Graph-Based and Sat-Based Planning

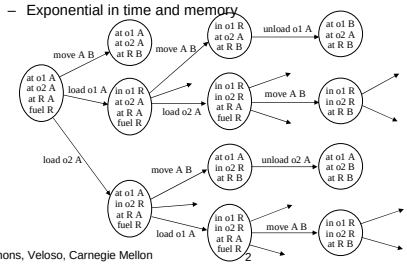
Manuela Veloso
Reid Simmons

PEL, Fall 2010

September 27, 2010

Planning Graph – Forward Expansion

- State reachability – "until" goal
  - Can find ***all*** goals reachable from initial state
  - Exponential in time and memory

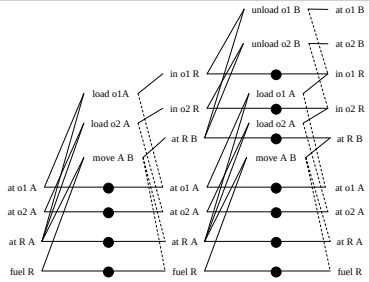


Graphplan

Blum & Furst 95

- Preprocessing before engaging in search.
- Forward search combined with backward search.
- Construct a *planning graph* to reveal constraints
- Two stages:
  - Extend: One time step in the planning graph.
  - Search: Find a valid plan in the planning graph.
- Graphplan finds a plan or proves that no plan has fewer "time steps."

Plan Graph
One-Way Rocket Example



Extending a Planning Graph - Actions

- To create an action-level *i*:
  - Add each instantiated operator, for which all of its preconditions are present at proposition-level *i* AND *no two of its preconditions are exclusive*.
  - Add all the no-op actions.
- Determine the exclusive actions.

Extending a Planning Graph - Propositions

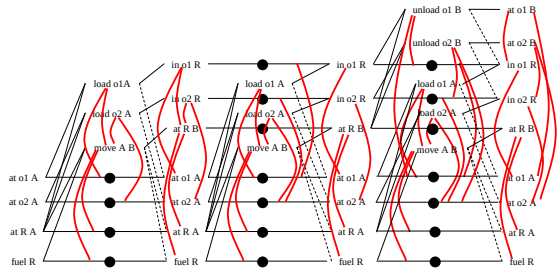
- To create a proposition-level *i* + 1:
  - Add all the effects of the inserted actions at action-level *i* - distinguishing add and delete effects.
- Determine the exclusive actions.

Planning Graphs

- A literal may exist at level *i* + 1 if it is an Add-Effect of some action in level *i*.
- Two propositions *p* and *q* are *exclusive* in a proposition-level if ALL actions that add *p* are exclusive of ALL actions that add *q*.
- Actions A and B are *exclusive* at action-level *i*, if:
  - Interference: A (or B) deletes a precondition or an Add-Effect of B (or A).
  - Competing Needs: *p* is a precondition of A and *q* is a precondition of B, and *p* and *q* are exclusive in proposition-level *i* - 1.

Mutex Exclusivity Relations

One-Way Rocket Example



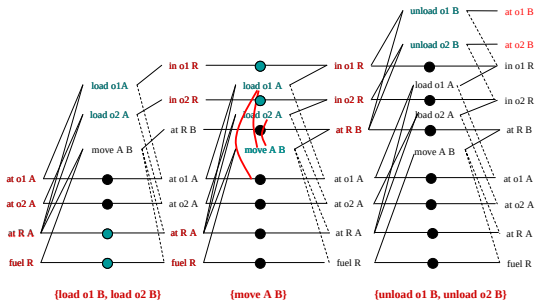
Exclusivity Examples

- Exclusive Actions: (Move A B) deletes a precondition of (Load o1 A). Therefore exclusive (existence of threats).
- Exclusive Propositions: (at R A) and (at R B) at time 2 are exclusive. (at R A) is added by a no-op and (at R B) is added by (Move A B) and no-op and (Move A B) are exclusive actions.
- Exclusive Actions: Then (Load o1 A) and (Load o2 B) are exclusive because (at R A) and (at R B) are exclusive.
- Propositions can be exclusive in some time step and not in others: If (at o1 A) and (at R A) at time 1, then (in o1 A) and (at R B) are exclusive at time 2, but not at time 3.

Searching a Planning Graph

- Level-by-level backward-chaining approach to use the exclusivity constraints.
- Given a set of goals at time *t*, identify all the sets of actions (including no-ops) at time *t* - 1 who add those goals and are not exclusive. The preconditions of these actions are new goals for *t* - 1.

Searching a Planning Graph



Recursive Search Implementation

- For each goal at time *t* in some arbitrary order:
  - Select some action at time *t* - 1 that achieves that goal and it is not exclusive with any other action already selected.
  - Do this recursively for all the goals at time *t* - do not add new action, but use the ones already selected if they add another goal.
  - If recursion returns failure, then select a different action.
- The new goal set is the set of all the preconditions of the selected actions.

Enhancements

- Forward-checking - for the goals ahead, check if all the actions that add it are exclusive with the selected action.
- Memoization - when a set of goals is not solvable at some time *t*, then this is recorded and hashed. If back at time *t*, the hash table is checked and search proceeds backing up right away.

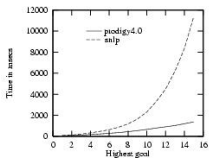
Planning as Satisfiability

- One interpretation: ``first-order deductive theorem-proving does not scale well.''
- One solution: ``propositional satisfiability''
- Uniform clausal representation for goals and operators.
- Stochastic local search is a powerful technique for planning.

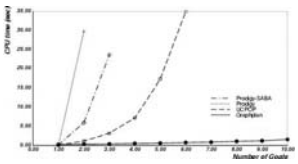
SatPlan

- Assume the plan has *n* (time-parallel) steps. *(strong assumption)*
- Initial state: completely specified at time 0.
  at-o1-A_0 ∧ at-o2-A_0 ∧ at-R-A_0
- Goal: specified at time 2*n*.
  at-o1-B_6 ∧ at-o2-B_6
- Actions: specified at *odd* times; An action implies its preconditions and effects.
  (¬load-o1-A_1 ∨ at-o1-A_0) ∧ (¬load-o1-A_1 ∨ at-R-A_0) ∧ (¬load-o1-A_1 ∨ in-R-A_2) ∧ (¬load-o1-A_1 ∨ ¬at-o1-A_2)

Performance



Nonlinear state-space planning can suffer from goal orderings, but can use state information to reduce the search space
Veloso et al.



Graph-based planning combines forward state expansion and exclusivity analysis with backwards goal search
Blum et al.

Performance (cont.)

Prob	UMOP OBDDs		Prodigy4.0 NL state		STAN graphplan		HSP A*heuristics		IPP graphplan		Blackbox satplan	
1	20	11	80	15	46	11	2007	13	50	15	113	11
2	150	17	200	23	1075	17	2150	21	380	23	7820	17
3	710	23	210	31	54693	23	2485	31	3270	31	-	-
4	1490	29	370	39	3038381	29	3060	37	26680	39	-	-
5	3600	35	430	47	-	-	3320	47	226460	47	-	-
6	7260	41	590	55	-	-	3779	53	-	-	-	-
7	13750	47	800	63	-	-	4797	63	-	-	-	-
8	23840	53	960	71	-	-	5565	71	-	-	-	-
9	36220	59	1240	79	-	-	6675	79	-	-	-	-
10	56200	65	1560	87	-	-	7583	85	-	-	-	-
11	84930	71	1820	95	-	-	9060	93	-	-	-	-
12	127870	77	2240	103	-	-	10617	101	-	-	-	-
13	197170	83	2660	111	-	-	12499	109	-	-	-	-
14	290620	89	3200	119	-	-	15050	119	-	-	-	-
15	411720	95	3740	127	-	-	16886	125	-	-	-	-
16	549610	101	4350	135	-	-	20084	135	-	-	-	-
17	746920	107	5030	143	-	-	23613	143	-	-	-	-
18	971420	113	5900	151	-	-	26973	151	-	-	-	-
19	1361580	119	6810	159	-	-	29851	157	-	-	-	-
20	1838110	125	7710	167	-	-	33210	165	-	-	-	-

planning time in ms; number of plan steps

Performance (cont.)

Prob	UMOP OBDDs		Prodigy4.0 NL state		STAN graphplan		HSP A*heuristics		IPP graphplan		Blackbox satplan	
1	20	7	30	7	19	7	2121	7	10	7	11	7
2	20	7	70	7	18	7	2104	7	10	7	12	7
3	20	7	30	7	19	7	2144	7	10	7	14	7
4	20	7	40	7	20	7	2188	7	10	7	16	7
5	20	7	30	7	21	7	2208	7	10	7	18	7
6	20	7	40	7	22	7	2617	7	10	7	20	7
7	30	7	50	7	22	7	2316	7	20	7	22	7
8	10	7	40	7	23	7	2315	7	20	7	24	7
9	20	7	50	7	25	7	2357	7	-	-	26	7
10	30	7	80	7	26	7	2511	7	10	7	29	7
11	10	7	60	7	27	7	2427	7	30	7	30	7
12	20	7	60	7	28	7	2456	7	30	7	32	7
13	20	7	90	7	29	7	3070	7	20	7	36	7
14	20	7	60	7	31	7	2573	7	30	7	35	7
15	20	7	100	7	32	7	2577	7	30	7	38	7
16	20	7	70	7	34	7	2699	7	10	7	39	7
17	20	7	100	7	35	7	2645	7	30	7	41	7
18	20	7	90	7	37	7	2686	7	10	7	43	7
19	20	7	110	7	39	7	2727	7	30	7	45	7
20	20	7	90	7	40	7	2787	7	20	7	47	7

planning time in ms; number of plan steps

Performance (cont.)

Prob	STAN graphplan		HSP A*heuristics		IPP graphplan		Blackbox satplan	
1	767	27	79682	43	900	26	2062	27
2	4319	32	97114	44	-	-	6436	32
5	364932	29	144413	26	2400	24	-	-
7	-	-	788914	112	-	-	-	-
11	12806	34	86195	30	6940	33	6544	32

planning time in ms; number of plan steps

- UMOP needs tractable representation (on-going research).
- Prodigy needs control knowledge (planning by analogy or control rules).

Summary

- Disjunctive reasoning in Graphplan
- Complete plans in SatPlan
- Next class – Heuristics in graph-based planning