Planning: Heuristics and CSP Planning

CPSC 322 Lecture 17

Lecture Overview

- **Recap: Planning Representation and Forward algorithm**
- Heuristics
- CSP Planning

Search for Specific R&R systems

Constraint Satisfaction (Problems) *(arc consistency/domain splitting is a bit different)*:

- State: assignments of values to a subset of the variables
- Successor function: assign values to a “free” variable
- Goal test: set of constraints
- Solution: possible world that satisfies the constraints
- Heuristic function: *none (all solutions at the same distance from start)*

Planning :

- State? A. Full assignment B. Partial assignment
- Successor function? A. Actions B. Random assignments
- Goal test? A. Full assignment B. Partial assignment C. It depends
- Solution? A. A goal state B. A path to a goal state
- Heuristic function….

Inference

- State
- Successor function
- Goal test
- Solution
- Heuristic function

Learning Goals for today’s class

You can:

- Construct and justify a **heuristic function** for forward planning.
- Translate a planning problem represented in STRIPS into a corresponding CSP problem (and vice versa)
- Solve a planning problem with CSP by expanding the horizon

Lecture Overview

- Recap: Planning Representation and Forward algorithm
- **Heuristics for forward planning**
- CSP Planning

Heuristics for Forward Planning

Heuristic function: estimate of the **distance** form a state to the goal

In planning this is the **number of actions**

Two simplifications we will need to make in the representation:

- All features are binary: T / F
- Goals and preconditions can only be assignments to T

Also: a subgoal is a particular assignment in the goal e.g., if the goal is <A=T, B=T, C=T> then..

subgoals: <A=T>,<B=T>,<C=T>

Heuristics for Forward Planning

Which makes the most sense as an admissible heuristic for forward planning under STRIPS with these simplifications applied?

A. Number of satisfied subgoals
B. Number of unsatisfied subgoals
C. A + B
D. A - B
E. $O(b^m)$

Note: none of these will actually work

Heuristics for Forward Planning (cont.)

What kind of simplifications of the actions would correspond to using the number of unsatisfied subgoals as a heuristic?

a) Removing all **preconditions**

b) Removing all **negative effects**

c) Assuming no action can achieve **more than one subgoal**

Heuristics for Forward Planning (cont.)

What kind of simplifications of the actions would correspond to using the number of unsatisfied subgoals as a heuristic?

~~a) Removing all **preconditions**~~ ***trivializes problem***

b) Removing all **negative effects**

~~c) Assuming no action can achieve **more than one subgoal**~~ ***renders the heuristic inadmissible***

Heuristics for Forward Planning: empty-delete-list

So, we only relax the problem according to (b)

i.e., we remove all the effects that make a variable ***False***

(also known as **emptying the delete list**)

Action *a* effects (~~B=*F*~~, C=*T*)

But then how do we compute the heuristic?

Solve the simplified planning problem

This is often fast enough to be worthwhile *(even though you may need to do it many times – once for each state)*

Empty-delete-list heuristics with **forward planning** is currently considered a very successful strategy

Final Comment

- You should view **(informed) Forward Planning** as one of the basic planning techniques
- By itself, it cannot go far, but it can work very well in combination with other techniques, for specific domains
  - See, for instance, descriptions of competing planners in the presentation of results for the **2008 planning competition** (posted in the class schedule)

Lecture Overview

- Recap: Planning Representation and Forward algorithm
- Heuristics for forward planning
- **CSP Planning**

Planning as a CSP

- An alternative approach to planning is to set up a planning problem as a **CSP**!

- We simply reformulate a STRIPS model as a set of **variables and constraints**

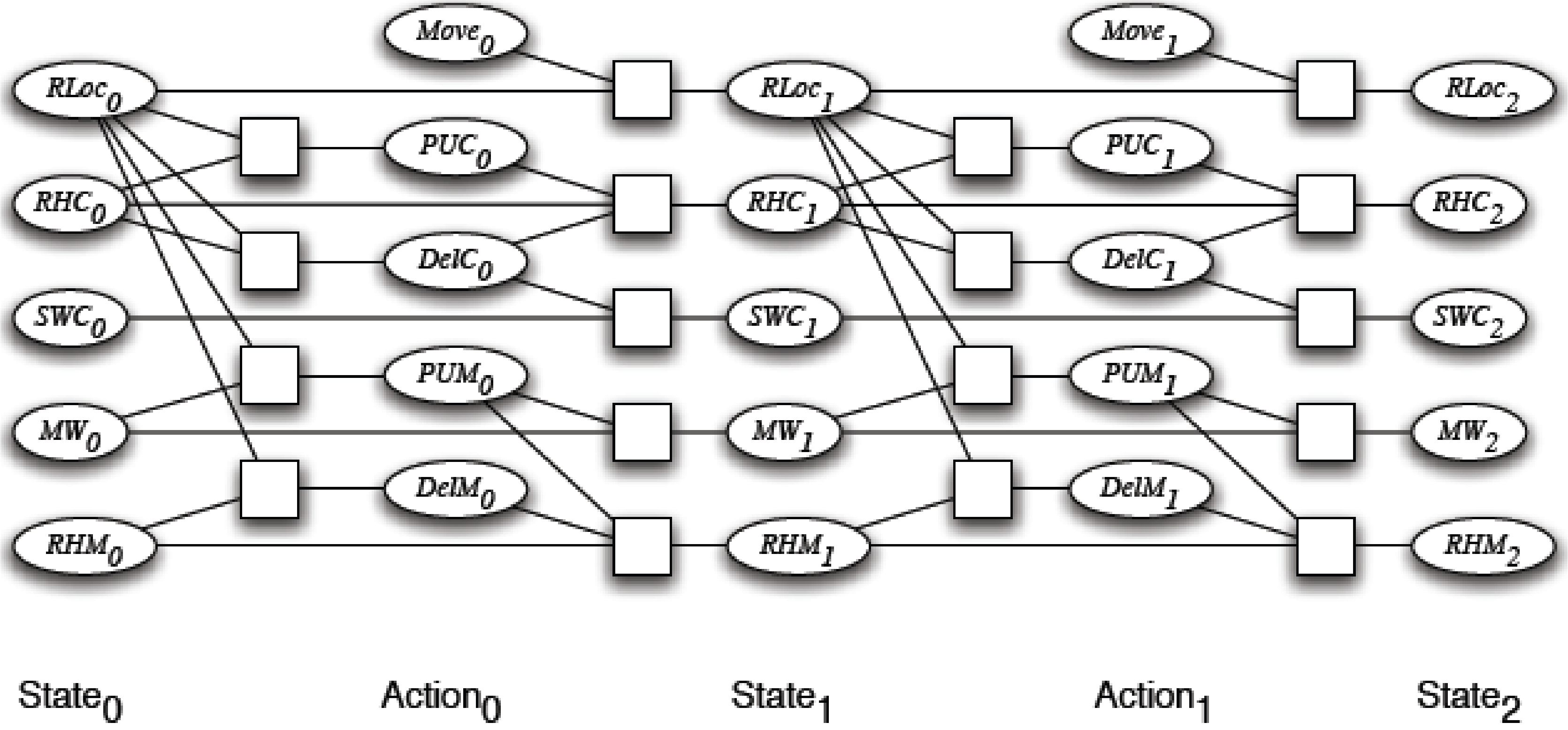
- Once this is done we can even express additional aspects of our problem (as additional constraints)

e.g., see Practice Exercise UBC commuting
"*careAboutEnvironment*" constraint

Planning as a CSP: Overview

- We need to "unroll the plan" for a fixed number of steps: this is called the horizon
- To do this with a horizon of k:
  - construct a CSP variable for each STRIPS variable **(eg. A,B,C)** at each time step from 0 to k
  - construct a boolean CSP variable for each STRIPS action **(eg. a1, a2)** at each time step from 0 to k - 1.
  - construct CSP constraints corresponding to start and goal values, as well as preconditions and effects of actions

CSP Planning: Robot Example



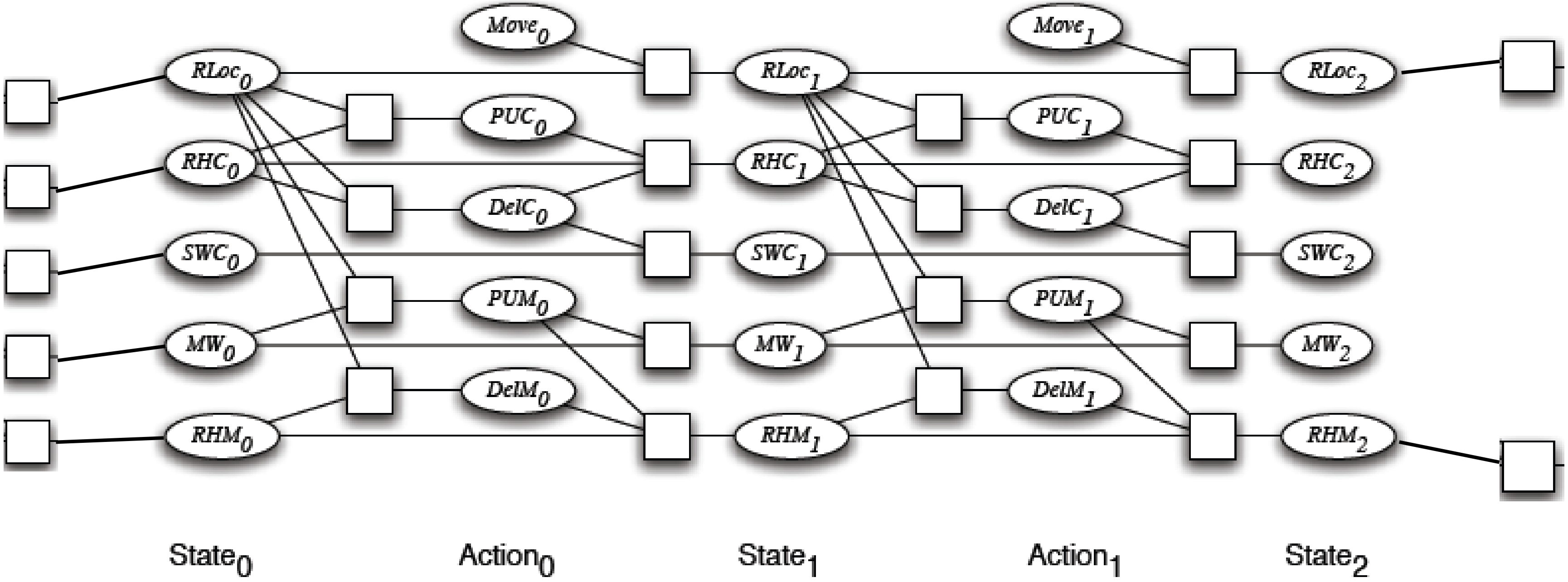
*Variables for actions are **binary***

action occurring (or not) at that step

CSP Planning: Initial and Goal Constraints

usually unary

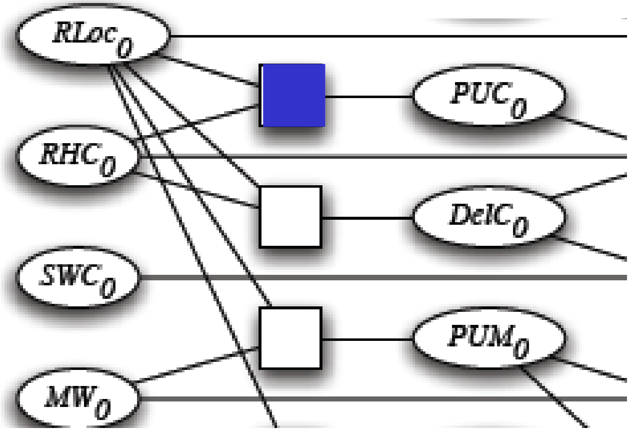
- initial constraints constrain state variables at time 0
- goal constraints constrain state variables at time *k*



CSP Planning: Prec. Constraints

As usual, we have to express the **preconditions** and **effects** of actions:

- precondition constraints
  - hold between state variables at time *t* and action variables at time *t* *(one per action variable)*
  - specify when actions may be taken

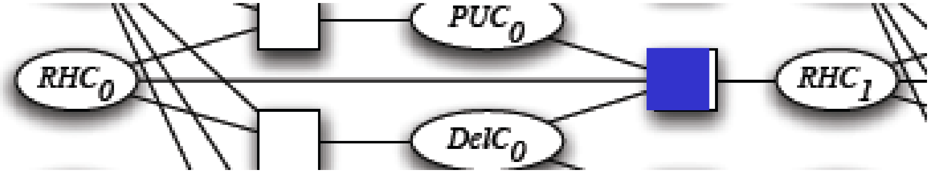


PUC_0

$RLoc_0$	RHC_0	PUC_0
cs	T	F
cs	F	T
cs	F	F
mr	*	F
lab	*	F
off	*	F

CSP Planning: Effect Constraints

- effect constraints
  - between state variables at time *t*, action variables at time t and state variables at time *t*+1 *(one per state variable at time t+1)*
  - explain how a state variable at time *t* + 1 is affected by the action(s) taken at time *t* and by its own value at time *t*



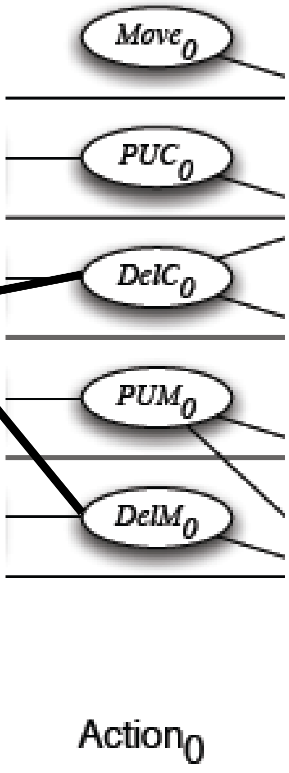
RHC_i	$DelC_i$	PUC_i	RHC_{i+1}
T	T	T	T
T	T	F	F
T	F	T	T
…	…	…	…
…	…	…	…

CSP Planning: Constraints Contd.

Other constraints we may want are action constraints:

- specify which actions **cannot occur simultaneously**
- these are sometimes called mutual exclusion (mutex) constraints

DelM$_i$	DelC$_i$
??	



Which is the best constraint to do achieve this mutual exclusion?

A.

DelM$_i$	DelC$_i$
T	T
T	F
F	T

B.

DelM$_i$	DelC$_i$
T	F
F	T

C.

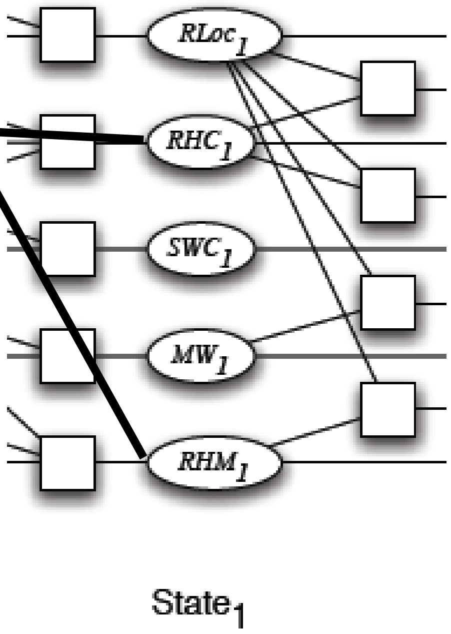
DelM$_i$	DelC$_i$
T	F
F	T
F	F

CSP Planning: Constraints Contd.

Other constraints we may want are state constraints

- hold between variables at the same time step
- they can capture physical constraints of the system (robot cannot hold coffee and mail)
- they can encode maintenance goals

RHC_i	RHM_i
T	F
F	T
F	F

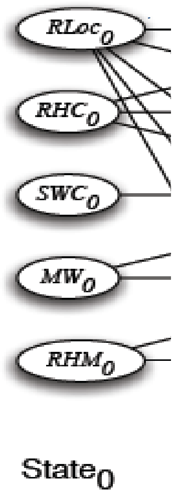


Charge 100% 80% ….
Distance to charger 1 2 3 4 5
Charge – 20% * distance >= 0

CSP Planning: Solving the problem

Map STRIPS Representation for horizon 1, 2, 3, …, until solution found

Run arc consistency and search!



h = 0

Is $State_0$ a goal?

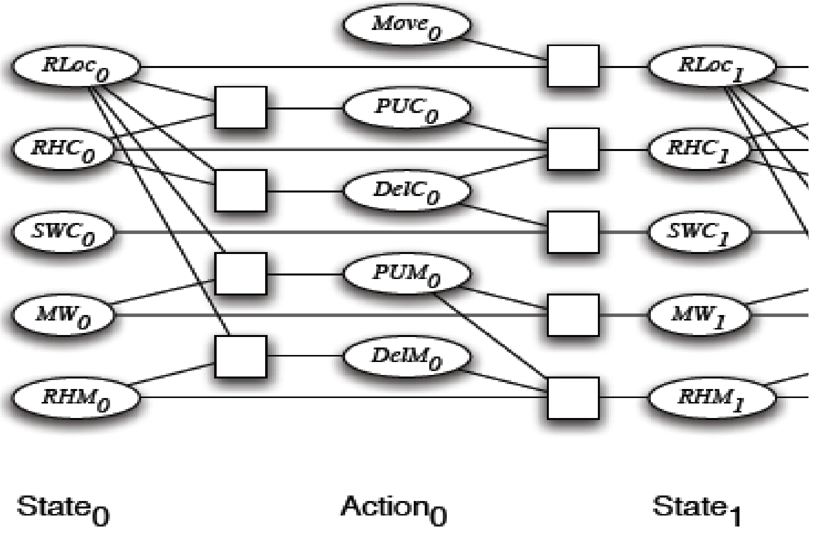
If yes, DONE!

If no,

CSP Planning: Solving the problem

Map STRIPS Representation for horizon h =1

Run arc consistency and search!



h = 1

Is State$_1$ a goal

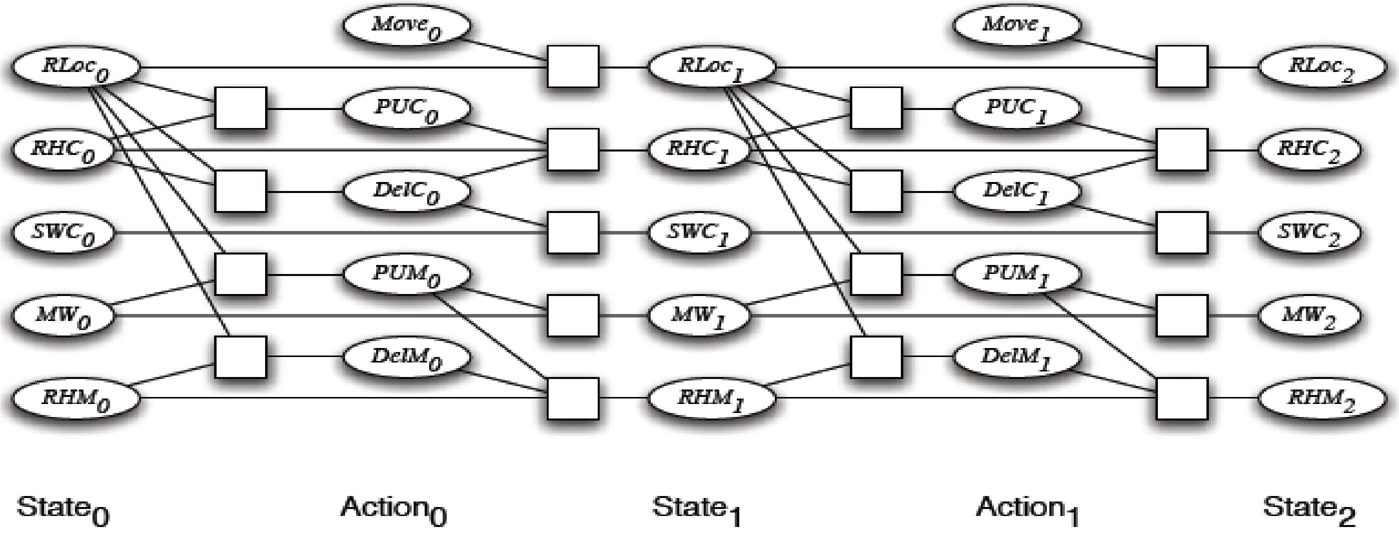
If yes, DONE!

If no,

CSP Planning: Solving the problem

Map STRIPS Representation for horizon h = 2

Run arc consistency, search!



h = 2: Is $State_2$ a goal

If yes, DONE!

If no….continue

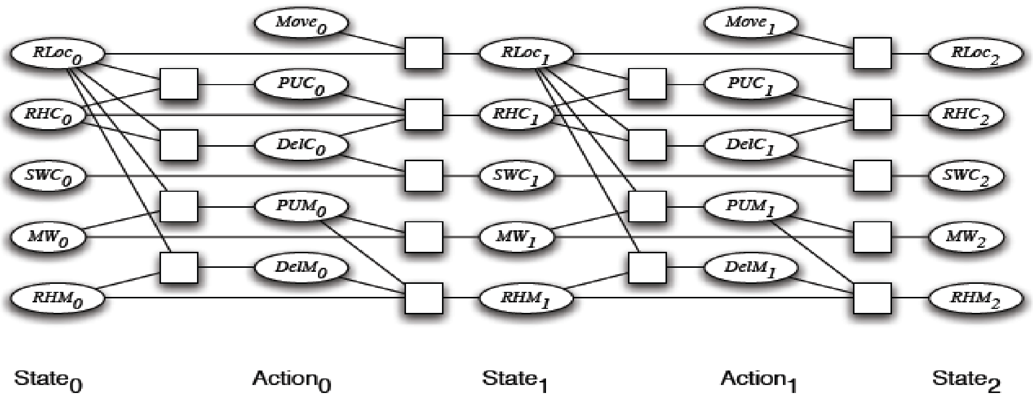
CSP Planning: Solving the problem

Map STRIPS Representation for horizon: 0, 1, 2, …

Run **arc consistency** and **search**,

Plan: all actions with assignment T

In order to find a plan, we expand our constraint network one layer at the time, until a solution is found



Solve planning as CSP: pseudo code

```
solved ← false
horizon ← 0
while not solved
    map STRIPS to CSP with horizon
    solve CSP → solution
    if solution found then
        solved ← true
    else
        horizon ← horizon + 1
return solution
```

Planning as CSP

If the algorithm for planning as CSP stops and returns a solution plan of length k, does it mean that there are no shorter solutions ?

A. Yes

B. No

C. It depends

D. By Einstein’s theory of relativity, the “length” of a solution, whether in time or space, depends on the speed of the system relative to the observer or the differences in local spacetime curvature

E. Bazinga

STRIPS to CSP applet

Allows you:

- to specify a planning problem in STRIPS
- to map it into a CSP for a given horizon
- the CSP translation is automatically loaded into the CSP applet where it can be solved

Practice exercise using STRIPS to CSP is available on AIspace

A related planner

A similar process is implemented (more efficiently) in the **Graphplan** planner

STRIPS to CSP applet

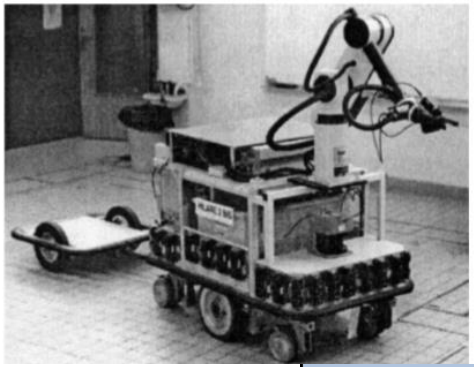
Allows you:

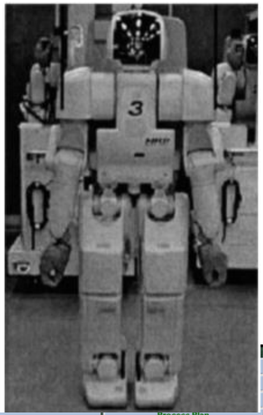
- to specify a planning problem in STRIPS
- to map it into a CSP for a given horizon
- the CSP translation is automatically loaded into the CSP applet where it can be solved

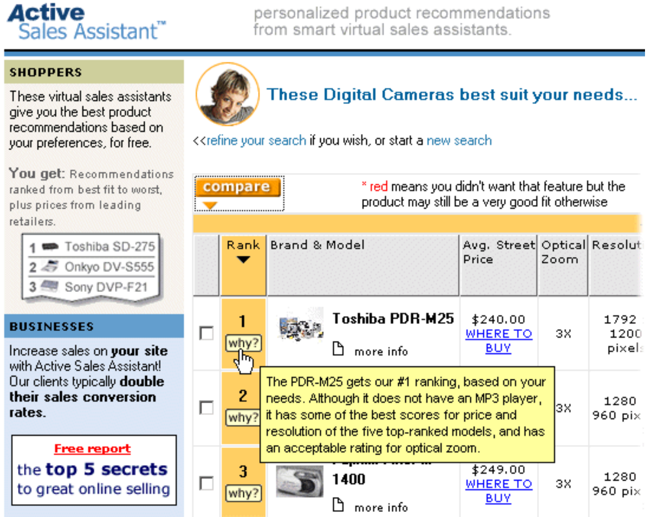
Practice exercise using STRIPS to CSP is available on AIspace

Now, do you know how to implement a planner for....

- Emergency Evacuation?
- Robotics?
- Space Exploration?
- Manufacturing Analysis?
- Games (e.g., Bridge)?
- Generating Natural language
  - Product Recommendations ….







Active Sales Assistant™

personalized product recommendations from smart virtual sales assistants.

SHOPPERS

These virtual sales assistants give you the best product recommendations based on your preferences, for free.

You get: Recommendations ranked from best fit to worst, plus prices from leading retailers.

1 Toshiba SD-275
2 Onkyo DV-S555
3 Sony DVP-F21

BUSINESSES

Increase sales on your site with Active Sales Assistant! Our clients typically double their sales conversion rates.

Free report
the top 5 secrets to great online selling

These Digital Cameras best suit your needs...

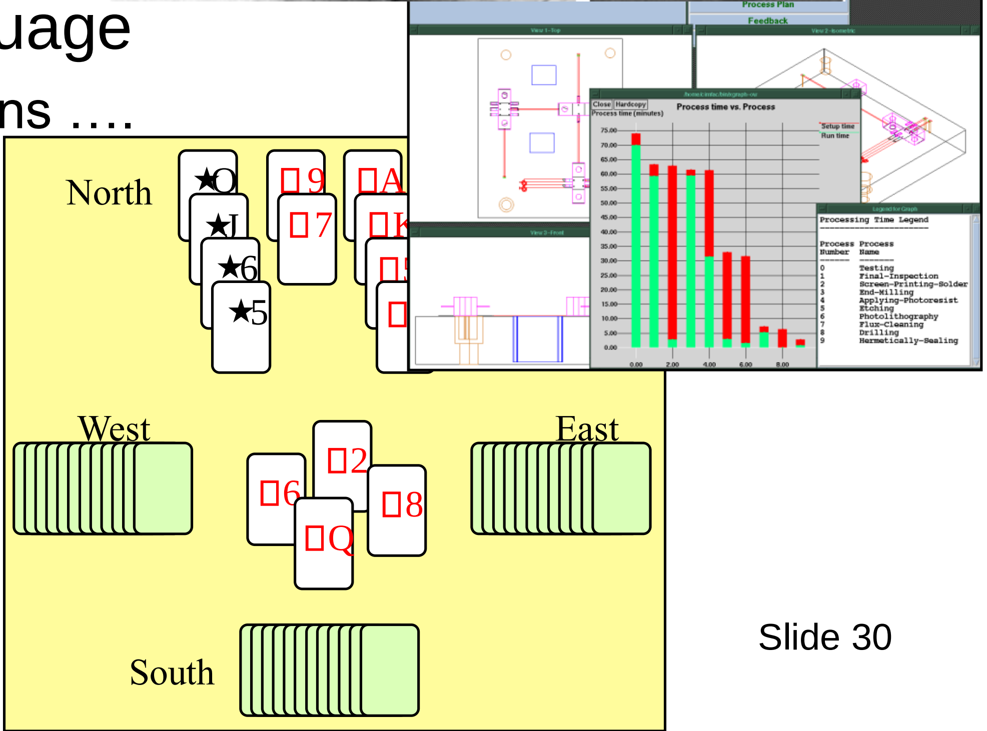
<<refine your search if you wish, or start a new search

compare

* red means you didn't want that feature but the product may still be a very good fit otherwise

	Rank	Brand & Model	Avg. Street Price	Optical Zoom	Resolut
□	1 why?	Toshiba PDR-M25 more info	$240.00 WHERE TO BUY	3X	1792 1200 pixel
□	2 why?			3X	1280 960 pix
□	3 why?	1400 more info	$249.00 WHERE TO BUY	3X	1280 960 pix

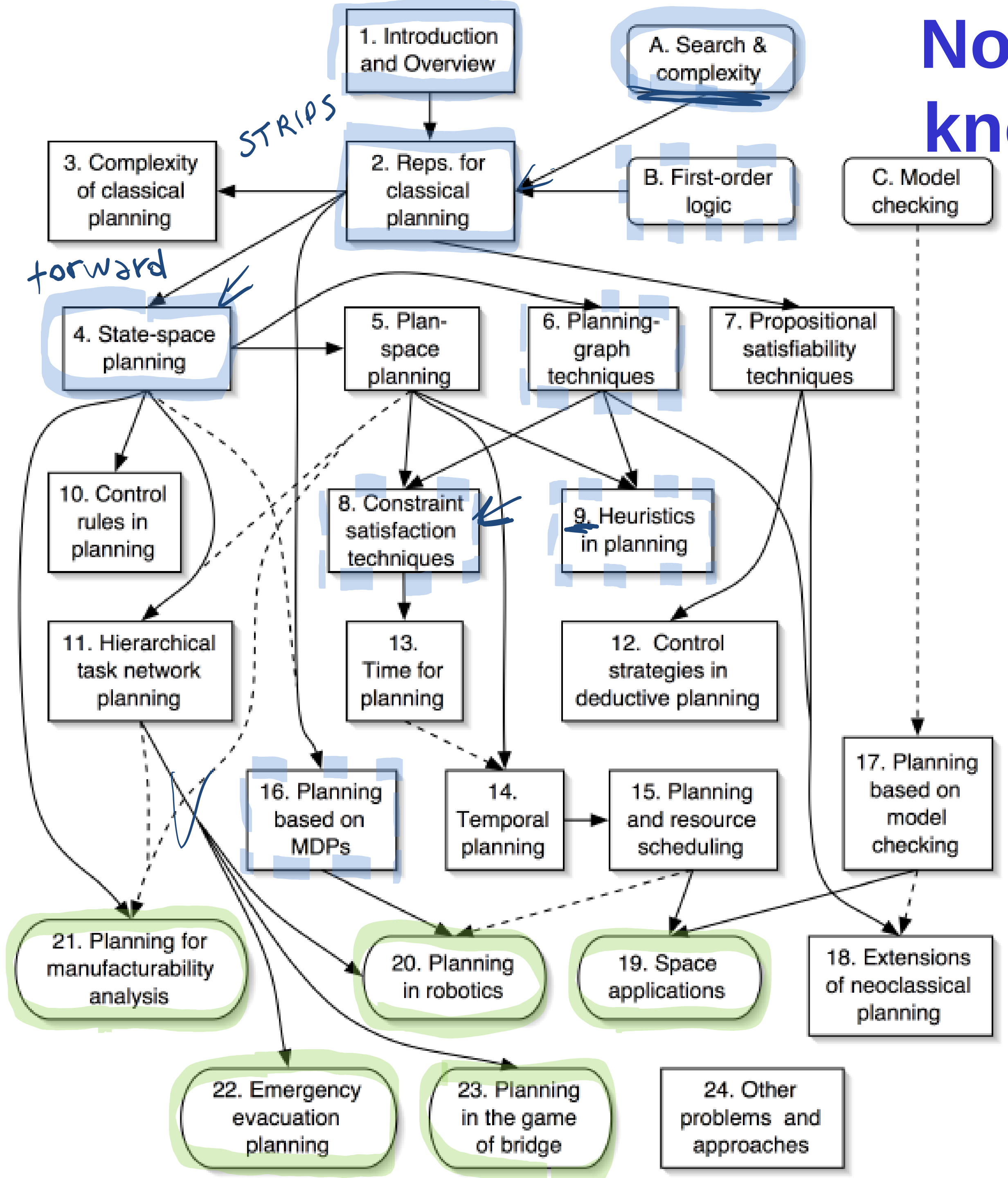
The PDR-M25 gets our #1 ranking, based on your needs. Although it does not have an MP3 player, it has some of the best scores for price and resolution of the five top-ranked models, and has an acceptable rating for optical zoom.



```
Processing Time Legend
----------------------

Process  Process
Number   Name
-------  -------
0        Testing
1        Final-Inspection
2        Screen-Printing-Solder
3        End-Milling
4        Applying-Photoresist
5        Etching
6        Photolithography
7        Flux-Cleaning
8        Drilling
9        Hermetically-Sealing
```

No ☹, but you (will) know the key ideas ☺!



STRIPS

forward

1. Introduction and Overview

A. Search & complexity

2. Reps. for classical planning

B. First-order logic

C. Model checking

3. Complexity of classical planning

4. State-space planning

5. Plan-space planning

6. Planning-graph techniques

7. Propositional satisfiability techniques

10. Control rules in planning

8. Constraint satisfaction techniques

9. Heuristics in planning

11. Hierarchical task network planning

13. Time for planning

12. Control strategies in deductive planning

16. Planning based on MDPs

14. Temporal planning

15. Planning and resource scheduling

17. Planning based on model checking

21. Planning for manufacturability analysis

20. Planning in robotics

19. Space applications

18. Extensions of neoclassical planning

22. Emergency evacuation planning

23. Planning in the game of bridge

24. Other problems and approaches

Ghallab, Nau, and Traverso

Automated Planning: Theory and Practice

Morgan Kaufmann, May 2004

ISBN 1-55860-856-7

Web site:

- ✓ http://www.laas.fr/planning

NEW BOOK

- ✓ http://projects.laas.fr/planning/

▬ you Know

▬ ▬ ▬ you Know a little

Applications

R&R systems we’ll cover in this course

		Environment	
Problem		Deterministic	Stochastic
Static	Constraint Satisfaction	*Variables + Constraints* Search Arc Consistency Local Search	
	Query	*Logics* Search	*Bayesian (Belief) Networks* Variable Elimination
Sequential	Planning	*STRIPS* Search	*Decision Networks* Variable Elimination

Representation

Reasoning Technique

Logics

- **Mostly only propositional**…. This is the starting point for more complex ones ….
- **Natural** to express **knowledge** about the world
  - What is true (boolean variables)
  - How it works (logical formulas)
- Well understood formal properties
- Boolean nature can be exploited for efficiency
- ……