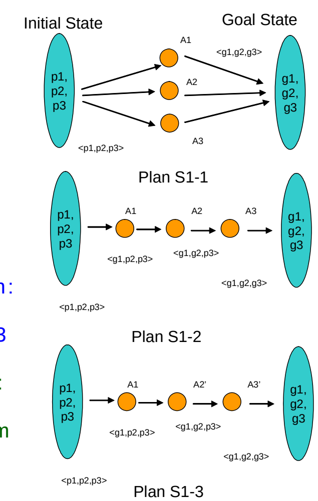
- *d*DISTANT*k*SET
  - Given a distance measure δ(.,.), and a parameter k, find k plans for solving the problem that have guaranteed minimum pair-wise distance *d* among them in terms of δ(.,.)

## Distance Measures

- In what terms should we measure distances between two plans?
  - The actions that are used in the plan?
  - The behaviors exhibited by the plans?
  - The roles played by the actions in the plan?
- Choice may depend on
  - The ultimate use of the plans
    - E.g. Should a plan P and a non-minimal variant of P be considered similar or different?
  - What is the source of plans and how much is accessible?
    - E.g. do we have access to domain theory or just action names?

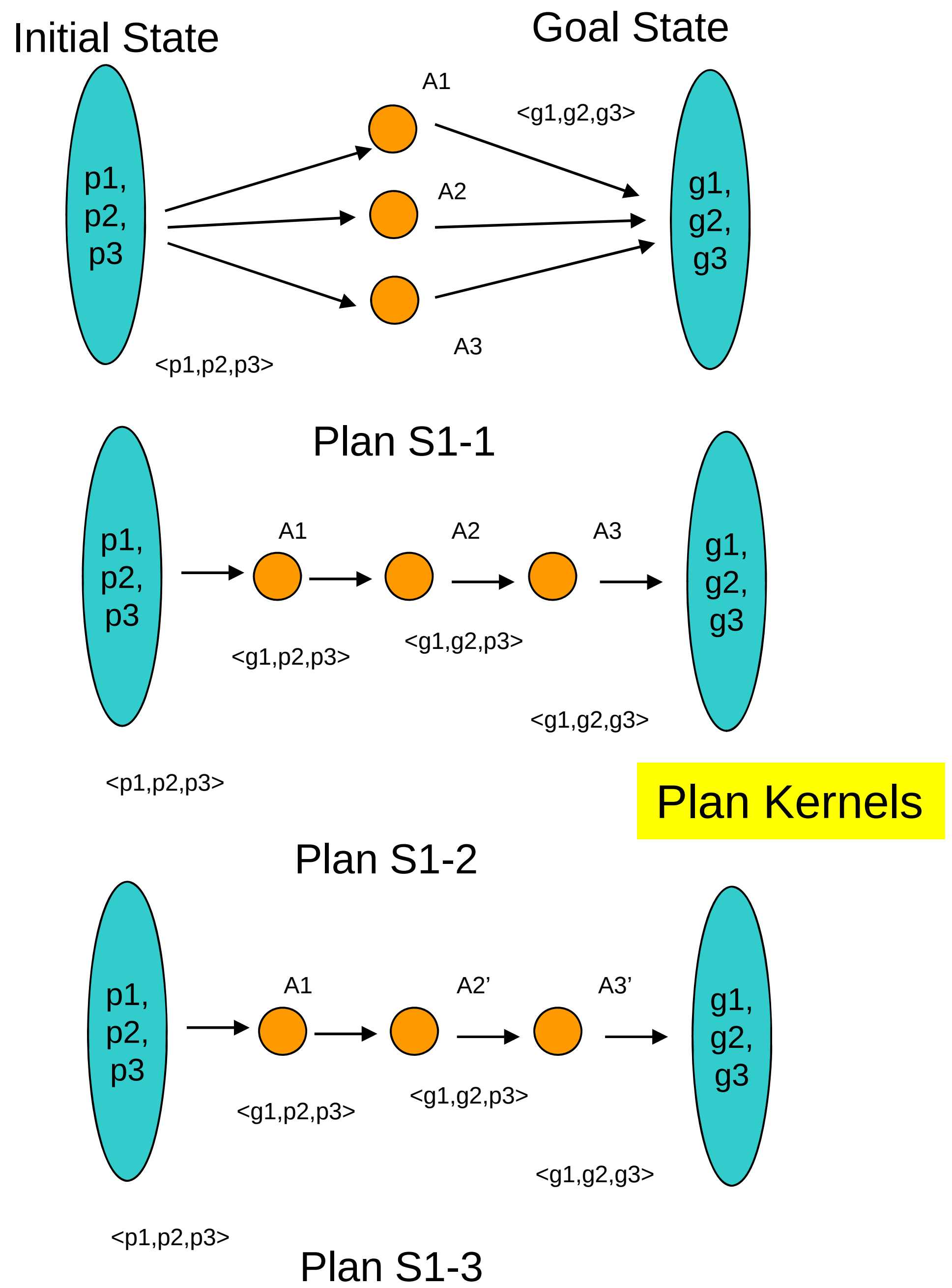
*Compute by Set-difference*

•Action-based comparison: S1-1, S1-2 are similar, both dissimilar to S1-3; with another basis for computation, all can be seen as different
•State-based comparison: S1-1 different from S1-2 and S1-3; S1-2 and S1-3 are similar
•Causal-link comparison: S1-1 and S1-2 are similar, both diverse from S1-3



*Compute by Set-difference*

- Action-based comparison: S1-1, S1-2 are similar, both dissimilar to S1-3; with another basis for computation, all can be seen as different
- State-based comparison: S1-1 different from S1-2 and S1-3; S1-2 and S1-3 are similar
- Causal-link comparison: S1-1 and S1-2 are similar, both diverse from S1-3



# Solution Approaches

- Possible approaches
  - [Parallel] Search simultaneously for k solutions which are bounded by given distance d
  - [Greedy] Search solutions one after another with each solution constraining subsequent search

- Explored in
  - CSP-based GP-CSP classical planner
    - Relative ease of enforcing diversity with different bases for distance functions
  - Heuristic-based LPG metric-temporal planner
    - Scalability of proposed solutions

# Exploring with LPG

$$\delta_a(S_i, S_j) = \frac{|S_i - S_j|}{|S_i| + |S_j|} + \frac{|S_j - S_i|}{|S_i| + |S_j|}.$$

$$\delta_a^A \geq d/\gamma \text{ and } \delta_a^B \geq d/\gamma$$

- Details of changes to LPG in the paper
- Looking for:
  - How large a problem can be solved easily
    - Large sets of diverse plans in complex domains can be found relatively easily
  - Impact of $\chi$
    - $\chi$ = 3 gives better results
  - Can randomization mechanisms in LPG give better result?
    - Distance measure needed to get diversity effectively

# Generating Diverse Plans with Local Search

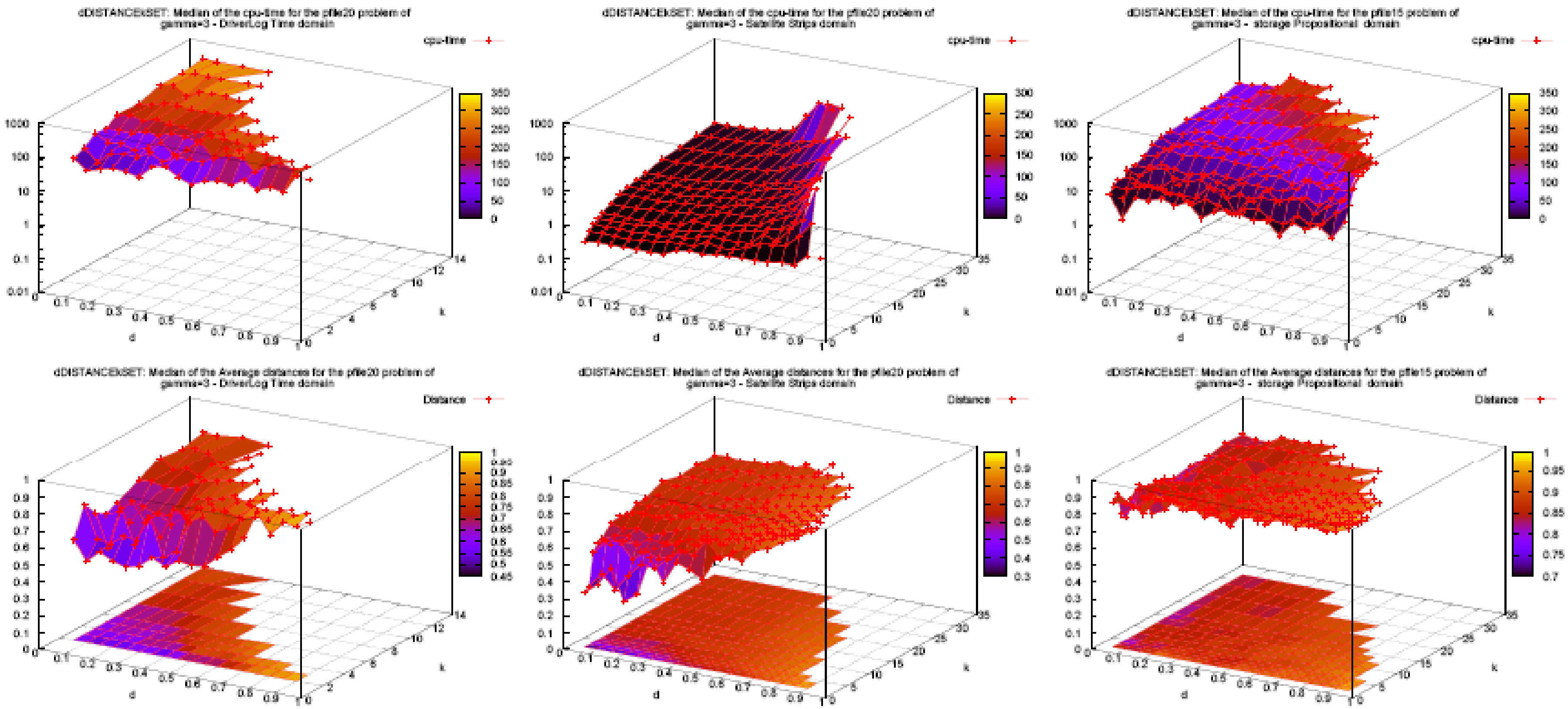



Figure 2: Performance of LPG-d (CPU-time and plan distance) for there problems in DriverLog-Time, Satellite-Strips and Storage-Propositional.

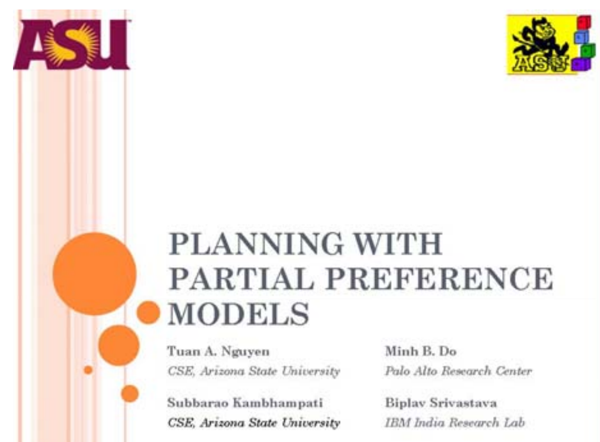
LPG-d solves 109 comb.
Avg. time = 162.8 sec
Avg. distance = 0.68
Includes d<0.4,k=10; d=0.95,k=2

LPG-d solves 211 comb.
Avg. time = 12.1 sec
Avg. distance = 0.69

LPG-d solves 225 comb.
Avg. time = 64.1 sec
Avg. distance = 0.88

# Unknown & Partially Known Preferences

- **Partially known**
  - We may know that user cares only about makespan and cost. But we don't know how she combines them..
- Returning a diverse set of plans may not be enough
  - *They may not differ on the attributes of relevance..*
- Focus on spanning the pareto set..



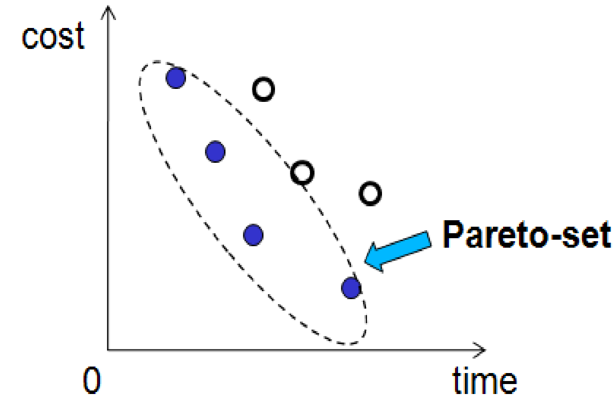
# Modeling Partially Known Objectives

- The user is interested in minimizing two objectives (say makespan and execution cost of plan *p*: *time*(*p*), *cost*(*p*).)
- The quality of plan *p* is given by *cost function*:

$$f(p, w) = w \times time(p) + (1 - w) \times cost(p) \quad (w \in [0,1])$$

  - $w \in [0,1]$ represents the trade-off between two competing objectives.

# Handling Partially Known Preferences

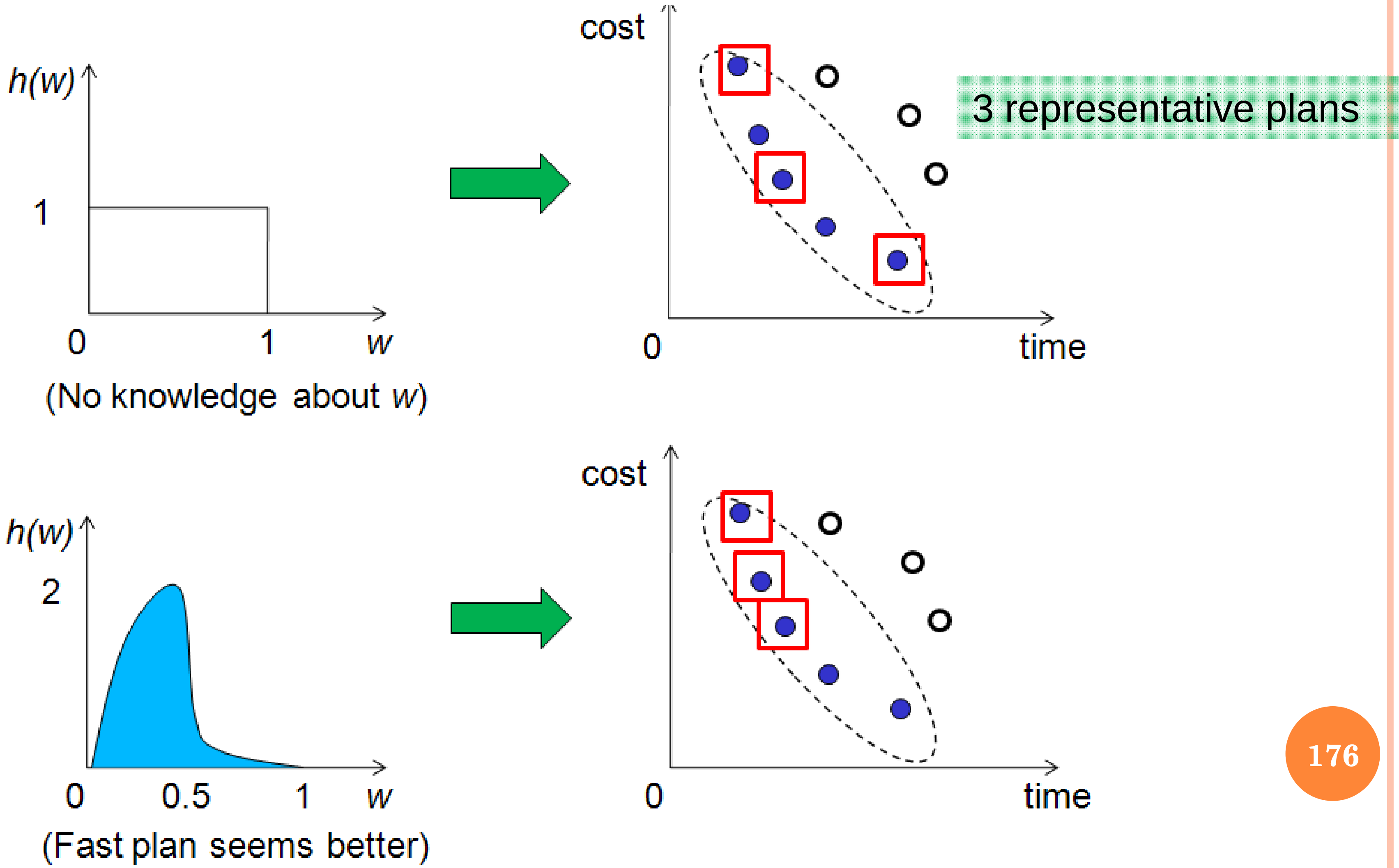



- **View it as a Multi-objective optimization**
  - Return the Pareto optimal set of plans
    (and let the user select from among them)
- Two problems
  - [Computational] Computing the full pareto set can be too costly
  - [Comprehensional] Lay users may suffer information overload when presented with a large set of plans to choose from
- Solution: Return *k* representative plans from the Pareto Set
  - Challenge 1: How to define "representative" robustly?
  - Challenge 2: How to generate representative set of plans efficiently?

# Measuring Representativeness: ICP

$$f(p,w) = w \times time(p) + (1-w) \times cost(p) \quad (w \in [0,1])$$

$$ICP(\mathcal{P}) = \sum_{i=1}^{k} \int_{w_{i-1}}^{w_i} h(w)(w \times t_{p_i} + (1-w) \times c_{p_i})dw$$



# Measuring Representativeness: ICP

- Set of plans $P = \{ p_1, p_2, \ldots, p_k \}$
  - Makespan and execution cost of plan $p_i : t_{p_i}, c_{p_i}$
  - Each plan $p_i$ gives the best cost for all $w \in [w_{i-1}, w_i]$

$$p_i = \arg\min_{p \in P} \{ f(p, w_j) \mid j = 0, 1, \ldots, k \}$$

- The belief distribution of $w$, $h(w)$.
- The expected value of plan set $P$

$$ICP(\mathcal{P}) = \sum_{i=1}^{k} \int_{w_{i-1}}^{w_i} h(w)(w \times t_{p_i} + (1-w) \times c_{p_i}) dw$$

# Handling Partial Preferences using ICP

Problem Statement:

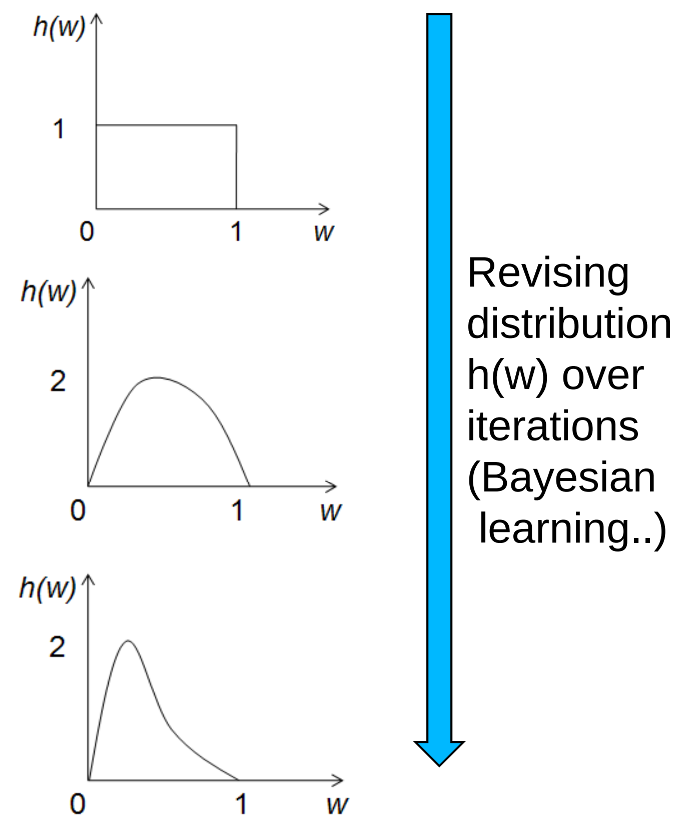
- Given
  - the objectives $O_i$,
  - the vector $w$ for convex combination of $O_i$
  - the distribution $h(w)$ of $w$,
- Return a set of $k$ plans with the minimum ICP value.

- Solution Approaches:
  - Sampling: Sample $k$ values of $w$, and approximate the optimal plan for each value.
  - ICP-Sequential: Drive the search to find plans that will improve ICP
  - Hybrid: Start with Sampling, and then improve the seed set with ICP-Sequential
  - [Baseline]: Find $k$ diverse plans using the distance measures from [IJCAI 2007] paper; LPG-Speed.

# Summary of Incomplete Preferences

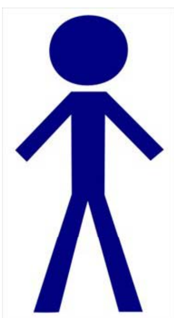
- Unrealistic to assume complete knowledge of user preferences
  - Our previous work [IJCAI 2007] considered the case where *no knowledge* is available. This paper focuses on cases where *partial knowledge* is available
- For ease of computation and comprehension, we need the ability to generate a representative set of plans from the pareto set
  - ICP measure to capture representativeness
  - A spectrum of approaches for generating plan sets with good ICP measure

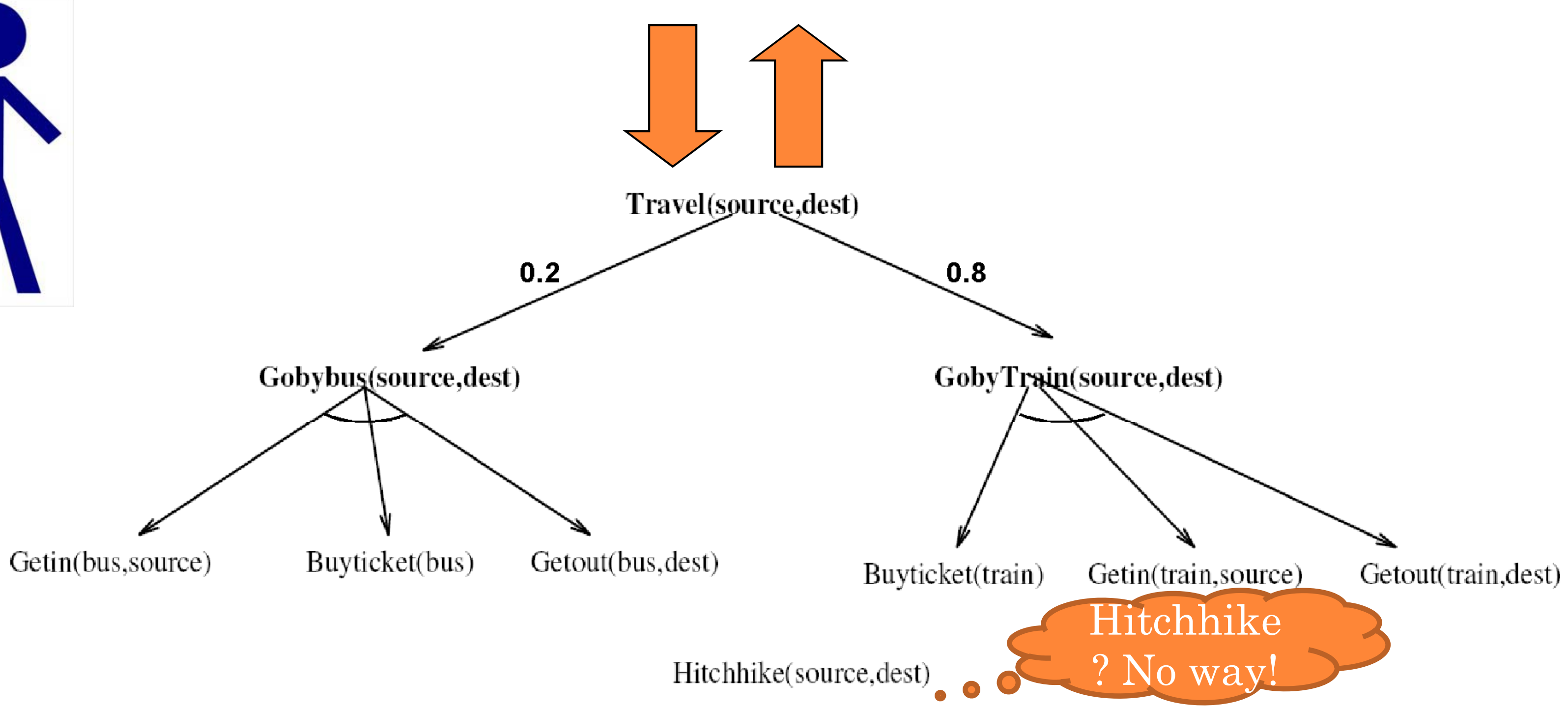
- We can learn to improve the preference model by revising the *h(w)* after every few iterations (through user interaction)



# LEARNING PLAN PREFERENCES
## From Observed Executions

- $P_{bus}$: Getin(bus, source), Buyticket(bus), Getout(bus, dest) 2
- $P_{train}$: Buyticket(train), Getin(train, source), Getout(train, dest) 8
- $P_{hike}$: Hitchhike(source, dest) 0






[IJCAI 2009]

# LEARNING USER PLAN PREFERENCES OBFUSCATED BY FEASIBILITY CONSTRAINTS

- Rescale observed plans
  - Undo the filtering caused by feasibility constraints
- Base learner
  - Acquires true user preferences based on adjusted plan frequencies

Input Plans:

$P_{plane}$ * 3

$P_{train}$ * 5

$P_{bus}$ * 6

Rescaled Plans:

$P_{plane}$ * 12

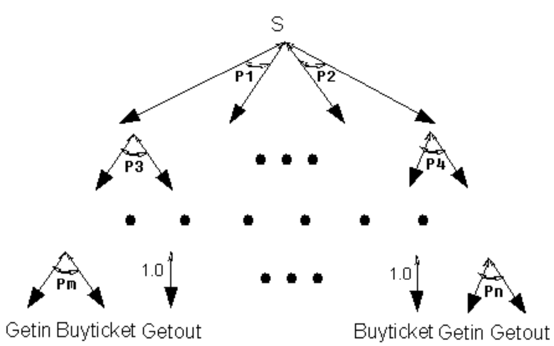
$P_{train}$ * 4

$P_{bus}$ * 1

***Base Learner***

*IJCAI '09*

**User Preference Model**




[ICAPS 2009]

# Our Contributions

## Preference incompleteness

Unknown Preferences [IJCAI 2007]

Partially known  Preferences [IJCAI 2009]

## Model incompleteness

Robust plan generation [ICAPS Wkshp 2010]

## World/Object incompleteness

OWQG [IROS 2009; BTAMP 2009; AAAI 2010]

There are known knowns; there are things we know that we know. There are known unknowns; that is to say, there are things that we now know we don't know. But there are also unknown unknowns; there are things we do not know we don't know.

# Planning with partial domain models: Motivation

- Planning, in traditional perspective, assumes a completely specified domain model
  - We know exactly the conditions and effects of action execution
  - Stochastic models also assume completeness ("known" probabilities)

```
(:action pick-up
        :parameters (?ob1)
        :precondition (and (clear ?ob1)
                           (on-table ?ob1)
                           (arm-empty)
                           (block ?ob1))
        :effect
        (and (not (on-table ?ob1))
             (not (clear ?ob1))
             (not (arm-empty))
             (holding ?ob1)))
```

- Domain modeling is a laborious, error-prone task
  - So much so that there is a Knowledge Engineering track for ICP
- Action descriptions have to be seen as "nominal"
  - May have missing preconditions and effects…
- Sometimes, the domain modeler may be able to annotate the action with sources of incompleteness
  - Possible preconditions/effects

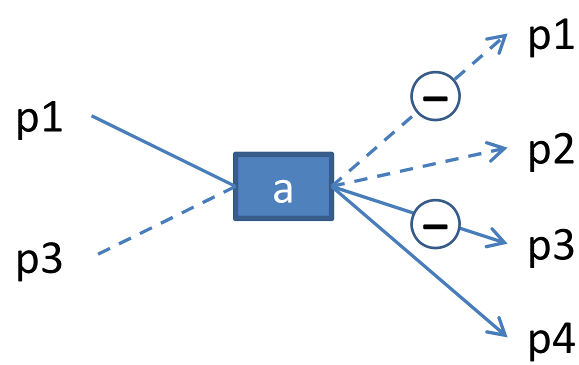
Can the planner exploit such partial knowledge?

# Deterministic Partial Domain Models

- We consider planning with deterministic, but incompletely specified domain model

- Each action **a** is associated with *possible* precond and effects (in addition to the normal precond/eff):
  - **PreP(a)** ***[p]:*** set of propositions that **a** *might* depend on during execution
  - **AddP(a)** ***[p]:*** : set of propositions that **a** *might* add after execution
  - **DelP(a)** ***[p]:*** : set of propositions that **a** *might* delete after execution

Example: An action **a** that is known to depend on **p1**, add p4 and delete **p3**. In addition, it might have **p3** as its precondition, might add **p2** and might delete **p1** after execution.



# More on Annotations

- We will focus on how to handle the possible precond/effect annotations on the ground actions…
- But they are more likely specified at the "schema" level
  - All groundings of an action schema will thus have the same possible preconds/effects
  - We can support "friendly" syntax to specify exceptions
    - E.g. that the annotations hold only for specific variable bindings

# Solution Concept: Robust Plans

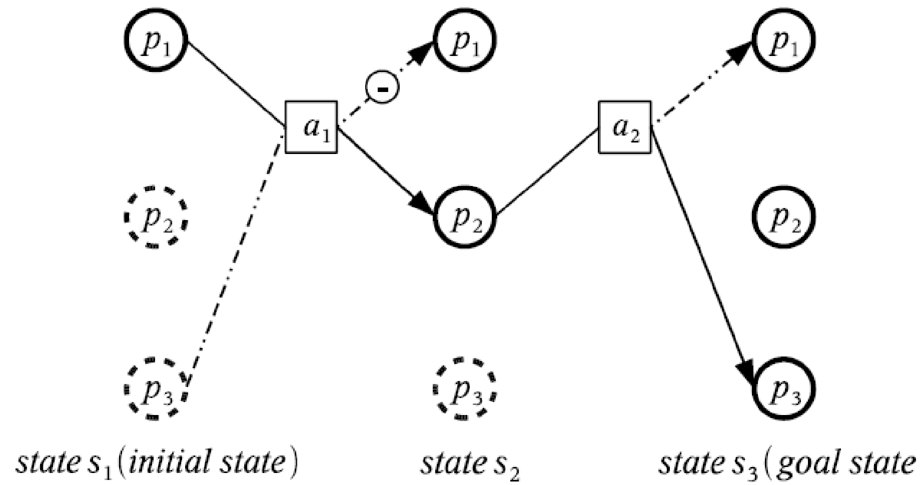
- Solution concept:
  - Robust plan
  - Plan is highly robust if executable in large number of most-likely candidate models
- Robustness measure
  - Set of candidate domain models **S** (consistent with the given deterministic partial domain model **D**)
  - A complete but unknown domain model **D***
    - Can be any model in **S**

$$R(\pi) = \frac{|\Pi|}{2^K}$$

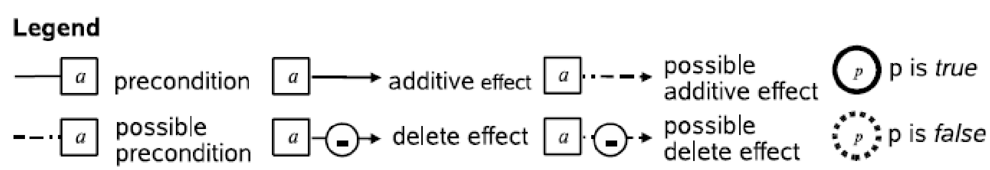
$|\Pi|$ Number of candidate models with which the plan succeeds

$$K = \sum_a \mathrm{PreP}(a) + \mathrm{AddP}(a) + \mathrm{DelP}(a)$$

Easily generalized to consider model likelihood



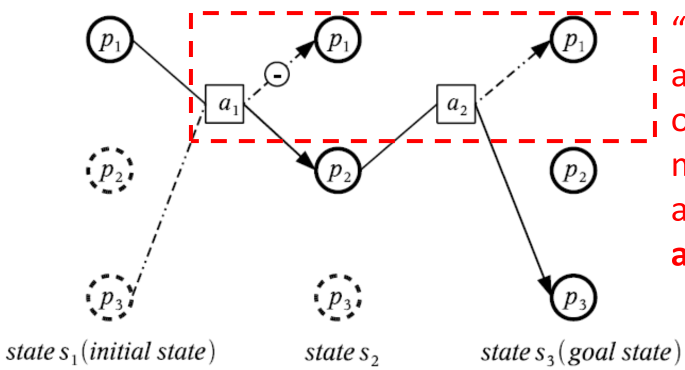

| Candidate models of plan | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $a_1$ **relies on** $p_3$ | yes | yes | yes | yes | no | no | no | no |
| $a_1$ **deletes** $p_1$ | yes | yes | no | no | yes | yes | no | no |
| $a_2$ **adds** $p_2$ | yes | no | yes | no | yes | no | yes | no |
| **Plan status** | fail | fail | fail | fail | succeed | fail | succeed | succeed |




Robustness value: 3/8

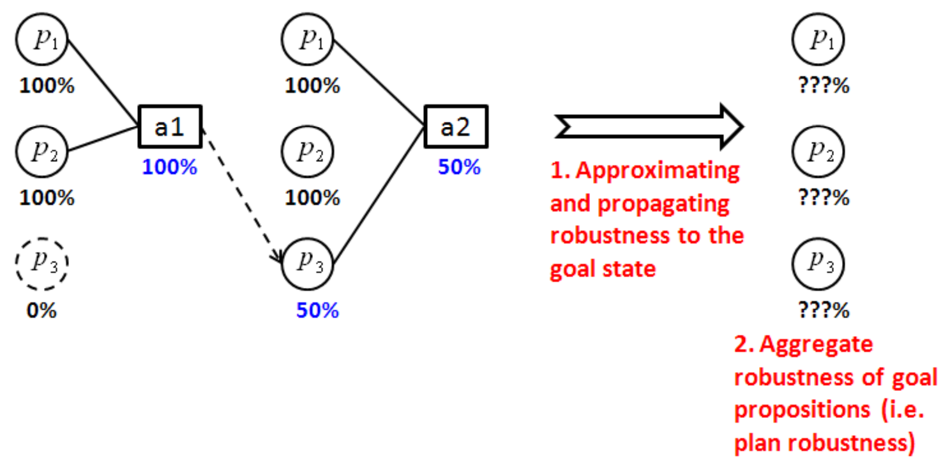
# Assessing Plan Robustness

- Number of candidate models: exponentially large. Computing robustness of a given plan is hard!!!
  - Exact and approximate assessment.
- **Exact methods:**
  - (Weighted) Model-counting approach:
    - Construct logical formulas representing *causal-proof* (Mali & Kambhampati 1999) for plan correctness
    - Invoke an exact model counting approach




"If **p1** is realized as a delete effect of **a1**, then it must be an additive effect of **a2**."

- **Approximate methods:**
  - Invoke *approximate* model counting approach
  - Approximate and propagate action robustness
    - Can be used in generating robust plans



# Generating Robust Plans

- Compilation approach: Compile into a *(Probabilistic) Conformant Planning* problem
  - One “unobservable” variable per each possible effect/precondition
    - Significant initial state uncertainty
  - Can adapt a probabilistic conformant planner such as POND [JAIR, 2006; AIJ 2008]
- Direct approach: Bias a planner’s search towards more robust plans
  - Heuristically assess the robustness of partial plans
    - Need to use the (approximate) robustness assessment procedures

[See work by Weber & Bryce, 2011]

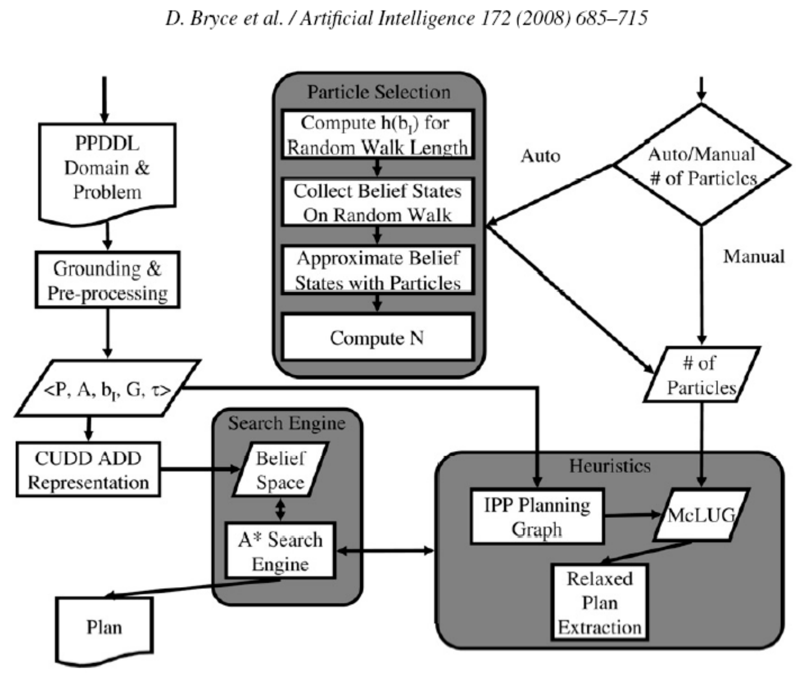
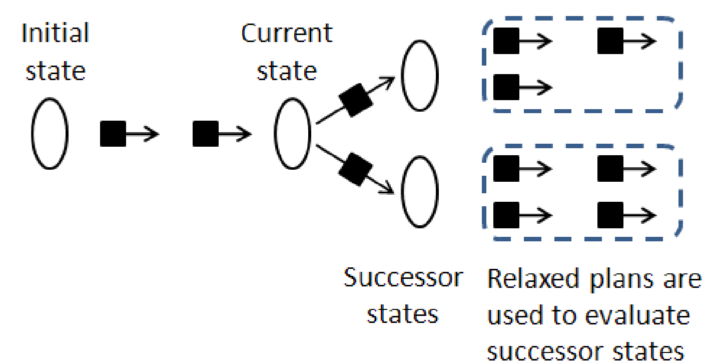



Fig. 6. *POND* architecture.

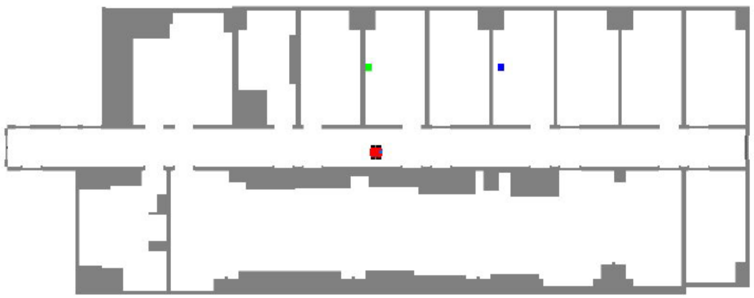



[Workshops of ICAPS 2010; AAAI 2011]

# Our Contributions

- Preference incompleteness
  - Unknown Preferences [IJCAI 2007]
  - Partially known  Preferences [IJCAI 2009]
- Model incompleteness
  - Robust plan generation [ICAPS Wkshp 2010]
- World/Object incompleteness
  - OWQG [IROS 2009; BTAMP 2009; AAAI 2010]

# Urban Search and Rescue



- Human-Robot team
- Robot starts the beginning of the hallway
- Human is giving higher level knowledge
- Hard Goal: Reach the end of the hallway
- Wounded people are in rooms
- Soft Goal: Report locations of wounded people

# Planning Support for USAR

- Good News: Some aspects of existing planning technology are very relevant
    - Partial Satisfaction
    - Replanning & Execution Monitoring
- Bad News: Incomplete Model / Open World
  – Unknown objects
    - Don't know where injured people are
  – Goals specified in terms of them
    - If the robot finds an injured person, it should report their location …

How do you make a deterministic closed-world planner believe in opportunities sans guarantees?

Open World Quantified Goals

Partial Satisfaction Planning (PSP)

Sensing and Replanning

Planner

**CLOSED WORLD**


Robot

**OPEN WORLD**

**Under Sensing**
Closed World Model

**Limited Sensing**
Planner guides robot in a limited way

**Over Sensing**
Robot senses its way through the world

# Handling Open World

- Extreme Cases
  - If the robot assumes "closed world", it will just go to the end of the corridor.
  - If the robot insists on "closing" the model before doing planning, it will do over-sensing.
- Need a way of combining sensing and planning
  - Information on unknown objects
  - Goals conditioned on these objects

# Open World Quantified Goals (OWQGs)

- Goals that allow for the specification of additional information
  - To take advantage of opportunities

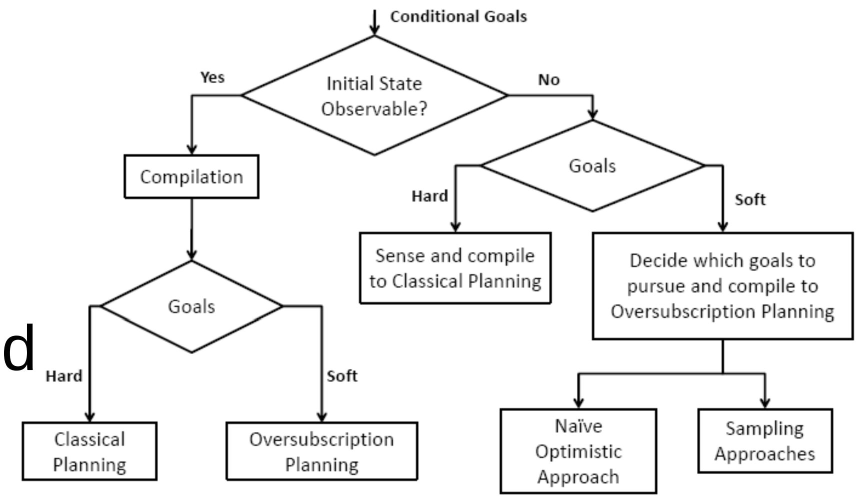
```
(:open  (forall ?r - room         Quantified Object(s)
   (sense ?p - person             Sensed Object
      (looked_for ?p ?r)          Closure Condition
     (and (has_property ?p wounded)
          (in ?p ?r))             Quantified Facts
(:goal
   (and (reported ?p wounded ?r)
   [100] - soft))))               Quantified Goal
```

# OWQGs as Conditional Rewards

Robot needs to
  sense wounded people
  before reporting them

Planner has to deal with open world

Naïve idea: Ask Robot to look
  everywhere (high sensing cost)




Conditional Goals can be compiled
  down when the world model is complete

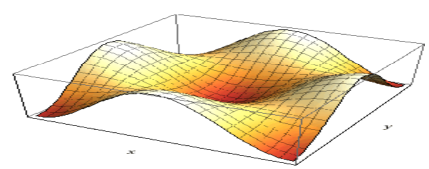
--Need to sense for those conditional goals
  whose antecedents are likely to hold

$$\hat{\mathcal{G}}_c = \arg\max_{\hat{\mathcal{G}}_c^i \subseteq \mathcal{G}_c} \mathbf{E}_{\mathbf{P}\sim\Psi}\mathcal{B}(G_o \cup [\mathcal{G}_c^i \setminus \mathbf{P}]) - \mathcal{S}(\mathcal{G}_c^i)$$

[AAAI, 2010; IROS 2009; BTAMP 2009]

# Planning with OWQGs

- Bias the planner's model
- Endow the planner with an **optimistic view**
  - Assume existence of objects and facts that may lead to rewarding goals
    - e.g. the presence of an injured human in a room
  - Create **runtime objects**
  - Add to the planner's database of ground objects
- Plans are generated over this reconfigured **potential** search space



# Replanning and Execution Monitoring

- Sensing is expensive …
  - Cannot be done at every step
- Planner needs to direct the architecture on:
  - when to sense
  - what to sense for
- Planning to sense in a goal-directed manner
  - Output all actions up to (and including) any action that results in "closing" the world
    - Obtaining information about unknown objects

# Challenges of Model-Lite Planning

1. Circumscribing the incompleteness

2. Developing the appropriate solution concepts

3. Developing planners capable of synthesizing them

4. Life Long Planing/Learning to reduce incompleteness

# Partial Solutions for Model-Lite Planning

**Can exploit Deterministic Planning technology!**

1. Circumscribing the incompleteness
   - Preference components; possible precond/effect annotations; OWQG
2. Developing the appropriate solution concepts
   - Diverse plans; Robust plans; Partial sensing plans
3. Developing planners capable of synthesizing them
   - Can adapt existing planners toward these solution concepts
4. Life Long Planning/Learning to reduce incompleteness
   - Learning preferences *h(.)* through interactions; learning model conditions through execution
     - [ Tutorial on Learning in Planning AI MAG 2003; Learning preferences as HTNs IJCAI 2009; ICAPS 2009]

**Model-Lite Planning:**
**Planning is more than pure inference over completely specified models!**

# Lecture Overview…

I'd rather learn from one bird how to sing
than teach ten thousand stars
how not to dance

*ee cummings*

- How to use our hammers *wisely*
  - Lessons from
    - *Partial Satisfaction Planning*
    - *Temporal Planning*
    - *Stochastic Planning*
- How to be skeptical of our benchmarks
  - (Lack of) Temporal Benchmarks
  - (Lack of) Relational Benchmarks
- How to go beyond pure inference over complete models: A call for model-lite planning
  - How to handle incomplete domain models?
  - How to handle incomplete preference models?
  - How to handle incomplete object models (open worlds)

## On Using Our Hammers Wisely

*Make things as simple as possible, but not simpler*
*-Attributed to Einstein*

- Classical Planners have justifiably become our hammers… This is mostly GOOD NEWS
  - We want to coax all other planning problems into formats that will allow us to maximally utilize the progress made in scaling up classical planning
  - ..But, we need to be careful, lest we lose the essence of the expressive planning problems during the coaxing (compilation)
  - Some examples..
    - Cost-based Planning (ε-cost trap)
    - Temporal Planning (Required Concurrency)
    - Stochastic Planning (Biased Determinizations)

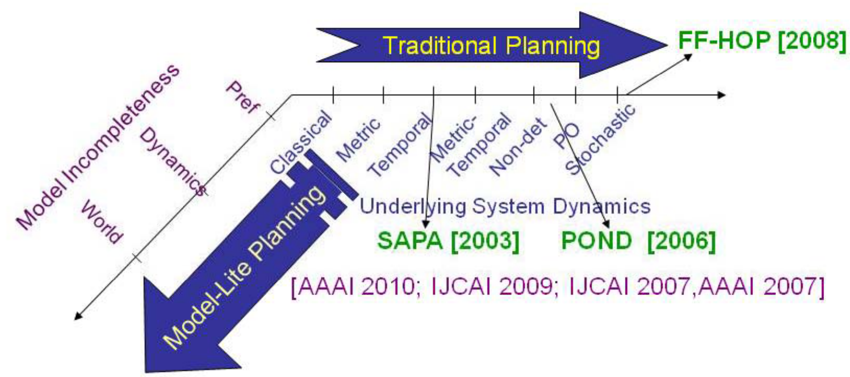


## On Being Skeptical About our Benchmarks

- Progress in planning in the old days was hampered by lack of common benchmarks
  - The arguments of expressiveness with no guarantees of comparative efficiency..
- Thanks to IPC competitions, we have a huge chest of benchmarks.. But they pose their own problems
  - Arguments of efficiency with little heed to expressiveness. Undivided benchmarks can themselves inhibit progress
- Examples
  - Temporal Planning benchmarks indirectly inhibited work on expressive temporal planners
  - Most benchmarks inhibited work on lifted planners

## On Going Beyond Pure Inference Over Complete Models

Assumption: Complete Models
→~~Complete~~ Action Descriptions (fallible domain writers)
→F~~ully Specif~~ied Preferences (indecisive users)
→All ~~objects~~ in the world known up front (open worlds)
→On~~e-shot~~ planning (continual revision)
**Planning is no longer a pure inference problem ☹**
☹ But humans in the loop can ruin a really a perfect day ☹




Effective ways to handle the more expressive planning problems by exploiting the deterministic planning technology


# going BACK TO THE FUTURE of Planning

Imagine there's no Landmarks
It's easy if you try
No benchmarks below us
Above us only blai
Imagine all the planners
Planning for real

Imagine there's no state
It isn't hard to do
Nothing to regress or relax
And no cost guidance too
Imagine all the planners
Lifting all the worlds

You may say that I'm a whiner
But I'm not the only one
I hope someday you'll join us
And the ICAPS will be more fun

Imagine there's no models
I wonder if you can
No need for preferences or groundings
A diversity of plans
Imagine all the planners
Living life incomplete

You may say that I'm a whiner
But I'm not the only one
I hope someday you'll join us
And the ICAPS will be more fun