Decomposed Hierarchical Planning

Jason Wolfe
UC Berkeley

Bhaskara Marthi
Willow Garage

Stuart Russell
UC Berkeley

Real World Decision Making

- Video of PR2 cleaning room

Levels of Decision Making

- Which object to put away next?
- How to arrange objects in cupboard?
- Where to place base to pick up object?
- Where to grasp object?
- What type of grasp to use?
- What is the full joint configuration at grasp?
- What path in cspace to take to achieve grasp?
- What joint efforts to apply to follow path?

These decisions are not independent!

Top-down Decision Making

- Make decisions in top-down order
- How to handle lower level planning failures?
- Can be unboundedly suboptimal

This work

- DASH-A* Planner
- Find hierarchically optimal plans
- For efficiency:
  - Decompose across subproblems whenever possible
  - State abstraction: reuse solutions across subproblems
  - Angelic bounds on reachable sets and costs at all levels -> pruning

Pick and Place Domain

- Video

Action Hierarchies

HLA	Refinements
`Act`	[`MoveToGoal`(o), `Act`] | o not at goal [] | all objects at goals
`MoveToGoal`(o)	[`GoPick`(o), `GoPlace`(o, p)] | p in goal region of o
`GoPick`(o)	[`Pick`(o)] | o in range [`ArmTuck`,`BaseRgn`(r),`Pick`(o)] | r is candidate base region
`Pick`(o)	[`ArmGraspAction`($pos(o), \theta$), `CloseGripperAction`(o), `TorsoAction`(up)] | $\theta \in [-1, 1]$ rad
`GoPlace`(o, p)	[`Place`(o, p)] | p in range [`ArmTuck`,`BaseRgn`(r),`Place`(o, p)] | r is candidate base region
`Place`(o, p)	[`ArmJointAction`($\boldsymbol{\theta}_1$), `TorsoAction`($down$), `OpenGripperAction`, `ArmJointAction`($\boldsymbol{\theta}_2$)] | $\boldsymbol{\theta}$ are candidate joint configs
`ArmTuck`	[`ArmJointAction`($tucked$)]
`BaseRgn`(r)	[`BaseAction`(p, θ)] | $p, \theta \in r$

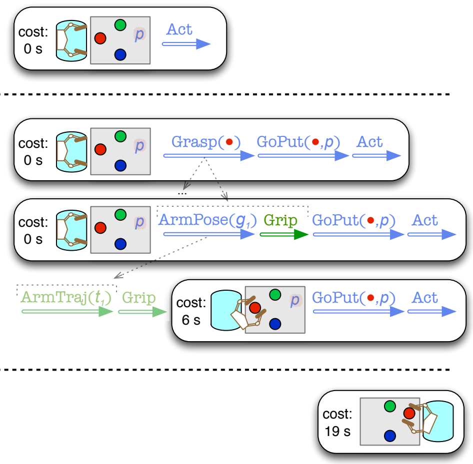
Hierarchical Uniform-Cost Search

- Nodes are (state, plan suffix) pairs

Init

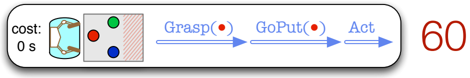
Successors

Goal



Angelic Semantics

- What if we want a heuristic, for hierarchical-A*?

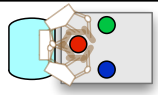
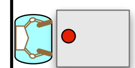


- Angelic semantics provides provably correct abstract transition models
  - optimistic descriptions:
    overestimate reachability,
    underestimate costs

Going to undersell things a bit here... also pessimistic descriptions, of potentially much greater interest, since they allow committing to provably correct abstract plans. For today, going to focus on the A* / optimistic-only story, may talk a bit about pessimistic at end.

Angelic Hierarchical Search Problems

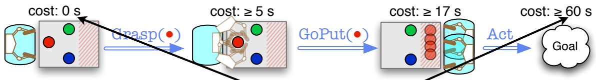
For each action *a*, input set *i*:

	Optimistic Outcome			Children	Context
	Set	Cost	Status		
description	reachable states	lower bound	solved, refinable, or blocked	action pairs	state frags.
example: 		5 s	refinable	ArmPose(p_1), CloseGripper; ArmPose(p_2), CloseGripper;	

Precise definition of what goes into angelic search problem. Unifies primitive, high-level semantics.
Still have designated initial state, top-level action Act.
In status, we see two ways to "refine" -- expand action, or narrow the input set.
One way to think about this framework: action-generated state abstractions.

Angelic Hierarchical A* (AHA*)

- Essentially just H-UCS with a heuristic



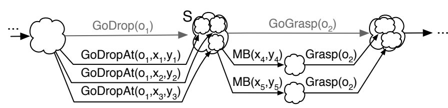
$$f = g + h$$

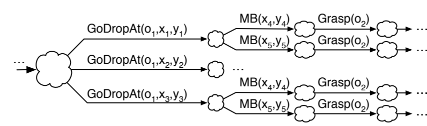
$$60 = 0 + 60$$

- Other difference: can refine any refinable action

AHA* Drawback

- Number of potential plans grows exponentially
  - # refs of action 1 * # refs of action 2 # refs of action 3
  - Even when state space is small!
  - Pruning doesn't help enough





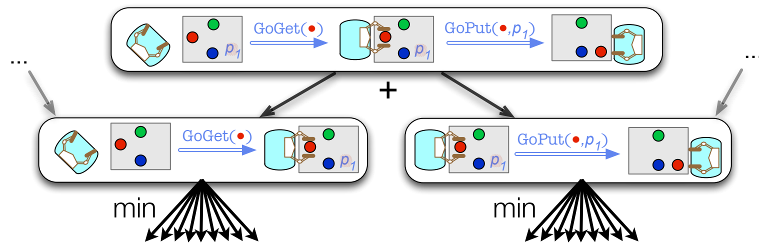
“Singleton” DASH-A*

- First step towards full DASH-A* algorithm
- DASH-A* features
  - Decomposed
  - Angelic
  - State-abstracted
  - Hierarchical

Essentially same as “explicit DASH-A*” algorithm I talked about awhile ago.

Decomposition

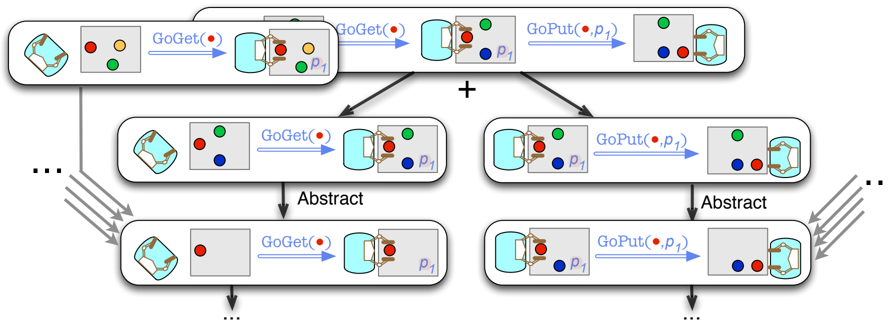
- Given fixed intermediate states, planning for a sequence of HLAs decomposes into independent subproblems
- Tree decomposition of hierarchical plan space



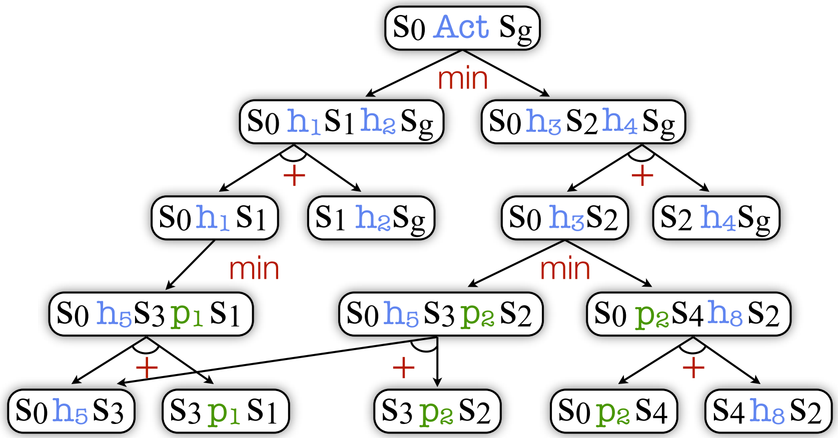
Concatenate two planning problems together.

State Abstraction

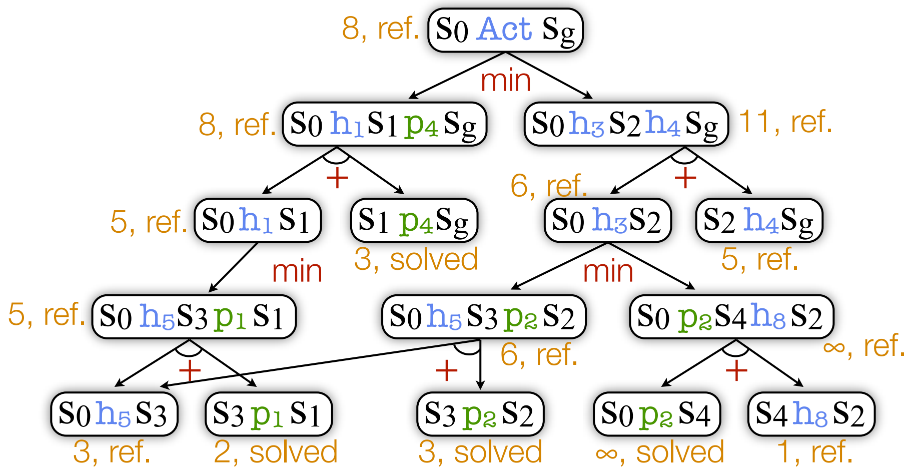
- Can use context (relevance) to further increase sharing/compression of open list



Search: min/sum graphs



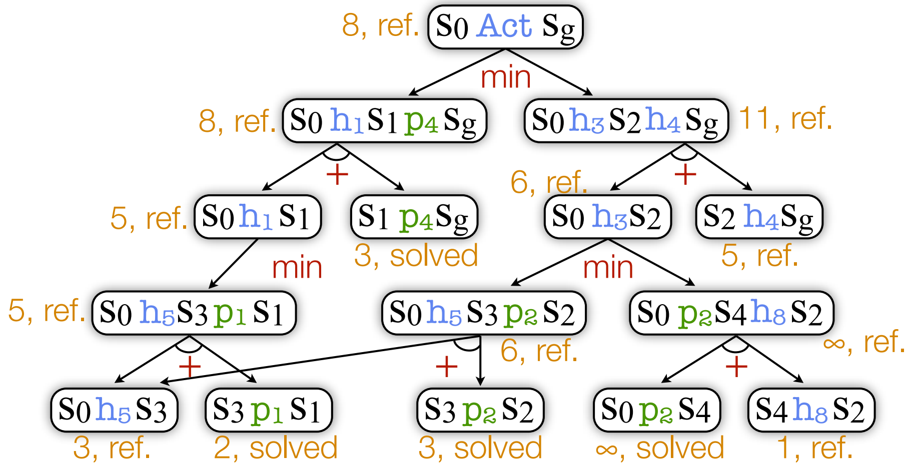
Search: summaries



Search: AO*

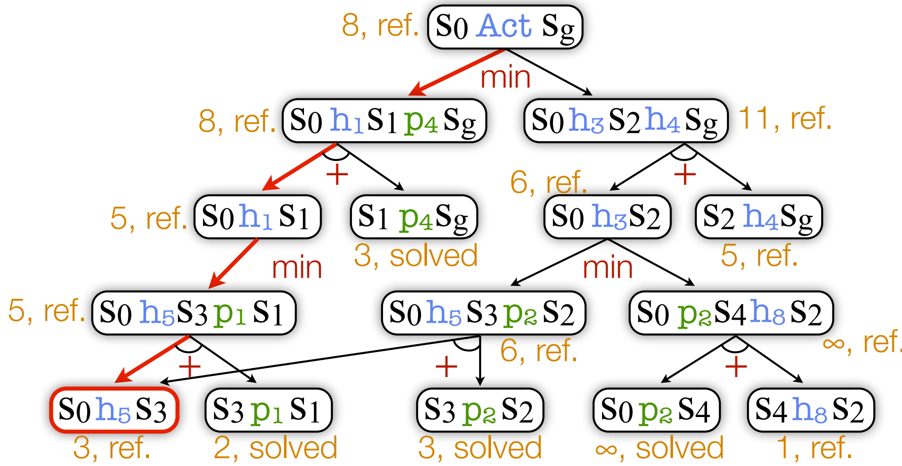
1. Expand a best leaf node
   - Start at root
   - Pick min-cost child at min
   - Pick any unsolved child at **+**
   - Expand reached leaf
2. Propagate labels upwards
   - Rewards follow labels
   - Break ties: solved < refinable
   - Sum solved iff both children solved
3. If root not solved, goto 1

Search: AO*



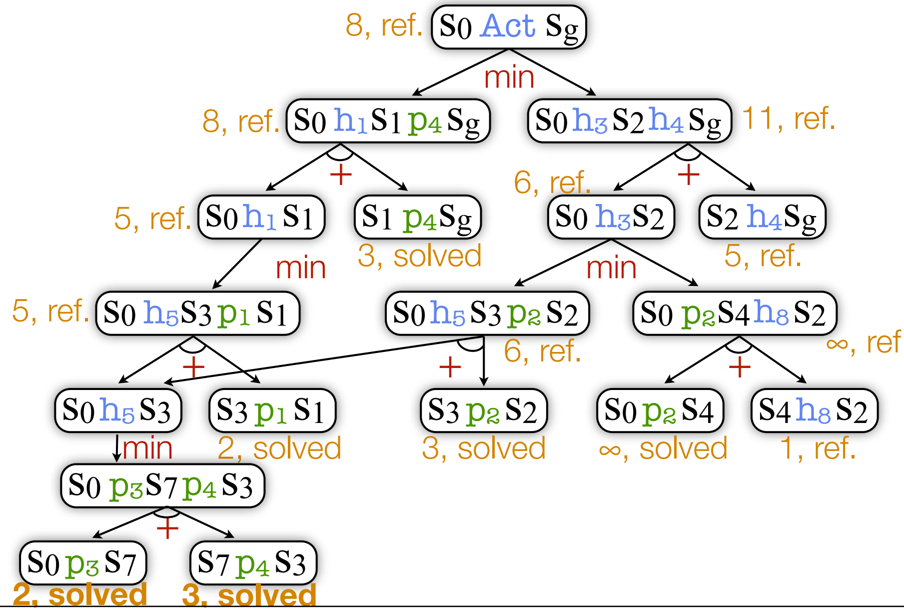
8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
+
5, ref.
S0 h1 S1
S1 p4 Sg
3, solved
6, ref.
S0 h3 S2
S2 h4 Sg
5, ref.
min
min
5, ref.
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
6, ref.
S0 p2 S4 h8 S2
∞, ref.
+
+
+
S0 h5 S3
3, ref.
S3 p1 S1
2, solved
S3 p2 S2
3, solved
S0 p2 S4
∞, solved
S4 h8 S2
1, ref.

Search: AO*



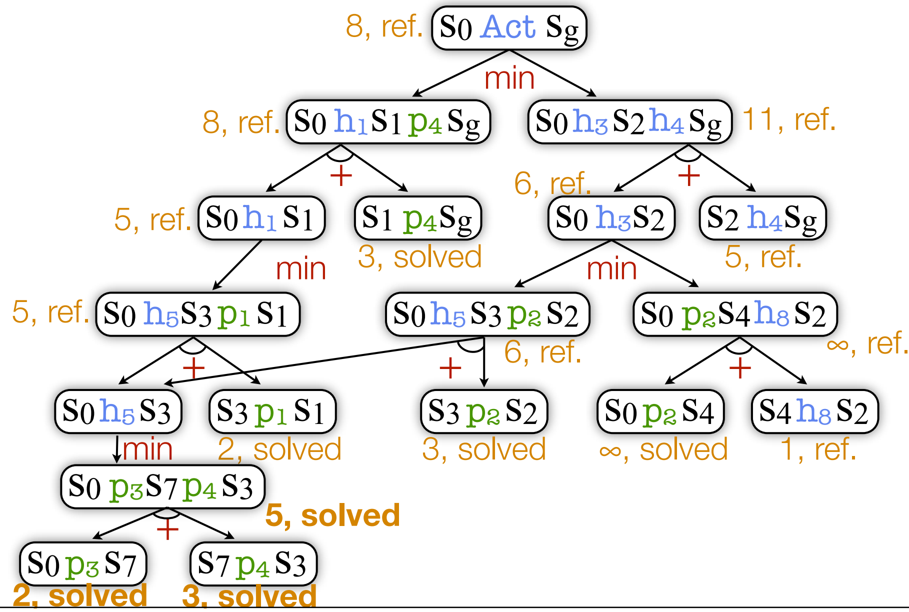
8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
+
6, ref.
5, ref.
S0 h1 S1
S1 p4 Sg
3, solved
S0 h3 S2
S2 h4 Sg
5, ref.
min
min
5, ref.
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
S0 p2 S4 h8 S2
∞, ref.
6, ref.
+
+
+
S0 h5 S3
3, ref.
S3 p1 S1
2, solved
S3 p2 S2
3, solved
S0 p2 S4
∞, solved
S4 h8 S2
1, ref.

Search: AO*



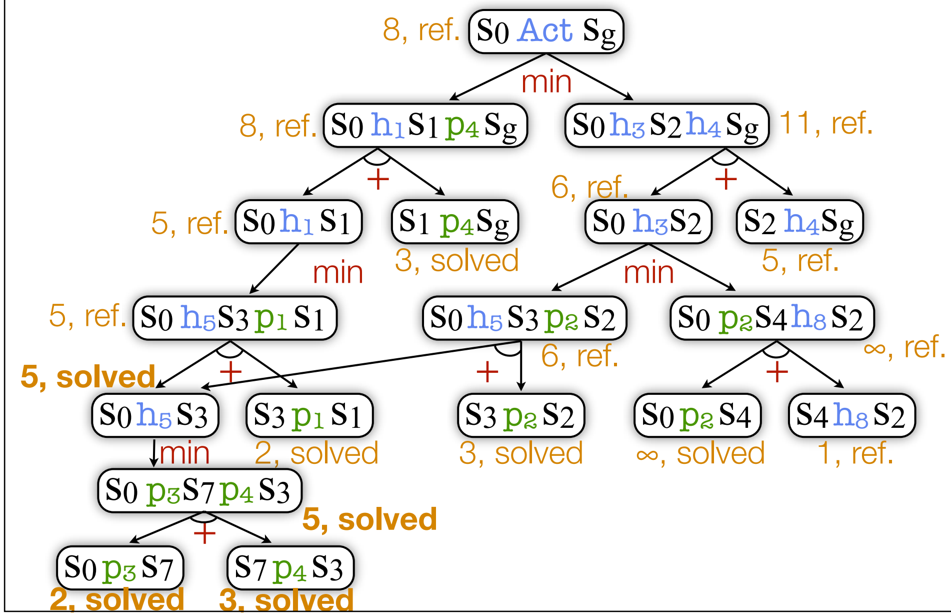
8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
+
5, ref.
S0 h1 S1
S1 p4 Sg
3, solved
6, ref.
S0 h3 S2
S2 h4 Sg
5, ref.
min
min
5, ref.
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
S0 p2 S4 h8 S2
6, ref.
∞, ref.
+
+
+
S0 h5 S3
S3 p1 S1
S3 p2 S2
S0 p2 S4
S4 h8 S2
min
2, solved
3, solved
∞, solved
1, ref.
S0 p3 S7 p4 S3
+
S0 p3 S7
S7 p4 S3
2, solved
3, solved

Search: AO*

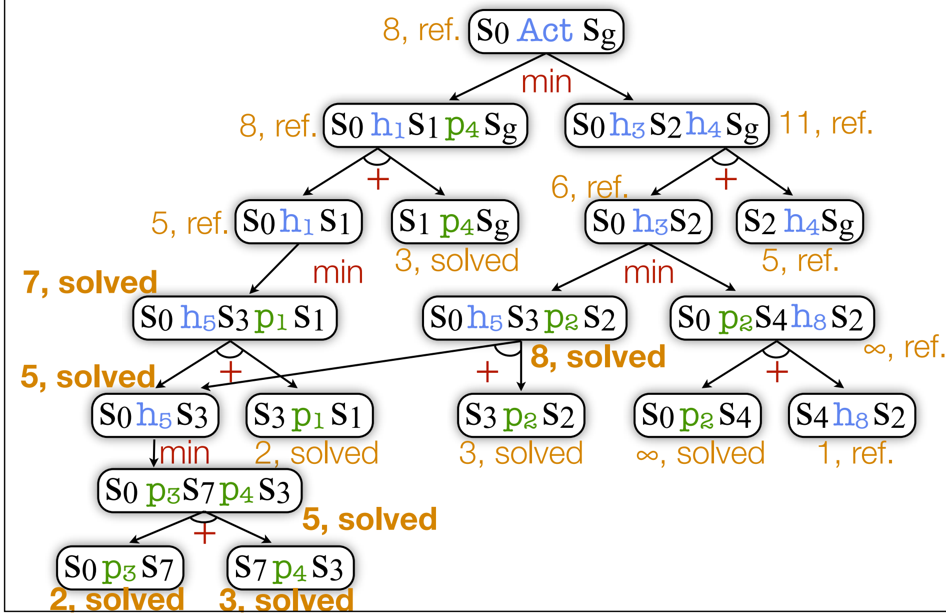


8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
+
5, ref.
S0 h1 S1
S1 p4 Sg
3, solved
6, ref.
S0 h3 S2
S2 h4 Sg
5, ref.
min
min
5, ref.
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
6, ref.
S0 p2 S4 h8 S2
∞, ref.
+
+
+
S0 h5 S3
min
S3 p1 S1
2, solved
S3 p2 S2
3, solved
S0 p2 S4
∞, solved
S4 h8 S2
1, ref.
S0 p3 S7 p4 S3
5, solved
+
S0 p3 S7
2, solved
S7 p4 S3
3, solved

Search: AO*

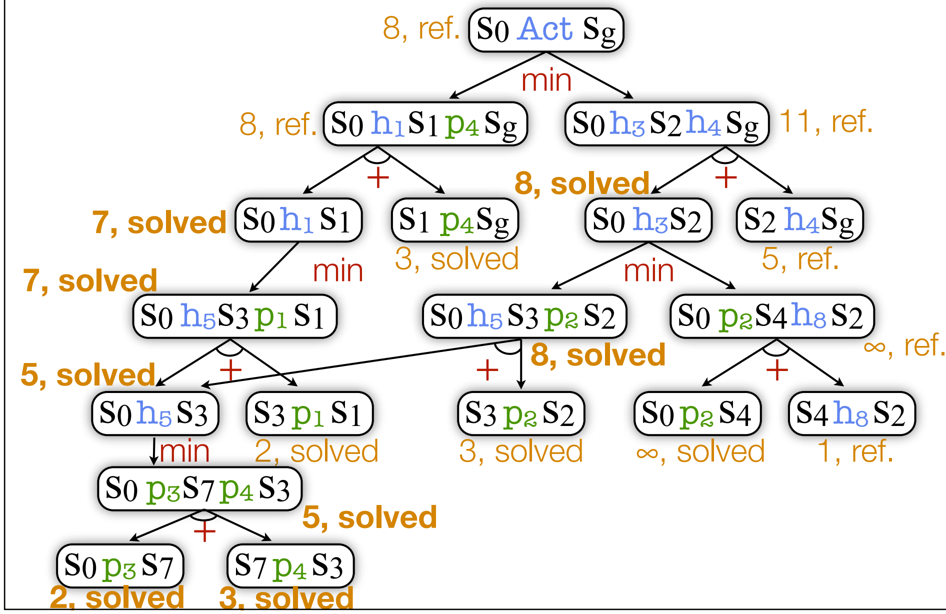

8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
5, ref.
S0 h1 S1
S1 p4 Sg
3, solved
6, ref.
S0 h3 S2
S2 h4 Sg
5, ref.
min
5, ref.
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
6, ref.
S0 p2 S4 h8 S2
∞, ref.
5, solved
S0 h5 S3
S3 p1 S1
2, solved
S3 p2 S2
3, solved
S0 p2 S4
∞, solved
S4 h8 S2
1, ref.
min
S0 p3 S7 p4 S3
5, solved
S0 p3 S7
2, solved
S7 p4 S3
3, solved

Search: AO*



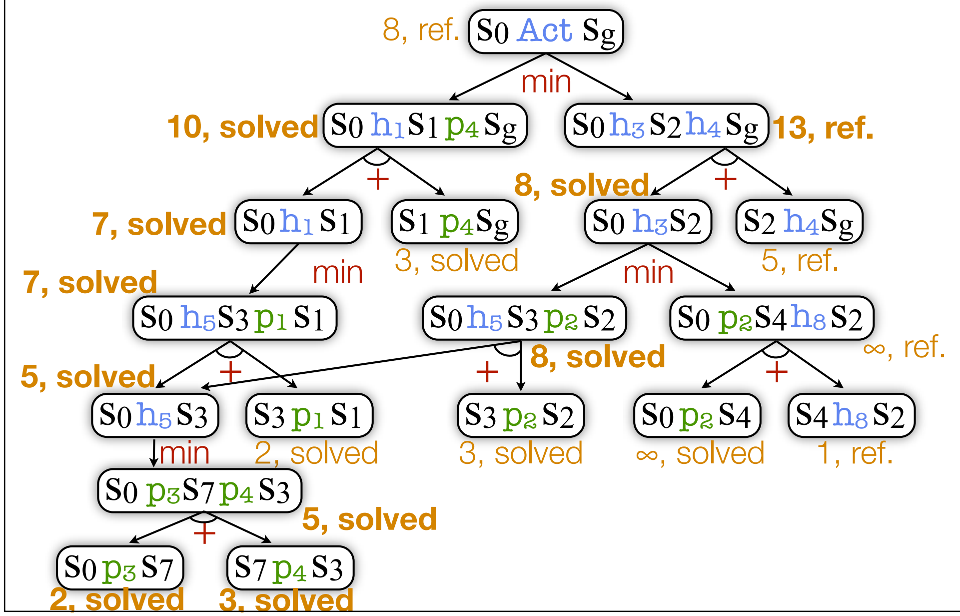
8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
+
5, ref.
S0 h1 S1
S1 p4 Sg
3, solved
6, ref.
S0 h3 S2
S2 h4 Sg
5, ref.
min
min
7, solved
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
S0 p2 S4 h8 S2
∞, ref.
8, solved
5, solved
+
+
+
S0 h5 S3
S3 p1 S1
S3 p2 S2
S0 p2 S4
S4 h8 S2
min
2, solved
3, solved
∞, solved
1, ref.
S0 p3 S7 p4 S3
5, solved
+
S0 p3 S7
S7 p4 S3
2, solved
3, solved

Search: AO*



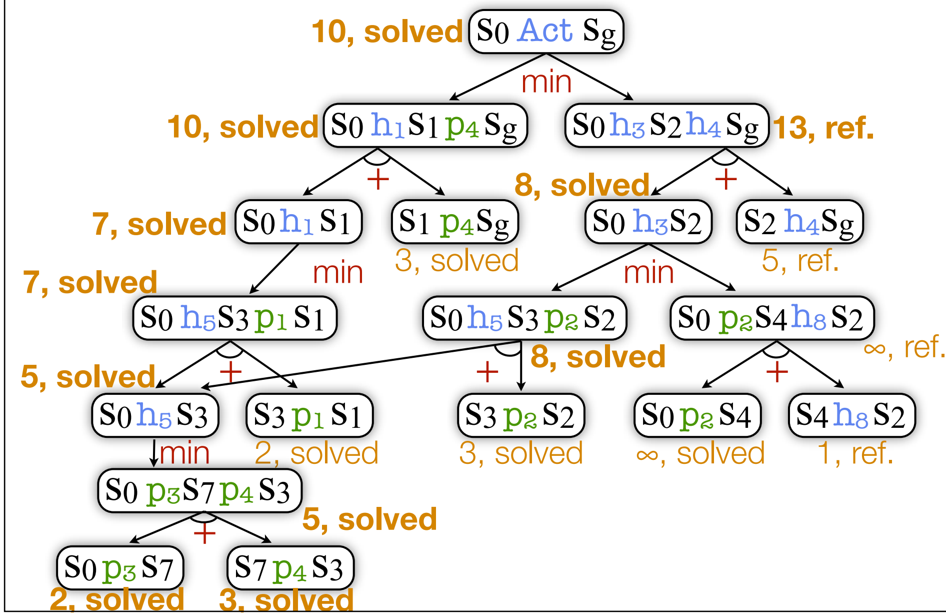
8, ref.
S0 Act Sg
min
8, ref.
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
11, ref.
+
8, solved
+
7, solved
S0 h1 S1
S1 p4 Sg
S0 h3 S2
S2 h4 Sg
min
3, solved
min
5, ref.
7, solved
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
S0 p2 S4 h8 S2
∞, ref.
8, solved
5, solved
+
+
+
S0 h5 S3
S3 p1 S1
S3 p2 S2
S0 p2 S4
S4 h8 S2
min
2, solved
3, solved
∞, solved
1, ref.
S0 p3 S7 p4 S3
5, solved
+
S0 p3 S7
S7 p4 S3
2, solved
3, solved

Search: AO*



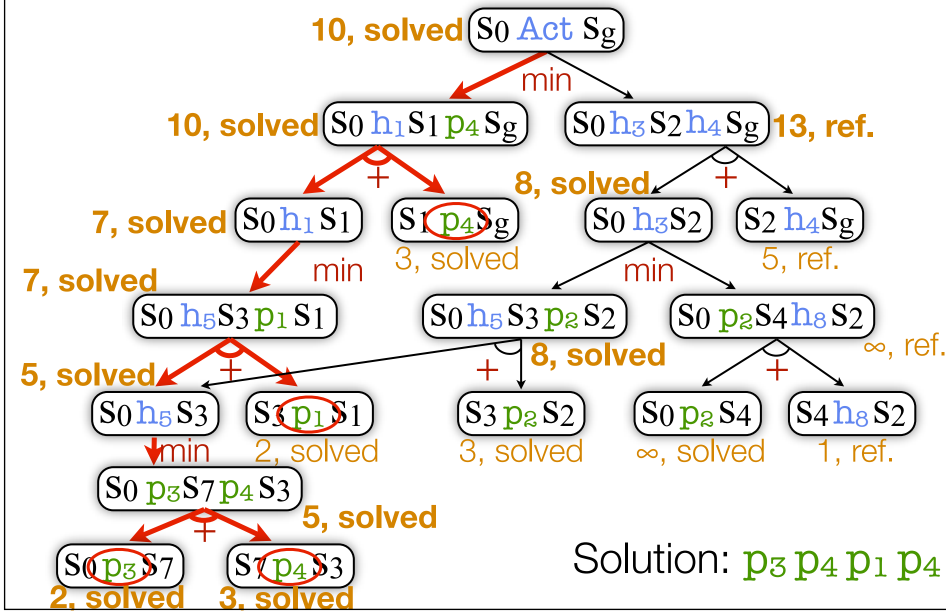
8, ref.
S0 Act Sg
min
10, solved
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
13, ref.
+
7, solved
S0 h1 S1
S1 p4 Sg
3, solved
8, solved
+
S0 h3 S2
S2 h4 Sg
5, ref.
min
7, solved
S0 h5 S3 p1 S1
min
S0 h5 S3 p2 S2
S0 p2 S4 h8 S2
∞, ref.
5, solved
+
8, solved
+
+
S0 h5 S3
S3 p1 S1
S3 p2 S2
S0 p2 S4
S4 h8 S2
min
2, solved
3, solved
∞, solved
1, ref.
S0 p3 S7 p4 S3
5, solved
+
S0 p3 S7
S7 p4 S3
2, solved
3, solved

Search: AO*



10, solved
S0 Act Sg
min
10, solved
S0 h1 S1 p4 Sg
S0 h3 S2 h4 Sg
13, ref.
+
8, solved
+
7, solved
S0 h1 S1
S1 p4 Sg
S0 h3 S2
S2 h4 Sg
min
3, solved
min
5, ref.
7, solved
S0 h5 S3 p1 S1
S0 h5 S3 p2 S2
S0 p2 S4 h8 S2
∞, ref.
5, solved
+
8, solved
+
+
S0 h5 S3
S3 p1 S1
S3 p2 S2
S0 p2 S4
S4 h8 S2
min
2, solved
3, solved
∞, solved
1, ref.
S0 p3 S7 p4 S3
5, solved
+
S0 p3 S7
S7 p4 S3
2, solved
3, solved

Search: AO*

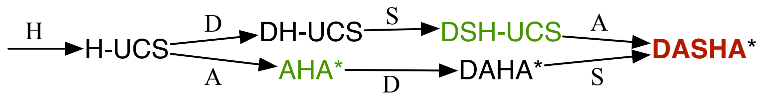


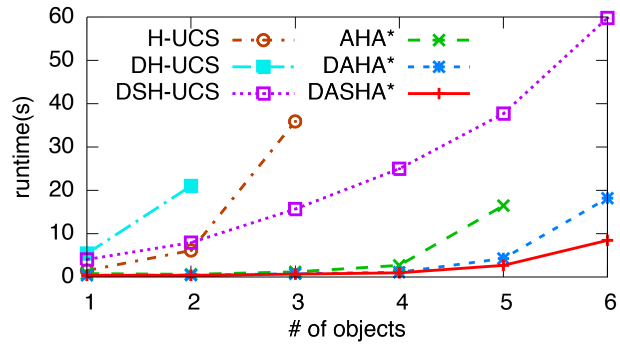
Solution: $p_3\ p_4\ p_1\ p_4$

Properties of “singleton” DASH-A*

- hierarchically optimal
- each subproblem solved at most once
- always works on subproblem that contributes to global cost bound
- can be exponentially faster than AH-A*

Singleton DASH-A*



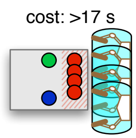


(on discrete version of mobile manipulation domain)

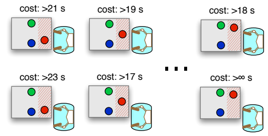
(General) DASH-A*

- What if angelic sets are not singletons?
  - Implicit sets are much more compact
  - Focusing on concrete states can break abstraction, bringing unimportant low-level details to high-level
  - Sometimes, explicit outcomes not known in advance

Implicit outcomes of GoPut

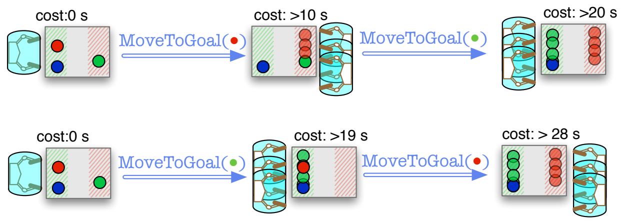


Explicit outcomes of GoPut

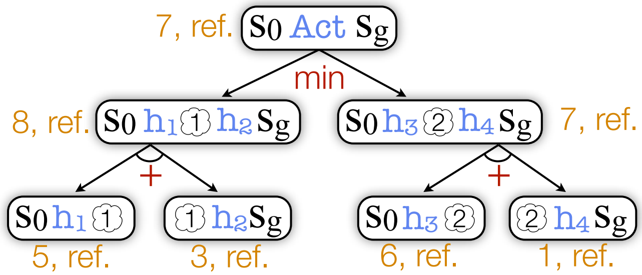


Planning with implicit sets

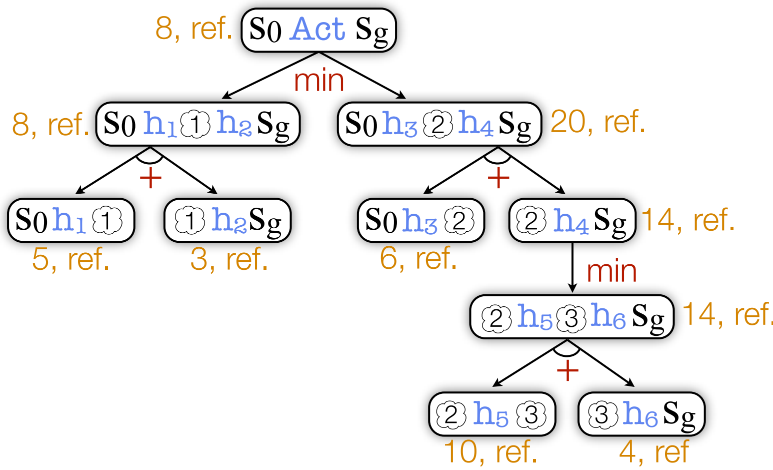
- Win: if we avoid refining a plan due to optimistic bounds being suboptimal (enough), never need to get to level of concrete states



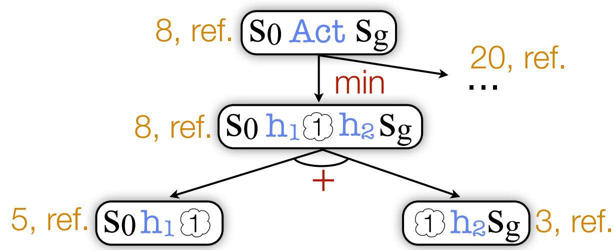
DASH-A*: first attempt



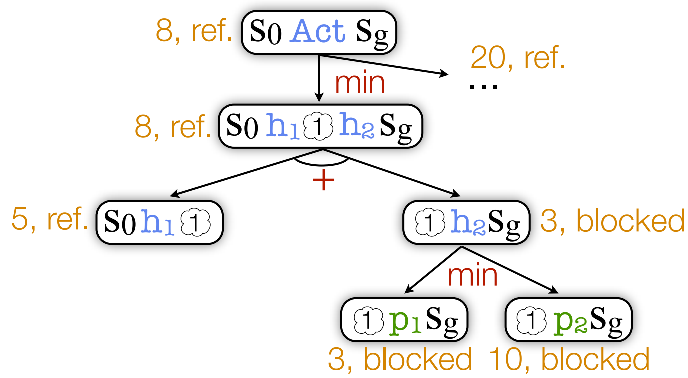
DASH-A*: looking good!



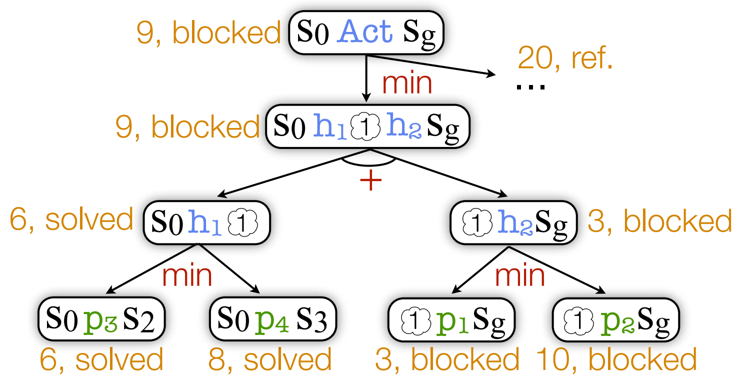
DASH-A*: looking good!



DASH-A*: looking good!

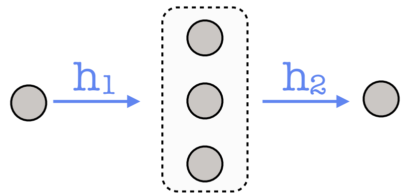


DASH-A*: what now?!



DASH-A*: challenges

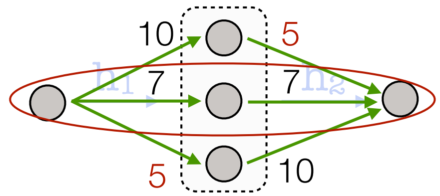
- Without concrete intermediate states, sequences do not cleanly decompose



e..g,

Implicit DASH-A*: challenges

- Without concrete intermediate states, sequences do not cleanly decompose



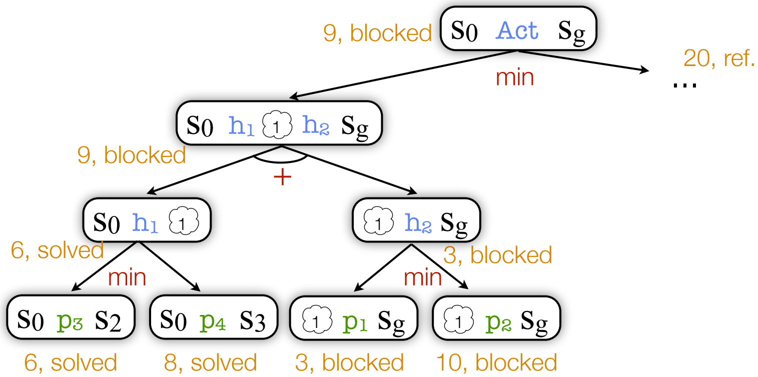
e..g,

Implicit DASH-A*: challenges

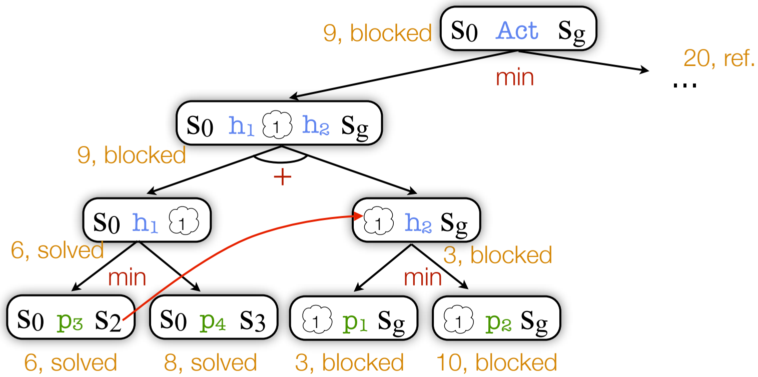
- Without concrete intermediate states, sequences do not cleanly decompose
  - must find multiple optimal solutions (to different states) for each subproblem
- As search proceeds, we must split outcome sets
  - structure of the graph changes as we go
  - splitting must propagate through later actions

e..g,

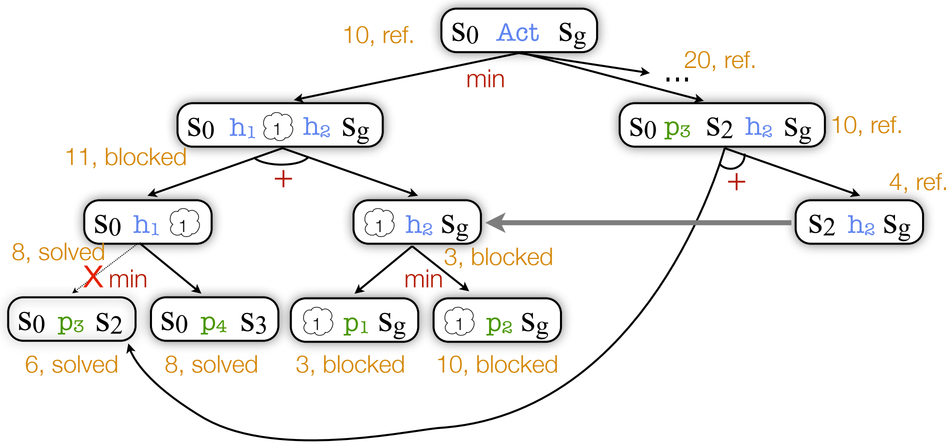
DASH-A*: specialization



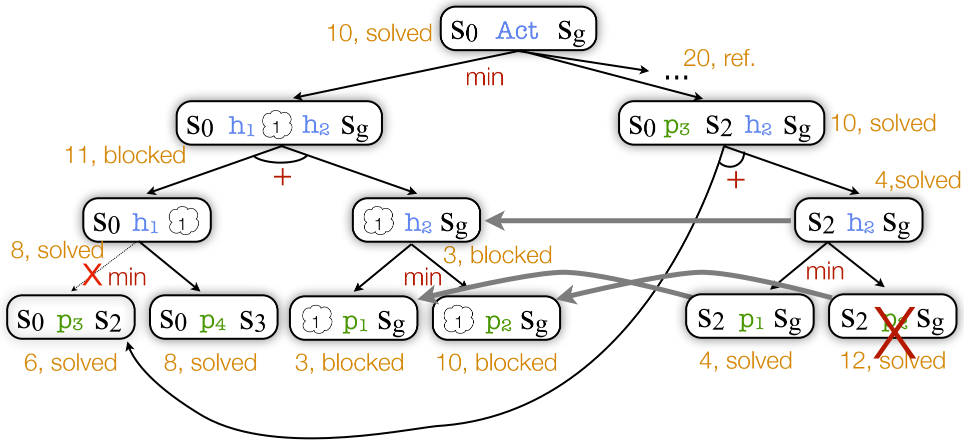
DASH-A*: specialization



DASH-A*: specialization



DASH-A*: specialization



Solution: $p_3\,p_1$

DASH-A*: Analysis and Results

- DASH-A* is systematic, hierarchically optimal
- Easy to construct examples where DASH-A* is exponentially faster than previous algorithms

domain	size	optimal len	LAMA seconds	LAMA evals	SAHTN seconds	SAHTN evals	AHA* seconds	AHA* evals	DASH-A* seconds	DASH-A* evals
witch	20x20	40			1.46	194	0.11	1017	0.23	514
	100x100	202			14.53	834	0.31	5439	0.82	1914
	500x500	1003			71.04	4034	1.74	19606	2.09	5466
te pulation	1 object	38	1.8	3667	45.73	1420	3.45	3316	0.64	213
	2 objects	53	28.52	51169	178.24	1728	8.01	5056	2.13	541
	3 objects	72	382.02	629563	656.80	1953	51.04	40752	8.02	1505
	4 objects	101					258.14	218025	21.83	3034
uous pulation	1 object	18			3.82	53	2.71	168	3.69	136
	2 objects	30			13.36	212	29.20	2473	15.17	519
	3 objects	42			17.76	319	235.28	20145	28.73	1051

: Runtimes in seconds and number of optimistic + primitive model evaluations to optimally solve random instances of omains. Results are medians over 5 instances of each size, with a memory limit of 512 MB.

Conclusion and Future work

- DASH-A* algorithm
  - Find hierarchically optimal plans
  - Decompose across subproblems
  - State abstraction to reuse solutions
  - Angelic bounds to prune search space
- Future work
  - Bounded-suboptimal DASH-A*
  - Concurrency
  - Partially observable/stochastic domains