

Maintaining Focus

Overcoming Attention Deficit Disorder in Contingent Planning

Ron Alford¹ Ugur Kuter¹ Dana Nau¹
Elnatan Reisner¹ Robert Goldman²

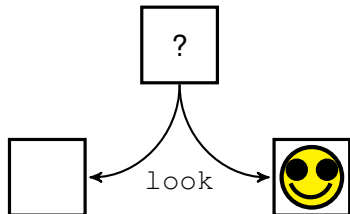
¹University of Maryland, College Park

²Smart Information Flow Technologies

FLAIRS-22 Technical Presentation

Contingent Planning

- Generate conditional plans given that some facts are unknown
 - For example: Is there someone in the next room?
- “Observations” can discover facts during plan execution
 - i.e., look in the next room, is it occupied
- Need to generate conditional plans that branch on the observations



State-of-the-art planners fail on small problems

Standard benchmark problems (time in seconds):

(a) The Unix domain (8 directories)

No. of files:	1	2	3	4
Contingent-FF	0.02	0.98	24.99	—
MBP	1.08	—	—	—
PKS	—	—	—	—
POND	0.37	—	—	—

(b) The Robot Navigation Domain (8 rooms)

No. of packages:	1	2	3	4
Contingent-FF	0.02	0.33	5.57	—
MBP	0.08	—	—	—
PKS	1.32	265.95	—	—
POND	—	—	—	—

(c) The Medicate Domain

No. of patients:	1	2	3	4
Contingent-FF	0.01	0.13	1.87	18.25
PKS	1.32	265.95	—	—
POND	0.00	—	—	—

(d) The Rovers Domain (5 waypoints)

No. of objectives:	1	2	3	4
Contingent-FF	0.01	0.20	3.17	—
PKS	1.32	265.95	—	—
POND	0.14	—	—	—

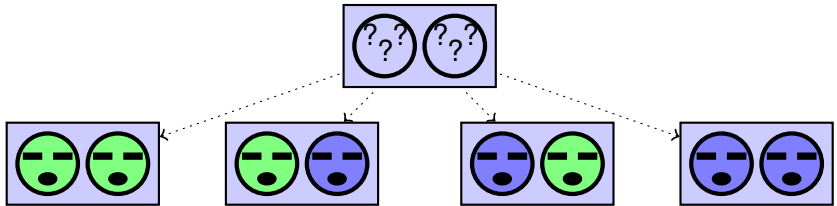
Why Didn't The Planners Do Well?

Example: The Medicate Domain

- n patients, $p_1 \dots p_n$.
- Each patient has exactly one of m possible illnesses: $i_1 \dots i_m$.
- For each illness i_j there is a treatment action treat_j that will:
 - Cure the patient if he/she has i_j
 - Kill the patient otherwise.
- For each illness i_j there is an observation action test_j that confirms whether or not the patient has i_j .

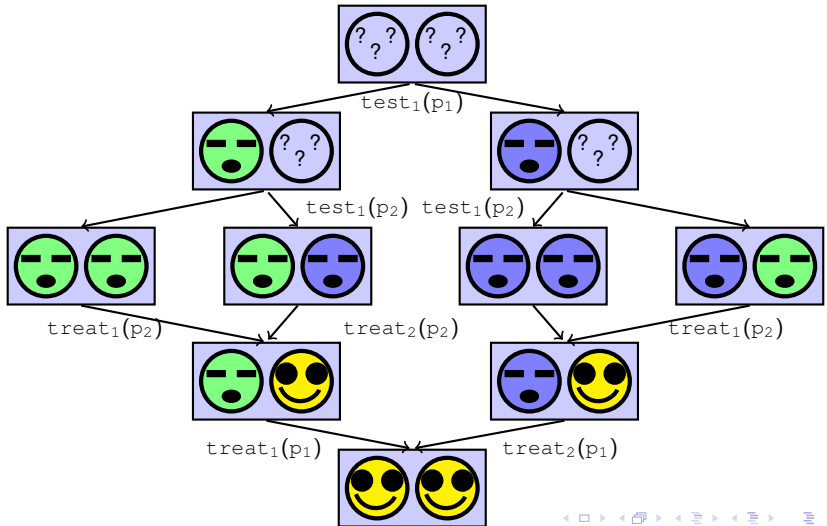


Medicate State Space



- n patients
- m possible disorders per patient
- $\implies m^n$ possible “true” initial states of the world
- For each initial state, we need a different plan.

Worst-case solutions to Medicate



Our Contributions

How to enable contingent planners to overcome this combinatorial explosion?

- A general way to modify a planning domain's operators
 - Helps a planner "focus" its search on whichever contingency the planner is currently trying to solve.
 - Experimental results
- FOCUS Planner
 - More extensive focussing information
 - A new algorithm, FOCUS, to make use of this knowledge
 - Experimental results

Giving Focus Information to Contingent Planners

Force a planner to consider a small number of contingencies at a time:

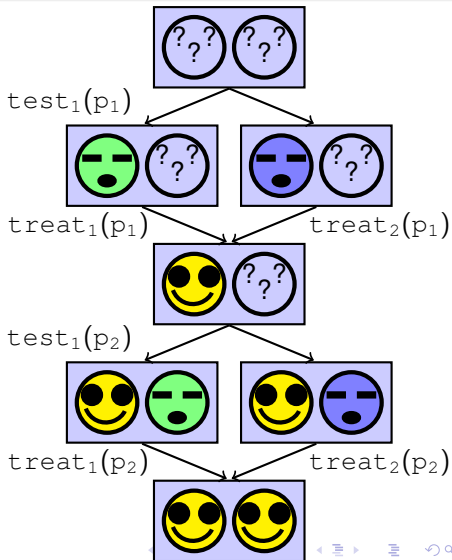
- Have a state assertion that specifies what subproblem the planner can work on.
- Add a “focus” operator that can add this assertion
- Add assert to the precondition of operators for the subproblem.
- Have final operator for subproblem remove this assertion.

Medicate Example

- $\text{focus}(p)$
Precond: $\neg \text{focusing}()$
Effects: $\text{focusing}(),$
 $\text{focused-on}(p)$
- $\text{test}(p, i)$
Precond: $\text{focused-on}(p)$
Effects: ...
- $\text{treat}(p, i)$
Precond: ...
 $\text{focused-on}(p)$
Effects: ...
 $\neg \text{focused-on}(p),$
 $\neg \text{focussing}()$

Focused Medicate Solutions

- Plan branches on tests
- Branches merge after treatments



Focus Results for PKS

Contingencies:	1	2	3	4	5
Unix	— 0.11	— 0.99	— 8.27	— 67.58	— 664.98
Robot	1.32 0.44	265.95 7.33	— 116.02	— —	— —
Medicate	1.32 0.00	265.95 0.06	— 0.38	— 1.73	— 10.13
Rovers	0.14 0.00	— 0.14	— 1.32	— 10.76	— 81.15

Time in seconds. **Baseline results in red.** Focus results in blue.

Summary

- Current contingent planners can only handle a small number of independent contingencies
- Adding a focus operator to the domain can reduce the burden on the planner