Advances in Multiagent Decision Making under Uncertainty

Frans A. Oliehoek

Maastricht University

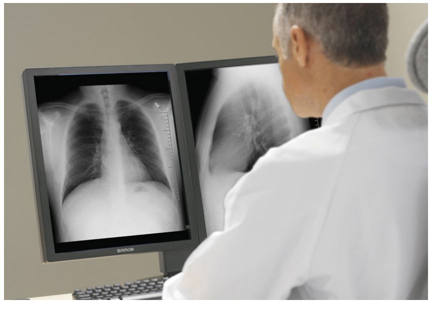
Coauthors: Matthijs Spaan (TUD), Shimon Whiteson (UvA), Nikos Vlassis (U. Luxembourg), Jilles Dibangoye (INRIA), Chris Amato (MIT)

Dynamics, Decisions & Uncertainty

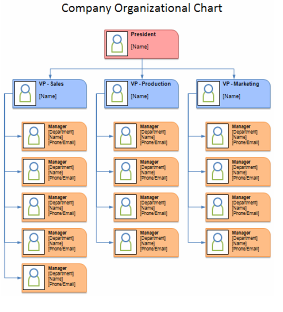
- Why care about formal decision making?







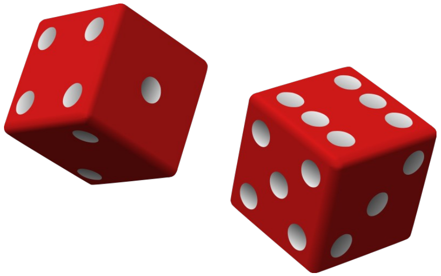


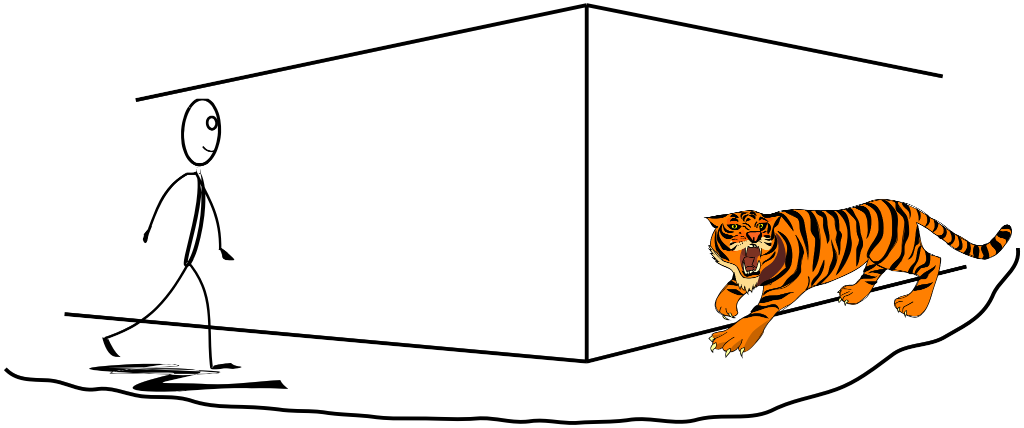


Uncertainty

- Outcome Uncertainty



- Partial Observability



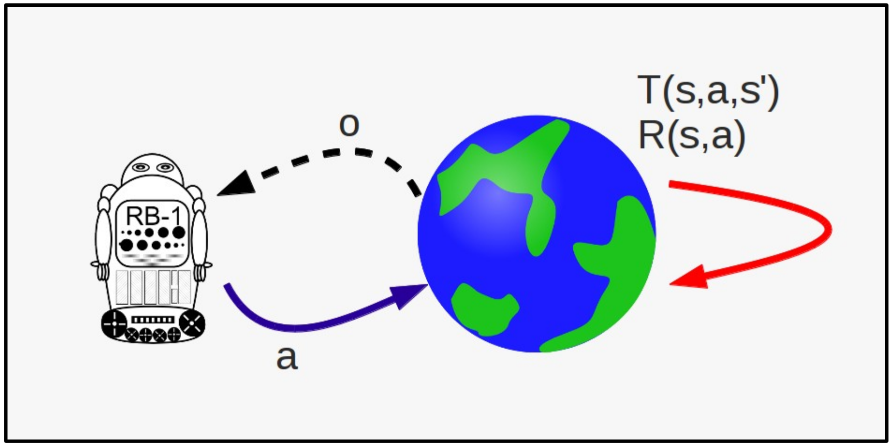
- Multiagent Systems: uncertainty about others

Outline

- Background: sequential decision making
- Optimal Solutions of Decentralized POMDPs [JAIR'13]
  - incremental clustering
  - incremental expansion
  - sufficient plan-time statistics [IJCAI'13]
- Other/current work
  - Exploiting Structure [AAMAS'13]
  - Multiagent RL under uncertainty [MSDM'13]

Background: sequential decision making

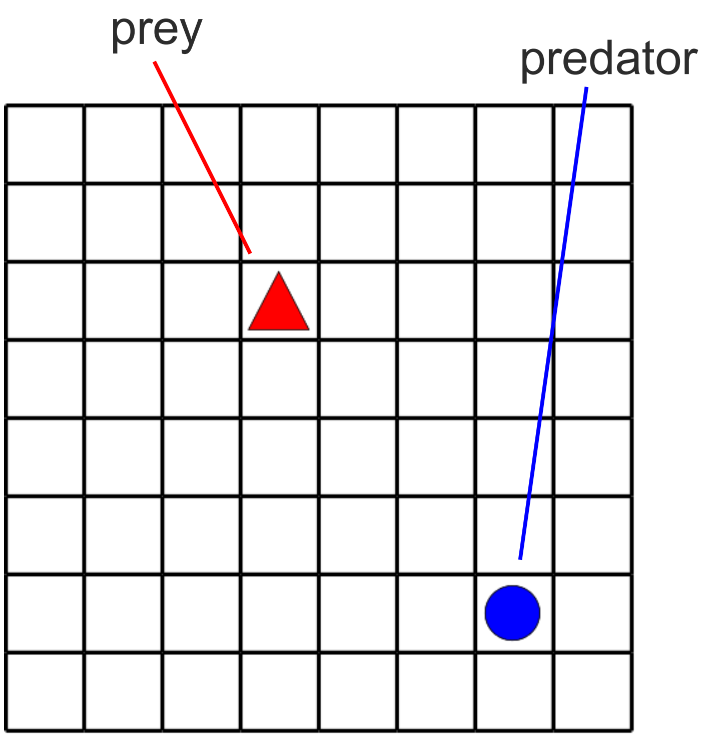
Single-Agent Decision Making



- Background: MDPs & POMDPs
- An MDP $\langle S, A, P_T, R, h \rangle$
  - S – set of states
  - A – set of actions
  - P_T – transition function $P(s'|s,a)$
  - R – reward function $R(s,a)$
  - h – horizon (finite)
- A POMDP $\langle S, A, P_T, O, P_O, R, h \rangle$
  - O – set of observations
  - P_O – observation function $P(o|a,s')$

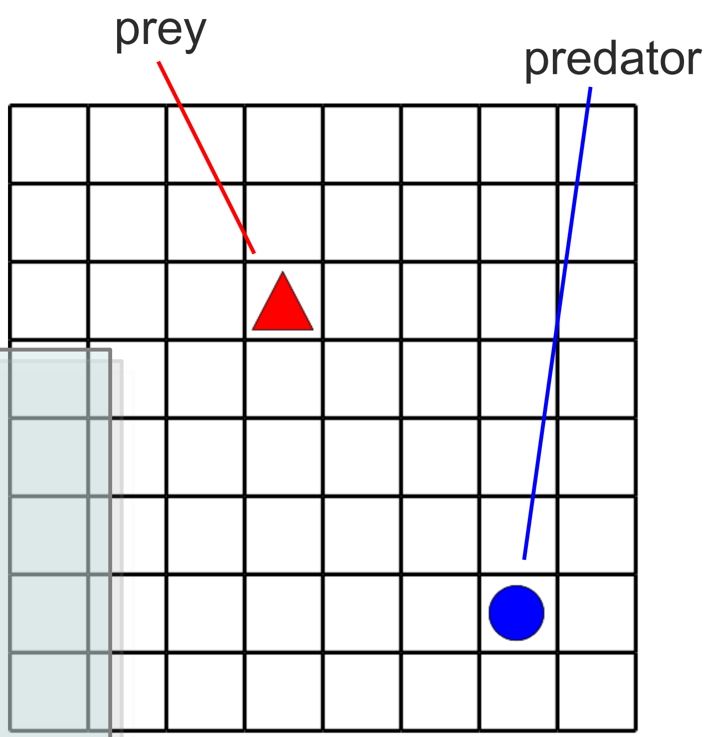
Example: Predator-Prey Domain

- Predator-Prey domain
  - 1 agent: predator
  - prey is part of environment



- Formalization:
  - states (-3,4)
  - actions N,W,S,E
  - transitions failing to move, prey moves
  - rewards reward for capturing

Example: Predator-Prey Domain



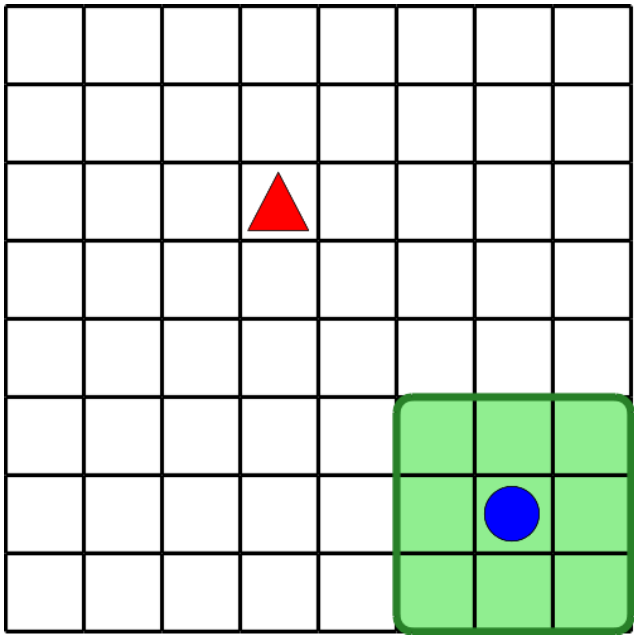
Markov decision process (MDP)

- Markovian state *s*... (which is observed!)
- policy π maps states → actions
- Value function Q(s,a)
- Compute via value iteration / policy iteration

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

Partial Observability

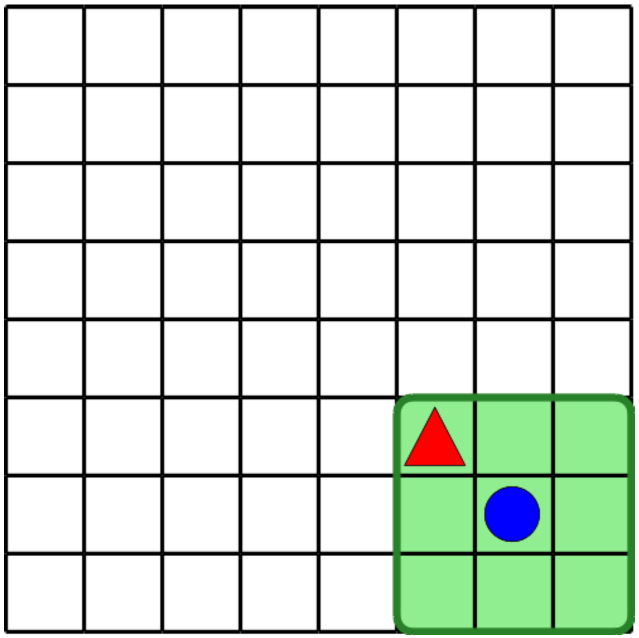
- Now: partial observability
  - E.g., limited range of sight
- MDP + observations
  - explicit observations
  - observation probabilities
    - noisy observations (detection probability)



$o = '\mathit{nothing}'$

Partial Observability

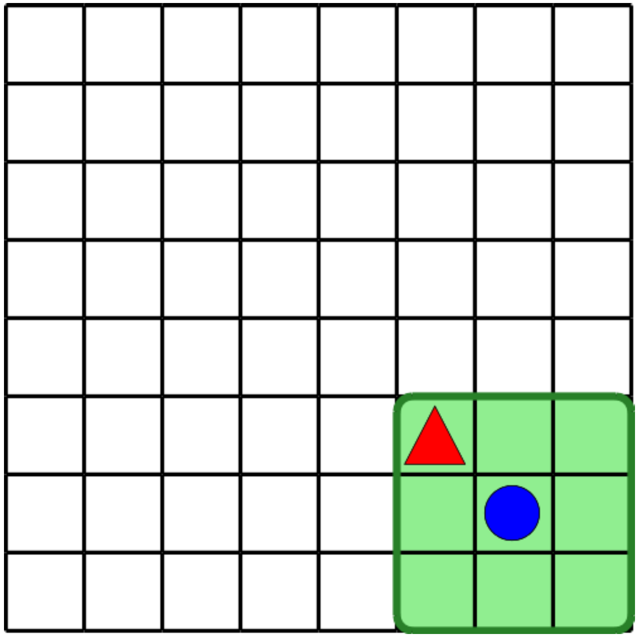
- Now: partial observability
  - E.g., limited range of sight
- MDP + observations
  - explicit observations
  - observation probabilities
    - noisy observations (detection probability)



$o=(-1,1)$

Partial Observability

- Now: partial observability
  - E.g., limited range of sight
- MDP + observations
  - explicit observations
  - observation probabilities
    - noisy observations (detection probability)



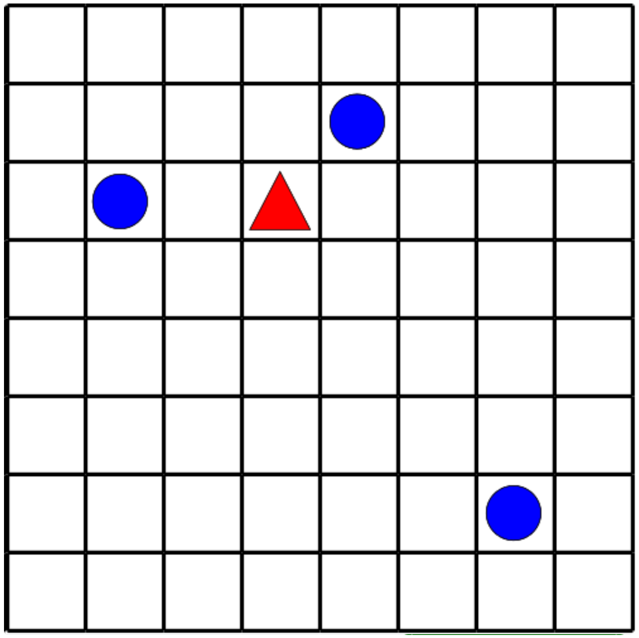
$o = (-1, 1)$

Can not observe the state
→ Need to maintain a belief over states *b(s)*
→ Policy maps beliefs to actions $\pi(b) = a$

Multiple Agents

- multiple agents, fully observable

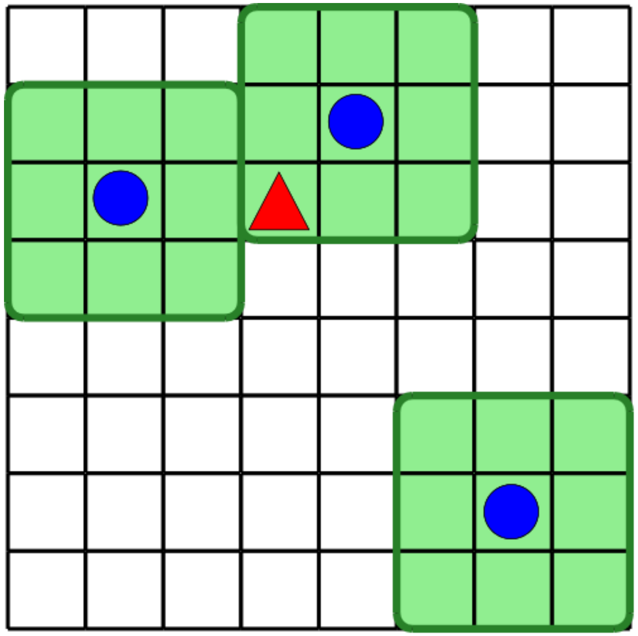
Can coordinate based upon the state
→ reduction to single agent: 'puppeteer' agent
→ takes joint action



- Formalization:
  - states ((3,-4), (1,1), (-2,0))
  - actions {N,W,S,E}
  - **joint** actions {(N,N,N), (N,N,W),...,(E,E,E)}
  - transitions probability of failing to move, prey moves
  - rewards reward for capturing jointly

Multiple Agents & Partial Observability

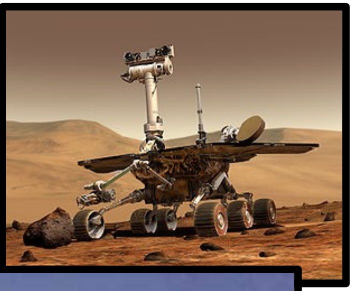
- Dec-POMDP [Bernstein et al. '02]



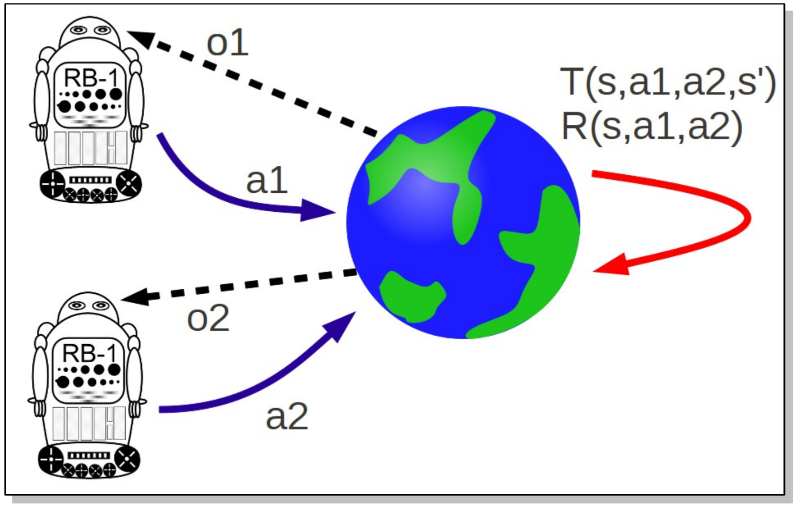
- Reduction possible
  → MPOMDP (multiagent POMDP)
  - requires broadcasting observations!
  - instantaneous, cost-free, noise-free communication → optimal [Pynadath and Tambe 2002]
  - Without such communication: no easy reduction.

Acting Based On Local Observations

- Acting on global information can be impractical:
  - communication not possible
  - significant cost (e.g battery power)
  - not instantaneous or noise free
  - scales poorly with number of agents!



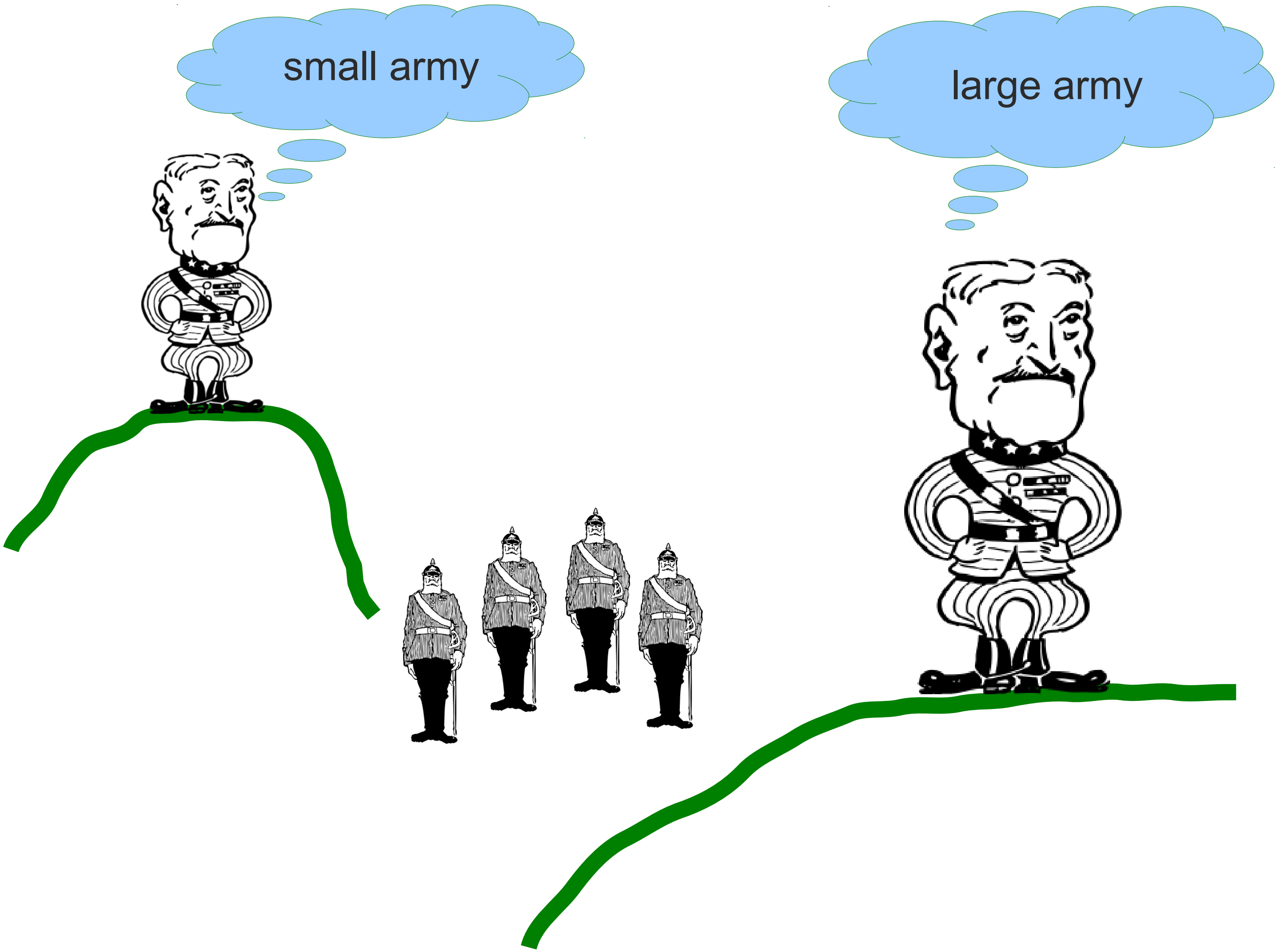
Formal Model



- A Dec-POMDP
  - $\langle S, A, P_T, O, P_O, R, h \rangle$
  - n agents
  - S – set of states
  - A – set of **joint** actions $a = \langle a_1, a_2, \ldots, a_n \rangle$
  - P_T – transition function $P(s' | s, a)$
  - O – set of **joint** observations $o = \langle o_1, o_2, \ldots, o_n \rangle$
  - P_0 – observation function $P(o | a, s')$
  - R – reward function $R(s, a)$
  - h – horizon (finite)

Running Example

- 2 generals problem



Running Example

S – { s_L, s_S }
A_i – { (O)bserve, (A)ttack }
O_i – { (L)arge, (S)mall }

Transitions

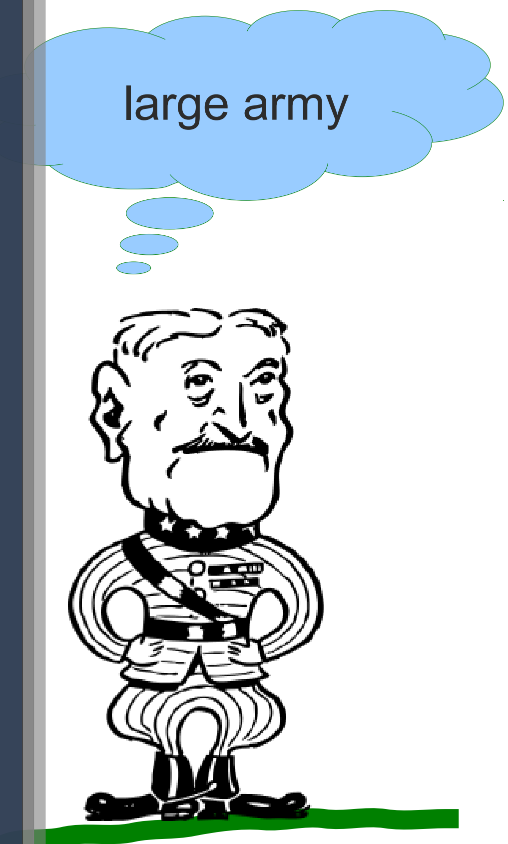
- Both Observe → no state change
- At least 1 Attack → reset (50% probability s_L, s_S)

Observations

- Probability of correct observation: 0.85
- E.g., $P(<L, L> \mid s_L) = 0.85 * 0.85 = 0.7225$

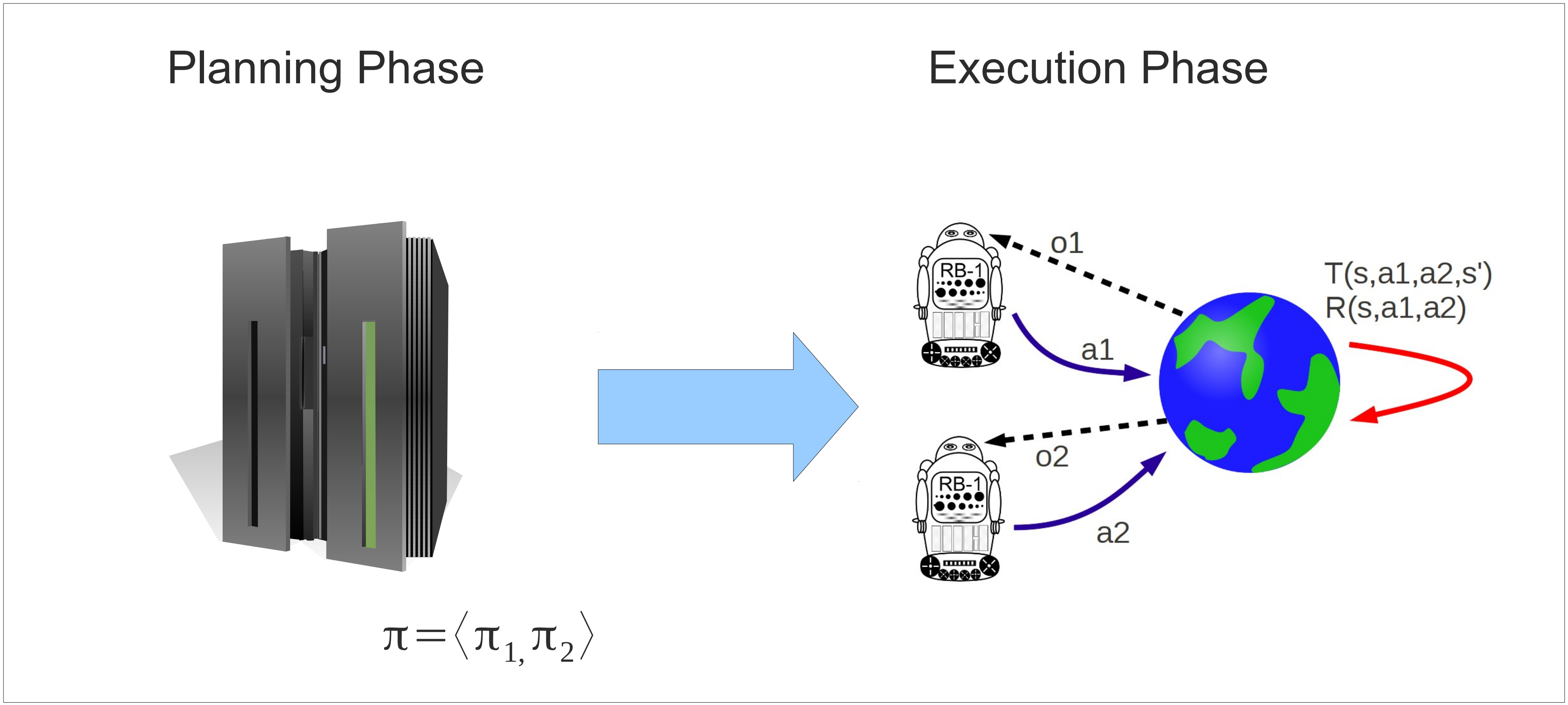
Rewards

- 1 general attacks → he loses the battle: $R(*,<A,O>) = -10$
- Both generals Observe → small cost: $R(*,<O,O>) = -1$
- Both Attack → depends on state: $R(s_L,<A,A>) = -20$
  $R(s_S,<A,A>) = +5$



Off-line / On-line phases

- off-line planning, on-line execution is decentralized

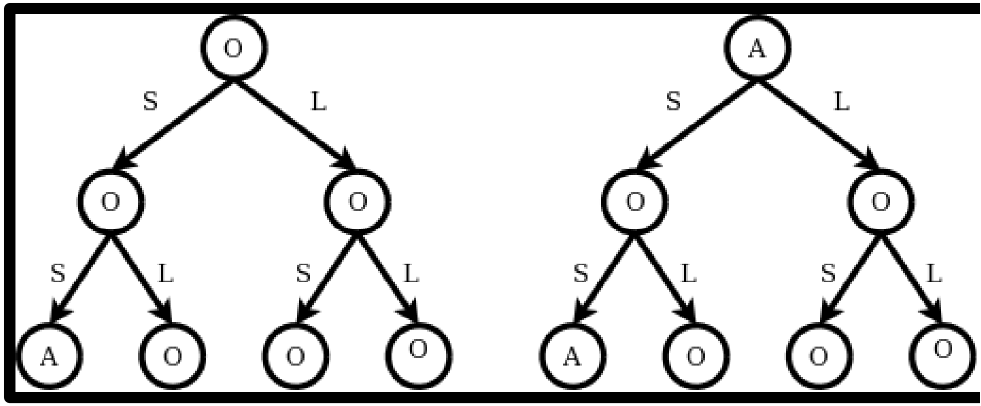


- (Smart generals make a plan in advance!)

Goal of Planning

- Find an **optimal** joint policy

$$\pi^* = \langle \pi_1, \pi_2 \rangle \qquad \pi_i : \vec{O}_i \rightarrow A_i$$



No compact representation...

- Value:
  expected sum of rewards:

$$V(\pi) = \boldsymbol{E}\left[\sum_{t=0}^{h-1} R(s,a) \mid \pi, b^0\right]$$

The problem is **NEXP-complete** [Bernstein et al. 2002]

► Also for ε-approximate solution! [Rabinovich et al. 2003]

Should we give up on optimality?

- but we care about these problems...

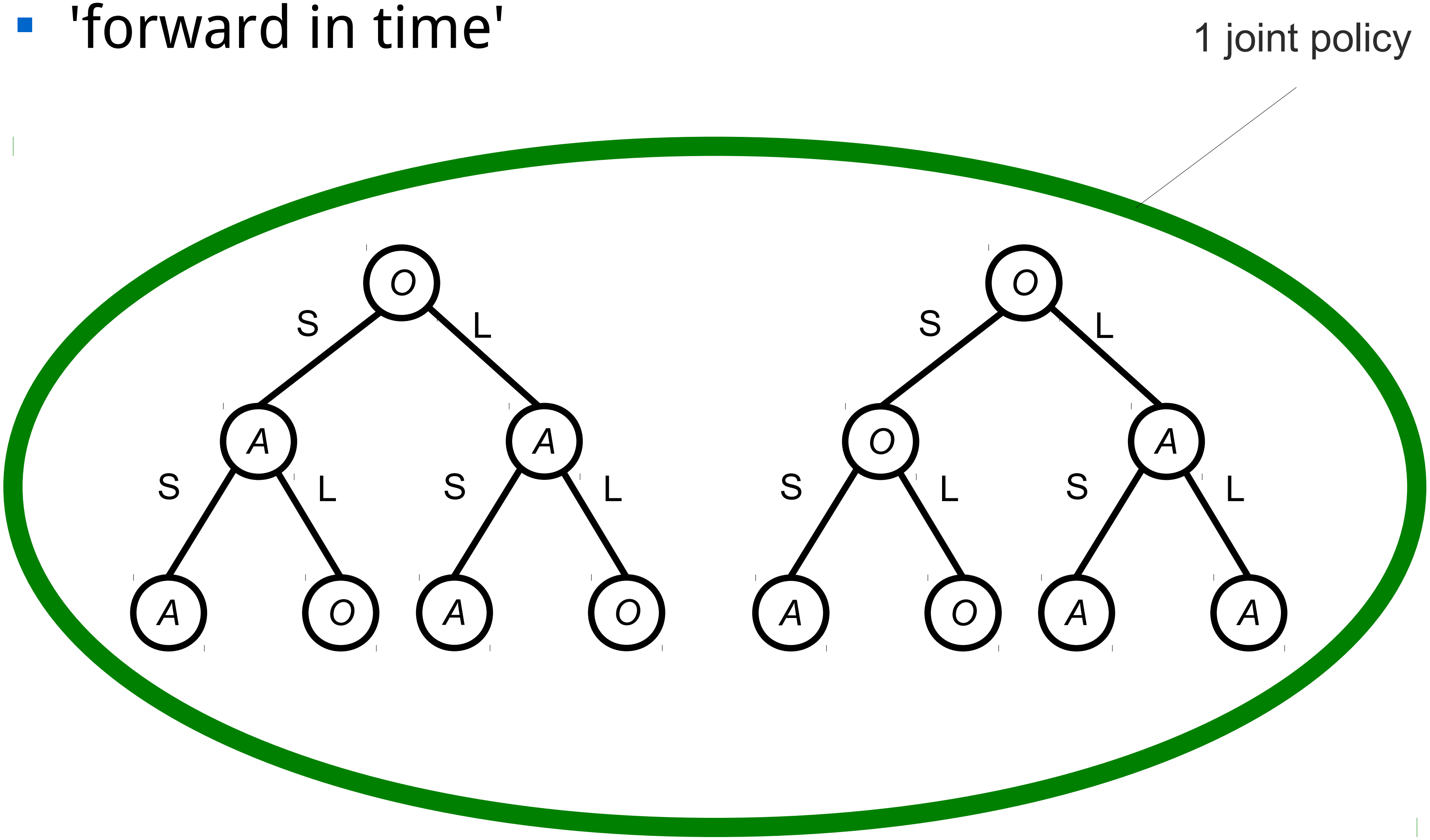
- complexity: **worst** case
  - may be able to optimally solve important problems
- optimal methods can provide **insight** in problems
- serve as inspiration for approximate methods
- need to **benchmark**: no usable upper bounds

Advances in Exact Planning Methods

- Heuristic search + limitations
- Interpret search-tree nodes as 'Bayesian Games'
- Incremental Clustering
- Incremental Expansion
- Sufficient plan-time statistics

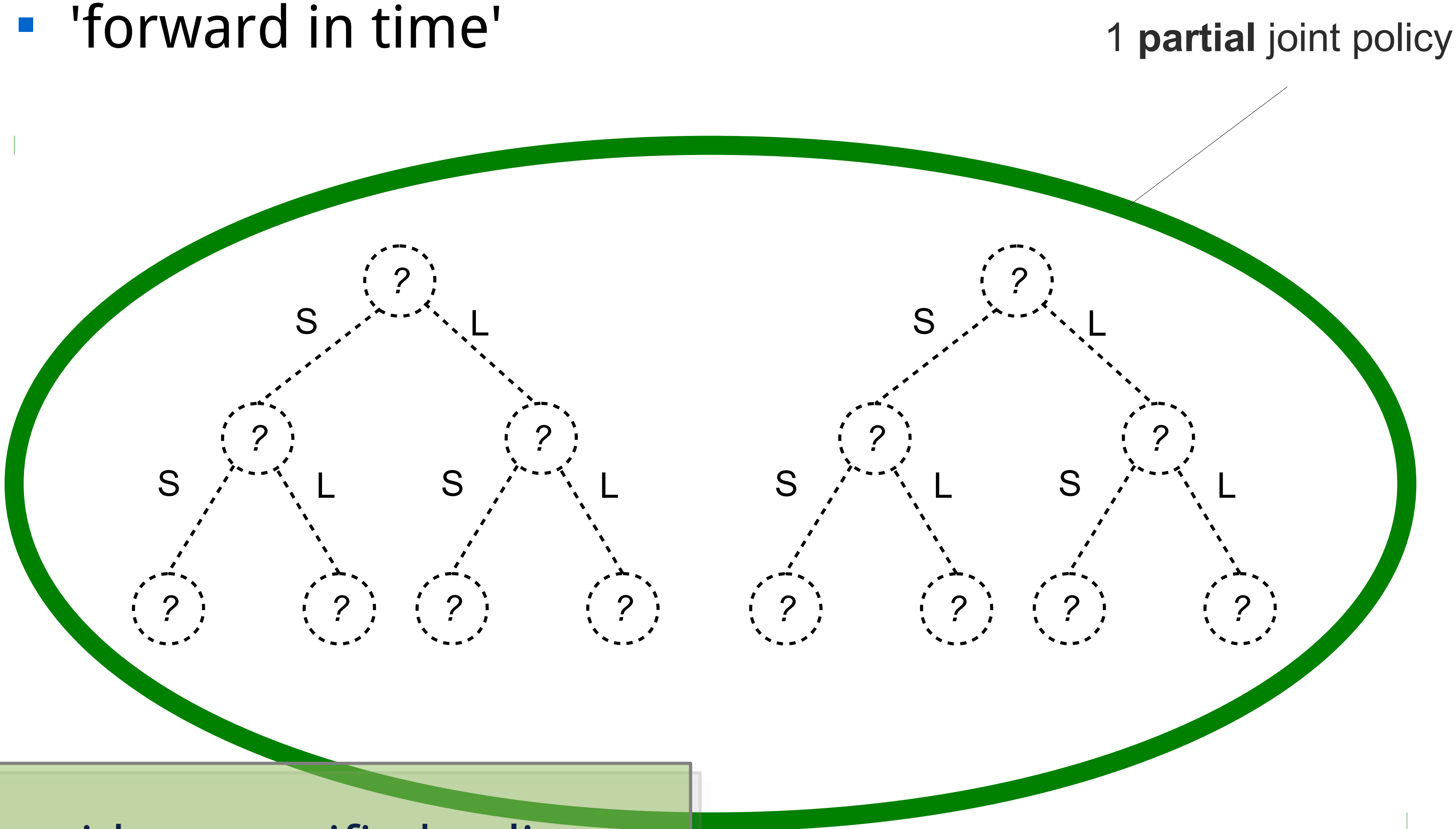
Heuristic Search – 1

- Incrementally construct all (joint) policies
  - 'forward in time'



Heuristic Search – 1

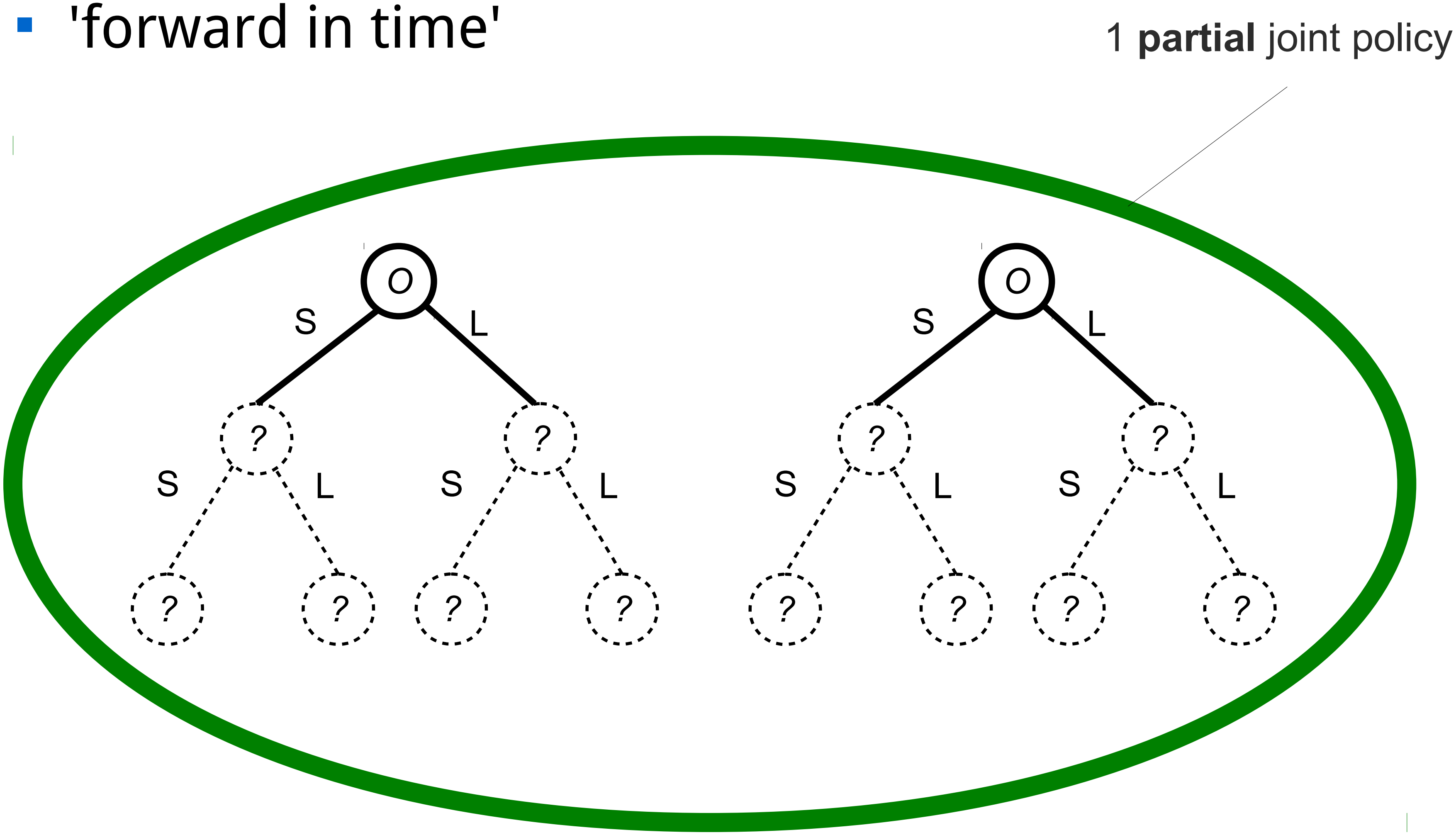
- Incrementally construct all (joint) policies
  - 'forward in time'



Start with unspecified policy

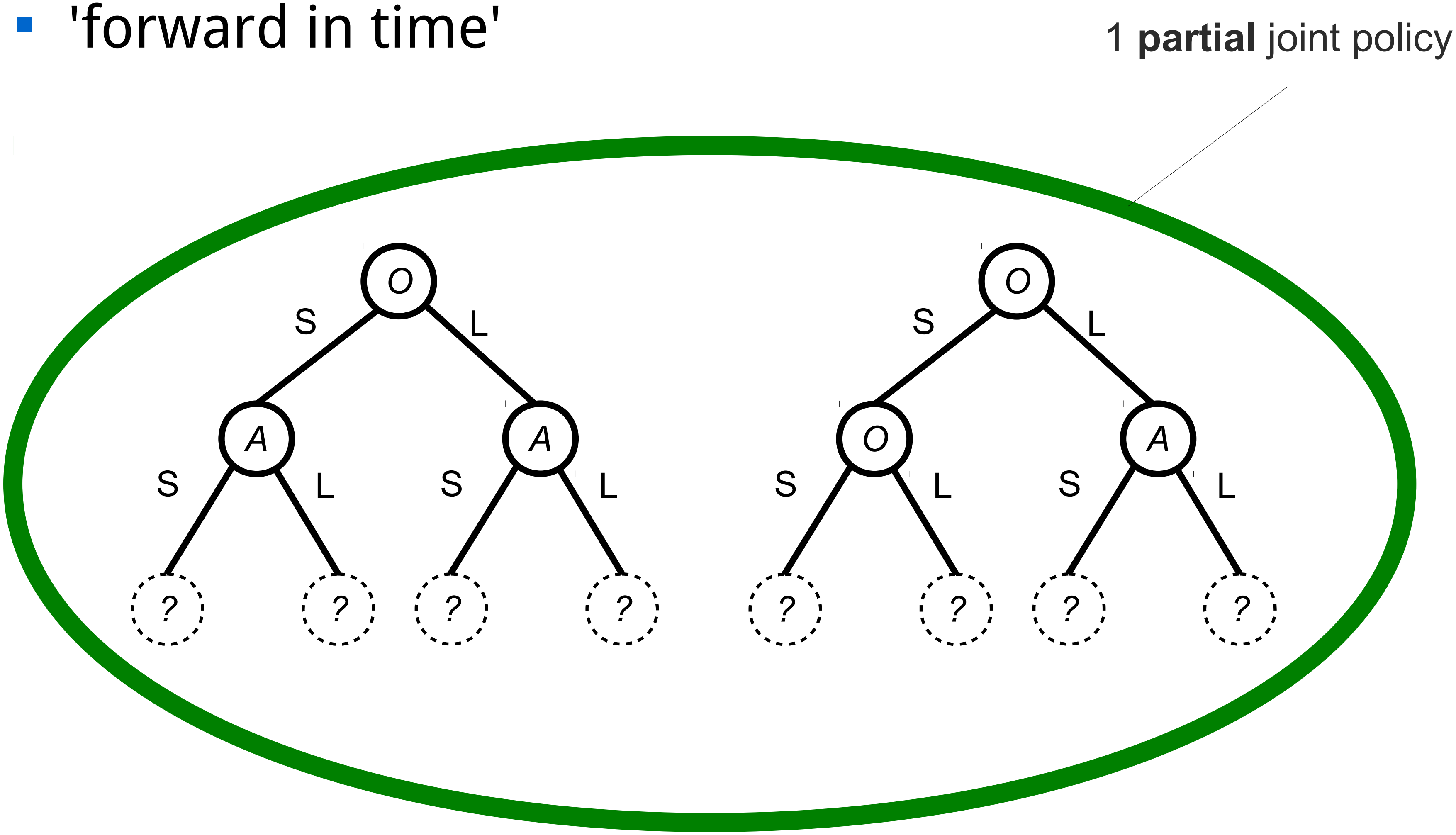
Heuristic Search – 1

- Incrementally construct all (joint) policies
  - 'forward in time'



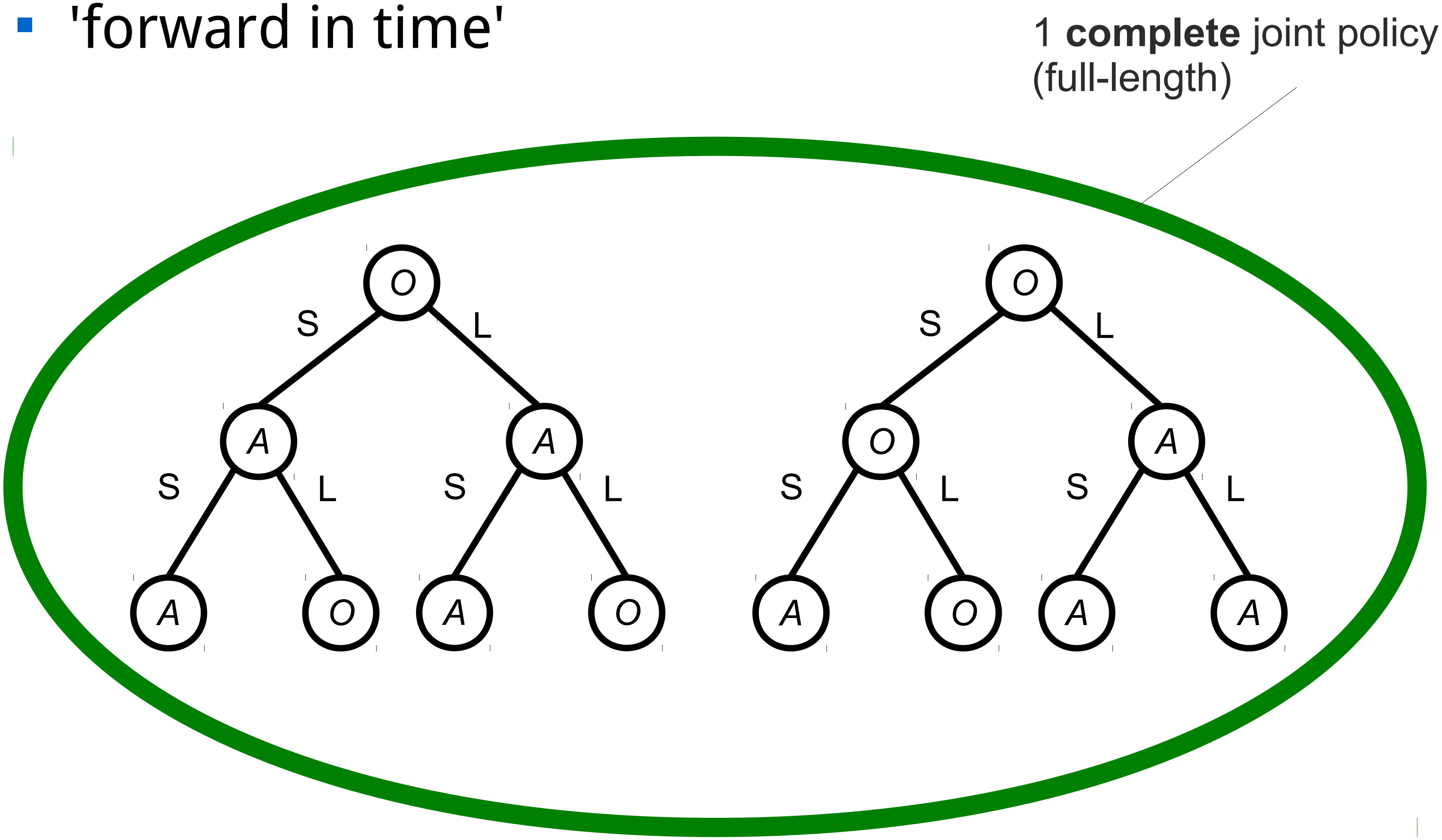
Heuristic Search – 1

- Incrementally construct all (joint) policies
  - 'forward in time'



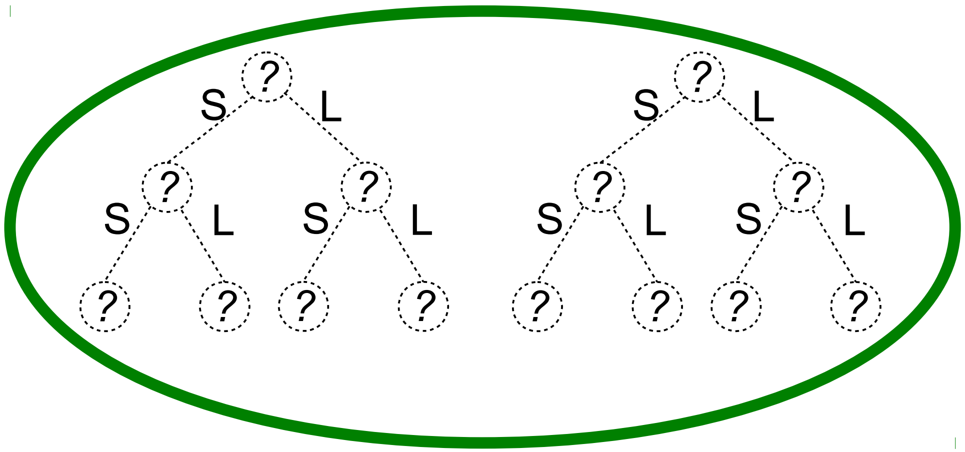
Heuristic Search – 1

- Incrementally construct all (joint) policies
  - 'forward in time'



Heuristic Search – 2

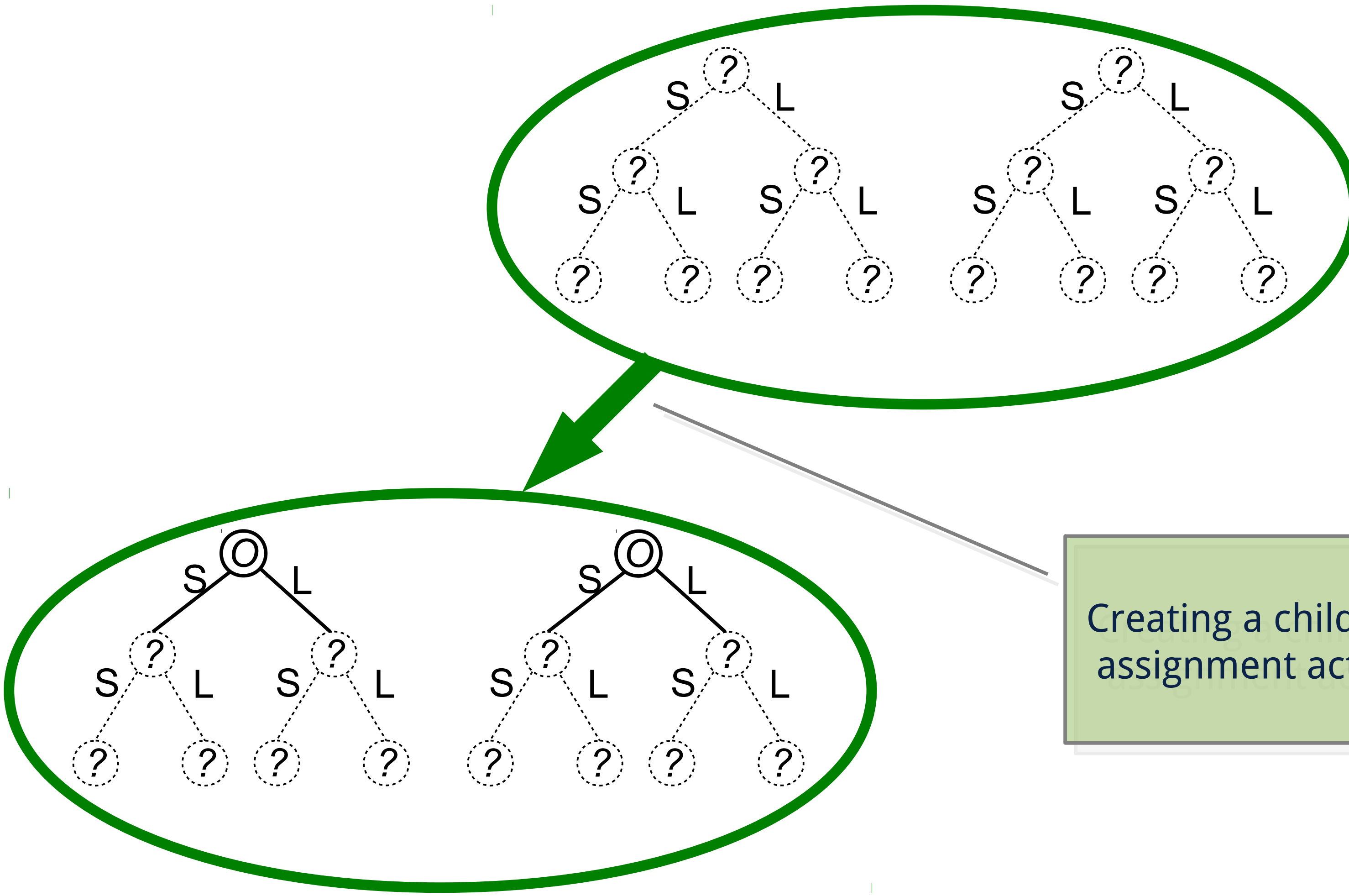
- Creating **ALL** joint policies → tree structure!



Root node:
unspecified joint policy

Heuristic Search – 2

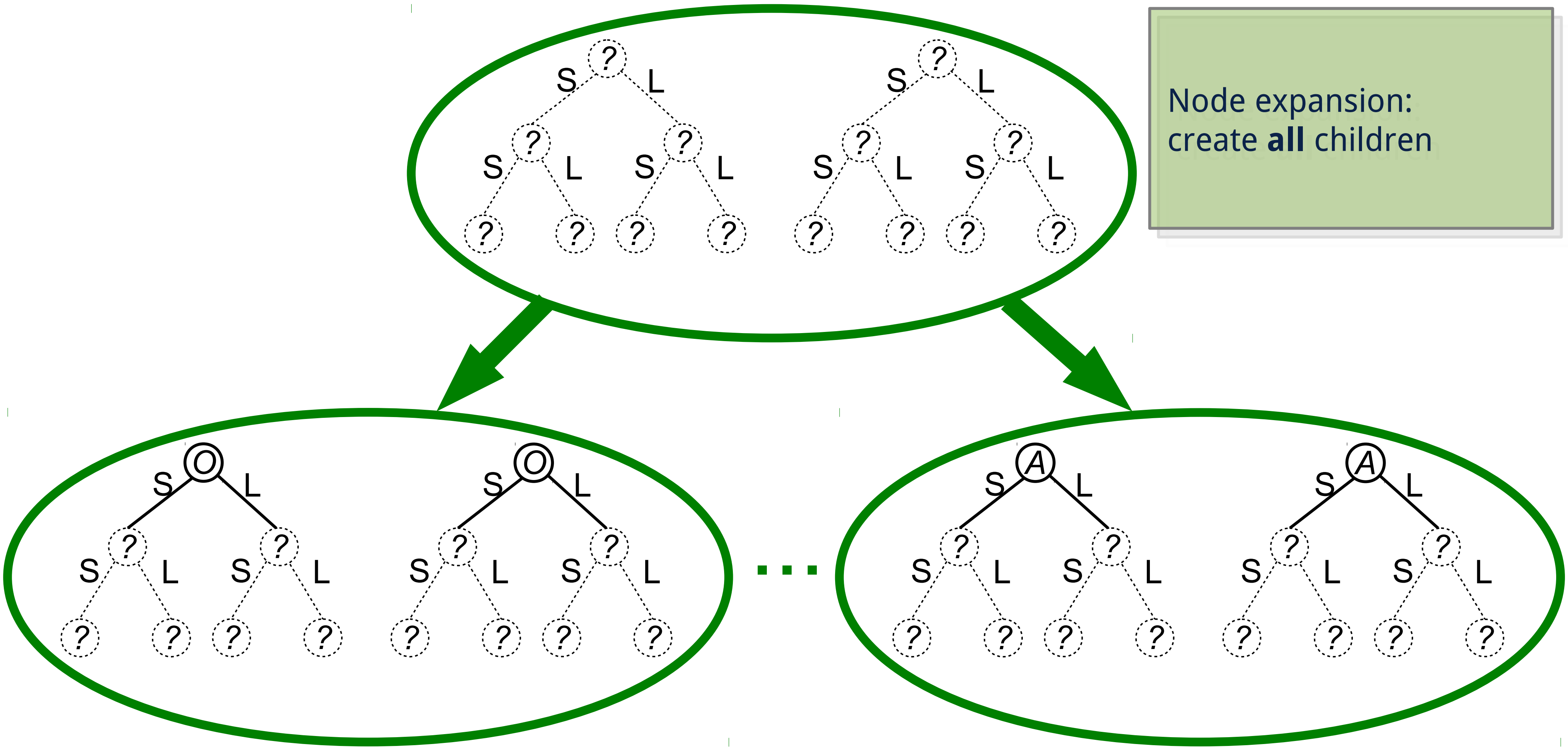
- Creating **ALL** joint policies → tree structure!



Creating a child node:
assignment actions at *t=0*

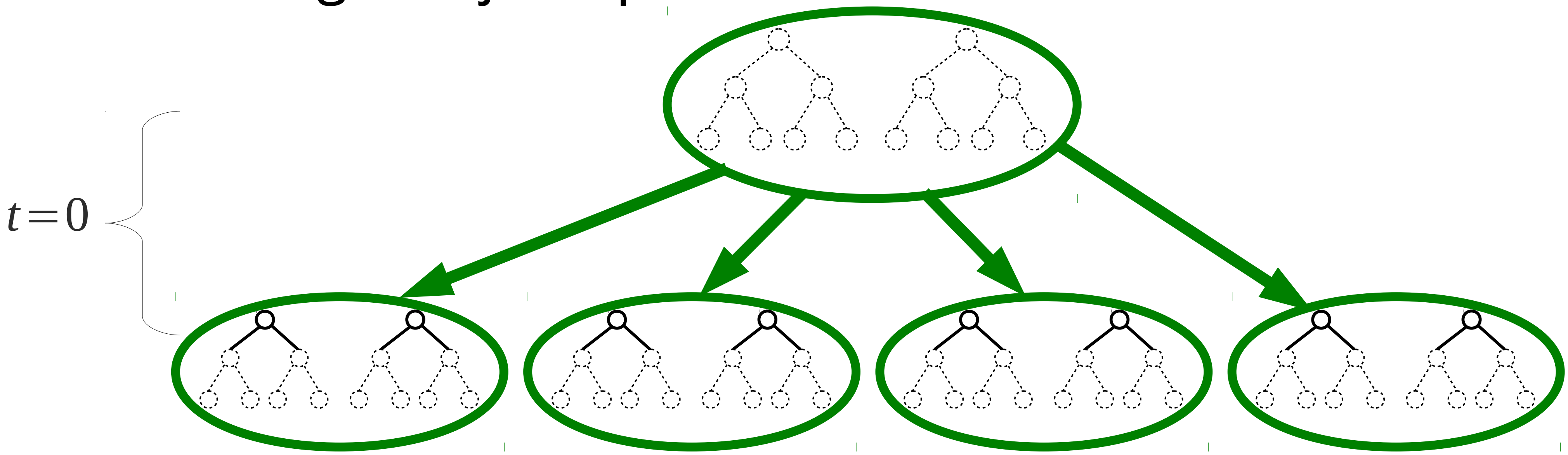
Heuristic Search – 2

- Creating **ALL** joint policies → tree structure!



Heuristic Search – 2

- Creating **ALL** joint policies → tree structure!

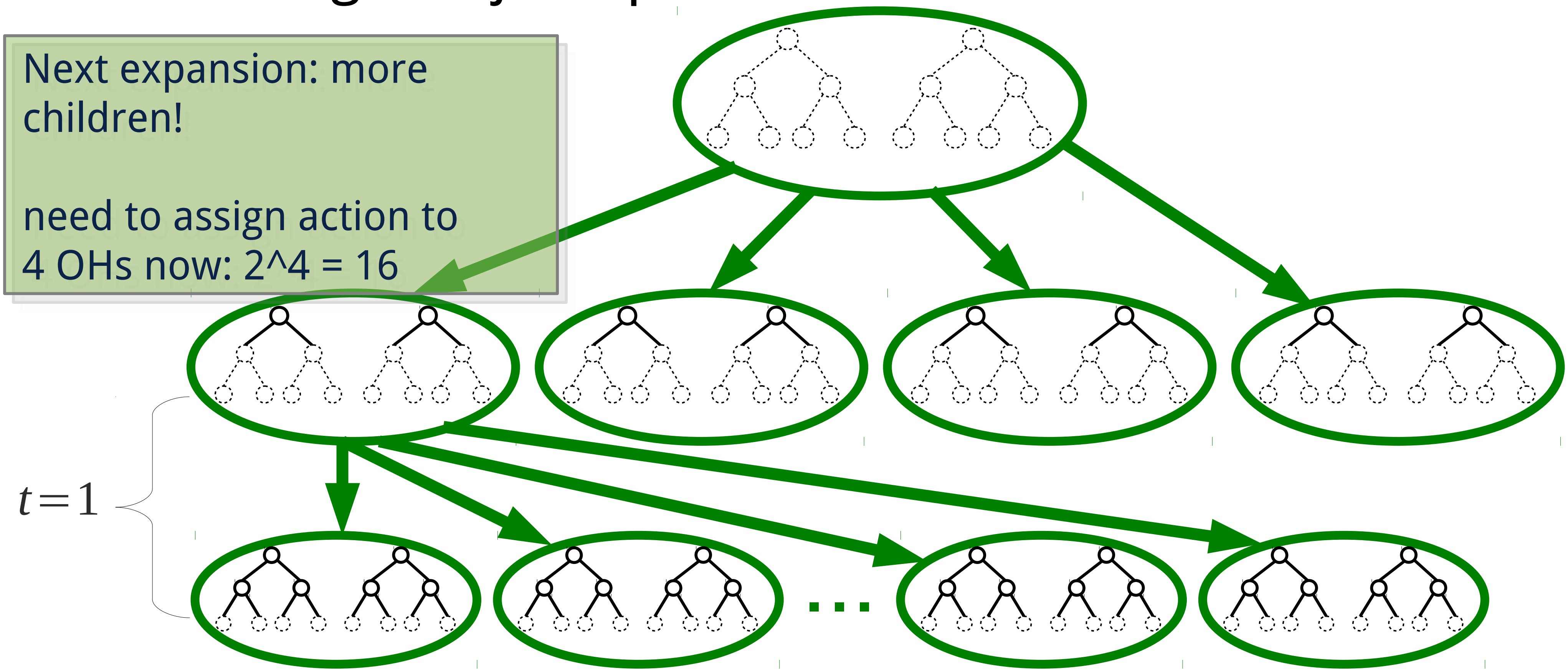


Heuristic Search – 2

- Creating **ALL** joint policies → tree structure!

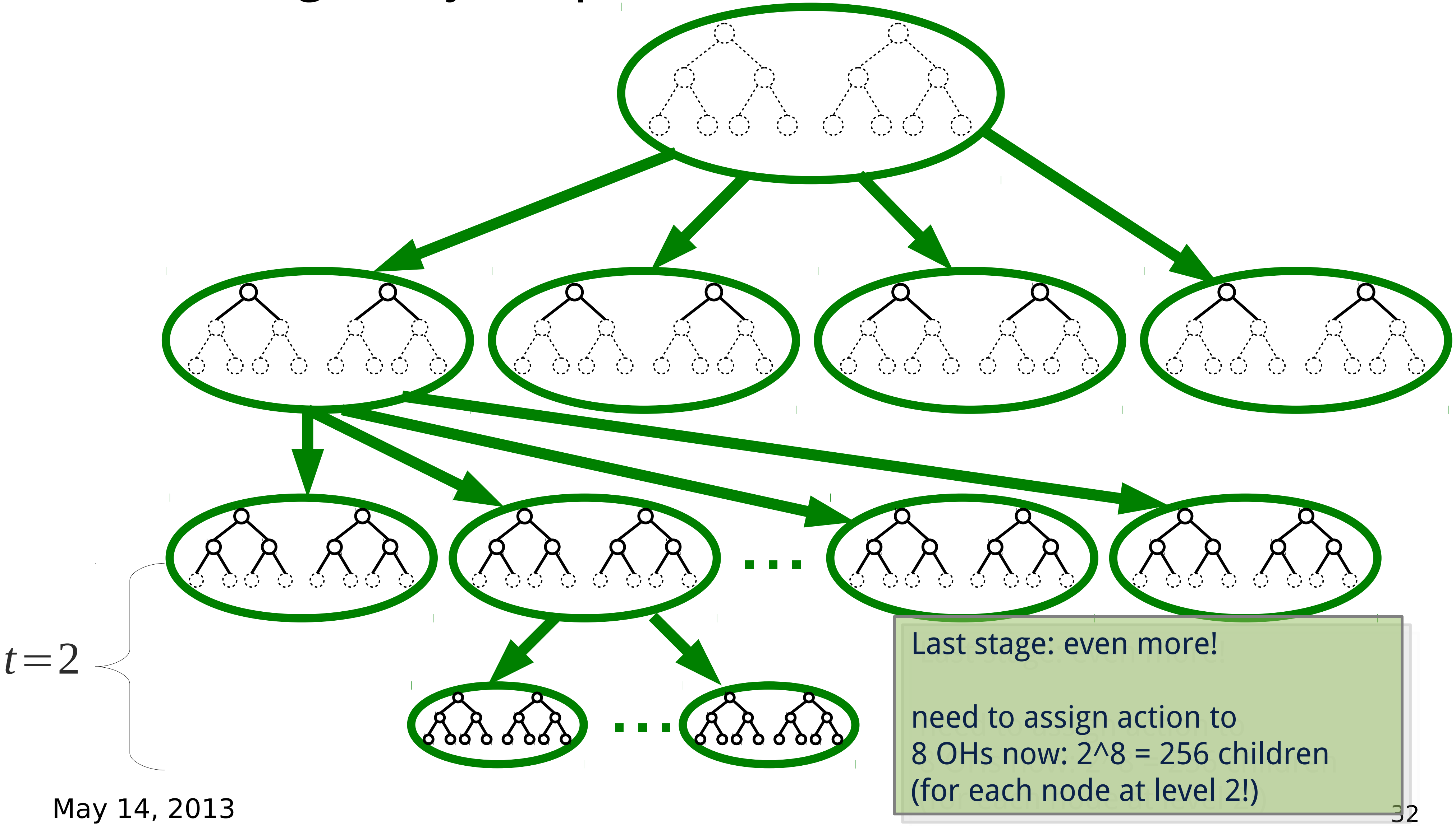
Next expansion: more children!

need to assign action to 4 OHs now: 2^4 = 16



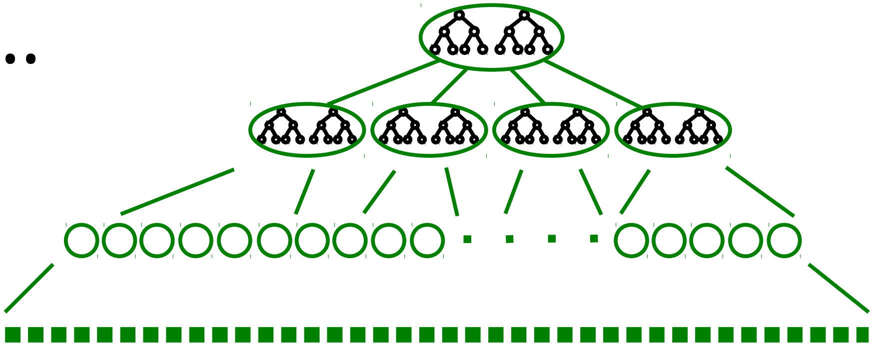
Heuristic Search – 2

- Creating **ALL** joint policies → tree structure!

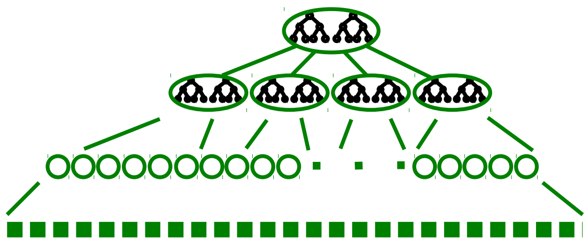


Heuristic Search – 3

- too big to create completely...
- Idea: use **heuristics**
  - avoid going down non-promising branches!
- Apply A* → **Multiagent A*** [Szer et al. 2005]



Heuristic Search – 4



- Use heuristics F(n) = G(n) + H(n)
- G(n) – actual reward of reaching n
  - a node at depth t specifies φ^t (i.e., actions for first t stages)
    → can compute $V(\varphi^t)$ over stages 0...t-1
- H(n) – should overestimate!
  - E.g., pretend that it is an MDP
  - compute

$$H(n) = H(\varphi^t) = \sum_s P(s|\varphi^t, b^0)\, \hat{V}_{MDP}(s)$$

Heuristics

- QPOMDP: Solve 'underlying POMDP'
  - corresponds to immediate communication

$$H(\varphi^t) = \sum_{\vec{\theta}^t} P(\vec{\theta}^t | \varphi^t, b^0) \hat{V}_{POMDP}(b^{\vec{\theta}^t})$$

- QBG corresponds to 1-step delayed communication
- Hierarchy of upper bounds [Oliehoek et al. 2008]

$$Q^* \leq \hat{Q}_{kBG} \leq \hat{Q}_{BG} \leq \hat{Q}_{POMDP} \leq \hat{Q}_{MDP}$$

MAA* Limitations

- Number of children grows **doubly exponentially** with nodes depth
  - For a node last stage, number of children: $O\left(|A_*|^{n|O_*|^{h-1}}\right)$
  - Total number of joint policies: $O\left(|A_*|^{(n|O_*|^{h}-1)/(|O_*|-1)}\right)$

→ MAA* can only solve 1 horizon longer than brute force search... [Seuken & Zilberstein '08]

- We introduce methods to fix this

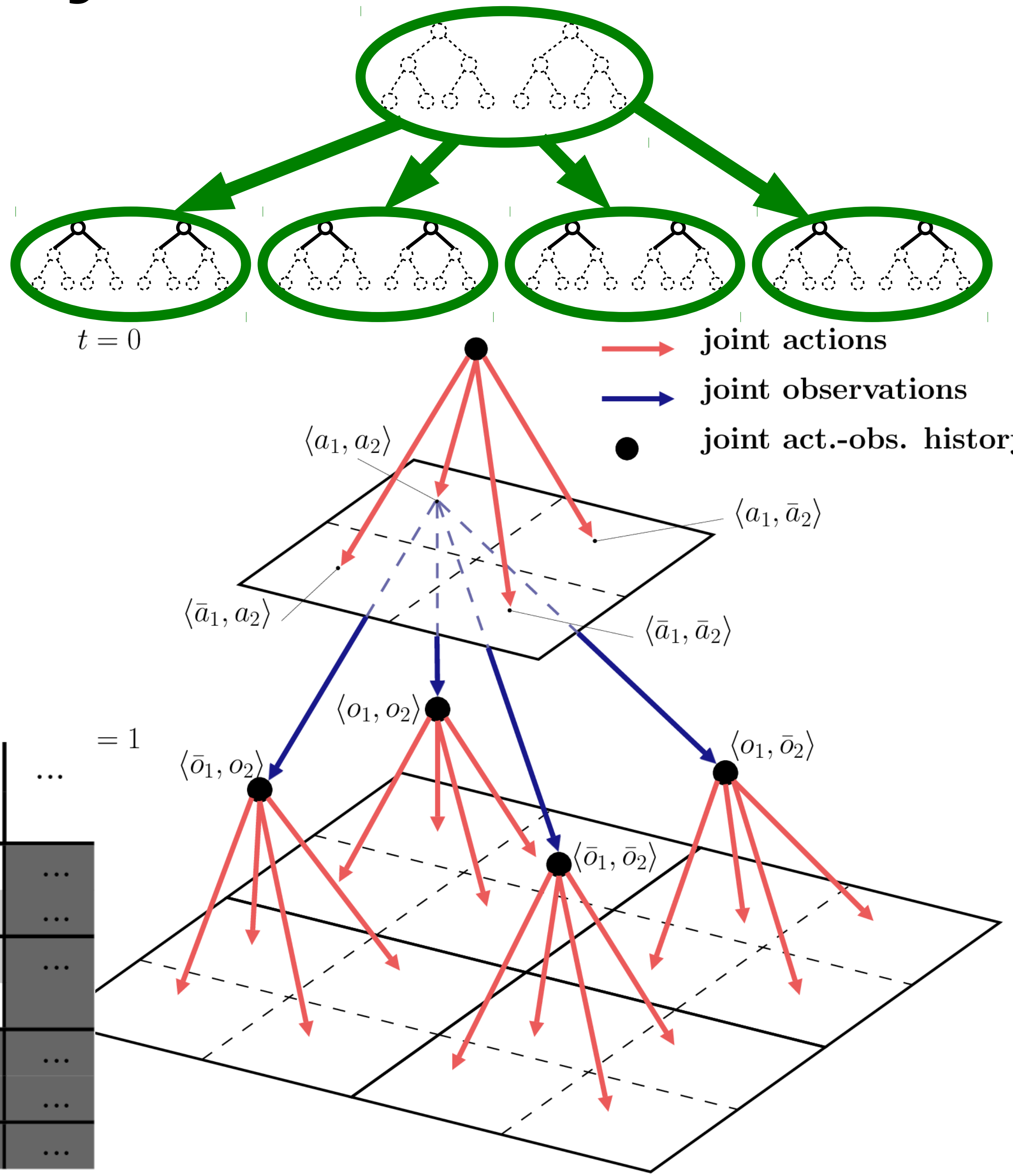
Collaborative Bayesian Games

		small		large	
		A	O	A	O
small	A	+2	-1	...	...
	O	...	...	...	...
large	A	...	...	...	...
	O	...	...	...	...

- agents, actions
- types $\theta_i \leftrightarrow$ histories
- probabilities: $P(\theta)$
- payoffs: $Q(\theta,a)$

MAA* via Bayesian Games

- Each node ↔ a φ^t
- decision problem for stage t

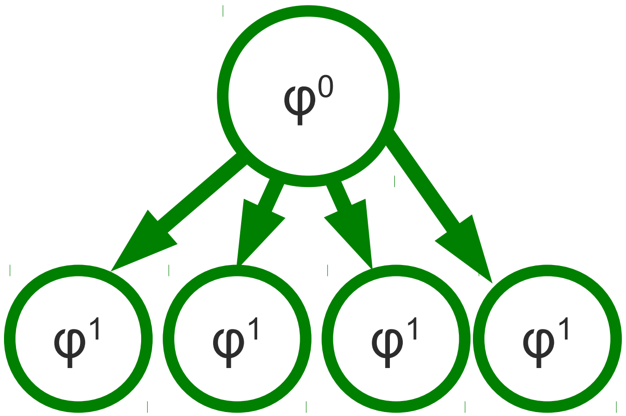


$\vec{\theta}_1^{t=0}$ \ $\vec{\theta}_2^{t=0}$		()	
		a_2	$\bar{a}_2$
()	a_1	+2.75	−4.1
	$\bar{a}_1$	−0.9	+0.3

$\vec{\theta}_1^{t=1}$ \ $\vec{\theta}_2^{t=1}$		(a_2, o_2)		$(a_2, \bar{o}_2)$		...
		a_2	$\bar{a}_2$	a_2	$\bar{a}_2$	
(a_1, o_1)	a_1	−0.3	+0.6	−0.6	+4.0	...
	$\bar{a}_1$	−0.6	+2.0	−1.3	+3.6	...
$(a_1, \bar{o}_1)$	a_1	+3.1	+4.4	−1.9	+1.0	...
	$\bar{a}_1$	+1.1	−2.9	+2.0	−0.4	
$(\bar{a}_1, o_1)$	a_1	−0.4	−0.9	−0.5	−1.0	...
	$\bar{a}_1$	−0.9	−4.5	−1.0	+3.5	...
$(\bar{a}_1, \bar{o}_1)$	...	...	...	...	...	...

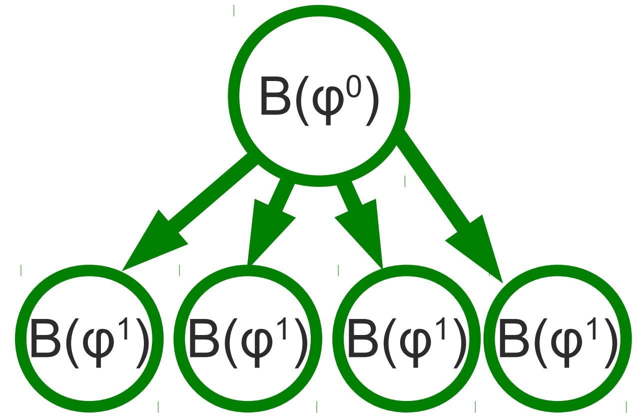
MAA* via Bayesian Games – 2

MAA* perspective



- node ↔ φ^t
- joint decision rule δ
  maps OHs to actions
- Expansion: appending all next-stage decision rules: $\varphi^{t+1}=(\varphi^t,\delta^t)$

BG perspective

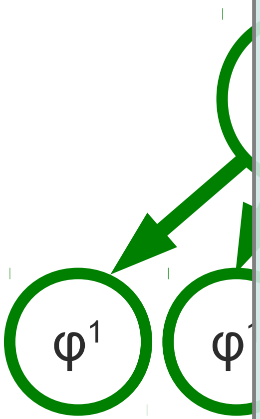


- node ↔ a BG
- joint BG policy β
  maps 'types' to actions
- Expansion: enumeration of all joint BG policies $\varphi^{t+1}=(\varphi^t,\beta^t)$

direct correspondence: $\delta \leftrightarrow \beta$

MAA* via Bayesian Games – 2

MAA* perspective



- node ↔ φ^t
- joint decision rule δ maps OHs to actions
- Expansion: appending all next-stage decision rules: $\varphi^{t+1}=(\varphi^t,\delta^t)$

BG perspective

- node ↔ a BG
- joint BG policy β maps types to actions
- Expansion: enumeration of all joint BG policies $\varphi^{t+1}=(\varphi^t,\beta^t)$

What is the point?

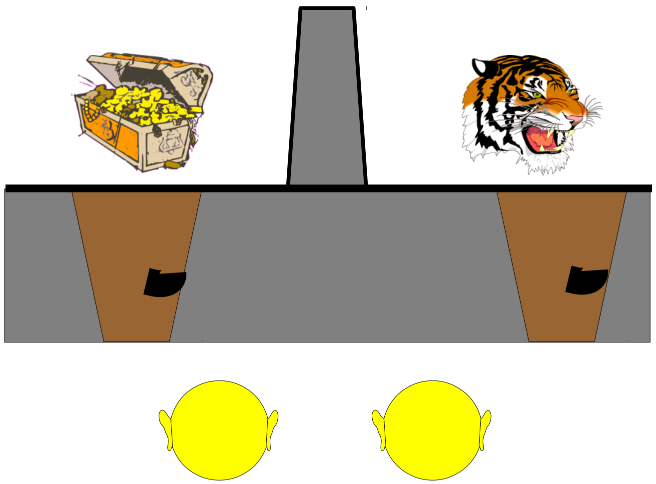
- ▶ Generalized MAA* [Oliehoek & Vlassis '07]
- ▶ Unified perspective of MAA* and 'BAGA' approximation [Emery-Montemerlo et al. '04]
- ▶ No direct improvements...

However...

- ▶ BGs provide abstraction layer
- ▶ Facilitated two improvements that lead to state-of-the-art performance [Oliehoek et al. '13]
  - Clustering of histories
  - Incremental expansion

The Decentralized Tiger Problem

- Two agents in a hallway



- States: tiger left (s_L) or right (s_R)
- Actions: listen, open left, open right
- Observations: hear left (HL), hear right (HR)
  - <Listen,Listen>
    - 85% prob. of getting right obs.
    - e.g. P(<HL,HL> | <Li,Li>, S_L) = 0.85*0.85 = 0.7225
  - otherwise: uniform random
- Reward: get the reward, acting jointly is better

Lossless Clustering

- Two types (=action-observation histories) in a BG are **probabilistically equivalent** iff

$$P(\vec{\theta}_{-i}|\vec{\theta}_{i,a}) = P(\vec{\theta}_{-i}|\vec{\theta}_{i,b})$$

$$P(s|\vec{\theta}_{-i},\vec{\theta}_{i,a}) = P(s|\vec{\theta}_{-i},\vec{\theta}_{i,b})$$

Note: φ^t, b^0

are implicit

$\vec{o}_1^{\,2}$ \ $\vec{o}_2^{\,2}$	(o_{HL},o_{HL})	(o_{HL},o_{HR})	(o_{HR},o_{HL})	(o_{HR},o_{HR})
(o_{HL},o_{HL})	0.261	0.047	0.047	0.016
(o_{HL},o_{HR})	0.047	0.016	0.016	0.047
(o_{HR},o_{HL})	0.047	0.016	0.016	0.047
(o_{HR},o_{HR})	0.016	0.047	0.047	0.261

(a) The joint type probabilities.

$\vec{o}_1^{\,2}$ \ $\vec{o}_2^{\,2}$	(o_{HL},o_{HL})	(o_{HL},o_{HR})	(o_{HR},o_{HL})	(o_{HR},o_{HR})
(o_{HL},o_{HL})	0.999	0.970	0.970	0.5
(o_{HL},o_{HR})	0.970	0.5	0.5	0.030
(o_{HR},o_{HL})	0.970	0.5	0.5	0.030
(o_{HR},o_{HR})	0.5	0.030	0.030	0.001

(b) The induced joint beliefs. Listed is the probability $\Pr(s_l|\vec{\boldsymbol{\theta}}^2,\boldsymbol{b}^0)$ of the tiger being behind the left door.

Lossless Clustering

- Two types (=action-observation histories) in a BG are **probabilistically equivalent** iff

$$P(\vec{\theta}_{-i}|\vec{\theta}_{i,a}) = P(\vec{\theta}_{-i}|\vec{\theta}_{i,b})$$

$$P(s|\vec{\theta}_{-i},\vec{\theta}_{i,a}) = P(s|\vec{\theta}_{-i},\vec{\theta}_{i,b})$$

$\vec{o}_1^2$ \ $\vec{o}_2^2$	(o_{HL},o_{HL})	(o_{HL},o_{HR})	(o_{HR},o_{HL})	(o_{HR},o_{HR})
(o_{HL},o_{HL})	0.261	0.047	0.047	0.016
(o_{HL},o_{HR})	0.047	0.016	0.016	0.047
(o_{HR},o_{HL})	0.047	0.016	0.016	0.047
(o_{HR},o_{HR})	0.016	0.047	0.047	0.261

(a) The joint type probabilities.

$\vec{o}_1^2$ \ $\vec{o}_2^2$	(o_{HL},o_{HL})	(o_{HL},o_{HR})	(o_{HR},o_{HL})	(o_{HR},o_{HR})
(o_{HL},o_{HL})	0.999	0.970	0.970	0.5
(o_{HL},o_{HR})	0.970	0.5	0.5	0.030
(o_{HR},o_{HL})	0.970	0.5	0.5	0.030
(o_{HR},o_{HR})	0.5	0.030	0.030	0.001

(b) The induced joint beliefs. Listed is the probability $\Pr(s_l|\vec{\boldsymbol{\theta}}^2,\boldsymbol{b}^0)$ of the tiger being behind the left door.

Lossless Clustering

- Two types (=action-observation histories) in a BG are **probabilistically equivalent** iff

$$P(\vec{\theta}_{-i}|\vec{\theta}_{i,a}) = P(\vec{\theta}_{-i}|\vec{\theta}_{i,b})$$

$$P(s|\vec{\theta}_{-i},\vec{\theta}_{i,a}) = P(s|\vec{\theta}_{-i},\vec{\theta}_{i,b})$$

Clustering is lossless

restricting the policy space to clustered policies does not sacrifice optimality

- ▶histories are **best-response equivalent**
- ▶if criterion holds → same 'multiagent belief' $b_i(s,q_{-i})$

$\vec{o}_1^{\,2}$ \ $\vec{o}_2^{\,2}$	(o_{HL},o_{HL})	(o_{HL},o_{HR})	(o_{HR},o_{HL})	(o_{HR},o_{HR})
(o_{HL},o_{HL})	0.261	0.047	0.047	0.016
(o_{HL},o_{HR})	0.047	0.016	0.016	0.047
(o_{HR},o_{HL})	0.047	0.016	0.016	0.047
(o_{HR},o_{HR})	0.016	0.047	0.047	0.261

(a) The joint type probabilities.

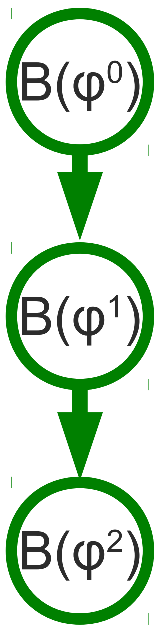
$\vec{o}_1^{\,2}$ \ $\vec{o}_2^{\,2}$	(o_{HL},o_{HL})	(o_{HL},o_{HR})	(o_{HR},o_{HL})	(o_{HR},o_{HR})
(o_{HL},o_{HL})	0.999	0.970	0.970	0.5
(o_{HL},o_{HR})	0.970	0.5	0.5	0.030
(o_{HR},o_{HL})	0.970	0.5	0.5	0.030
(o_{HR},o_{HR})	0.5	0.030	0.030	0.001

(b) The induced joint beliefs. Listed is the probability $\Pr(s_l|\vec{\boldsymbol{\theta}}^2,\boldsymbol{b}^0)$ of the tiger being behind the left door.

Incremental Clustering

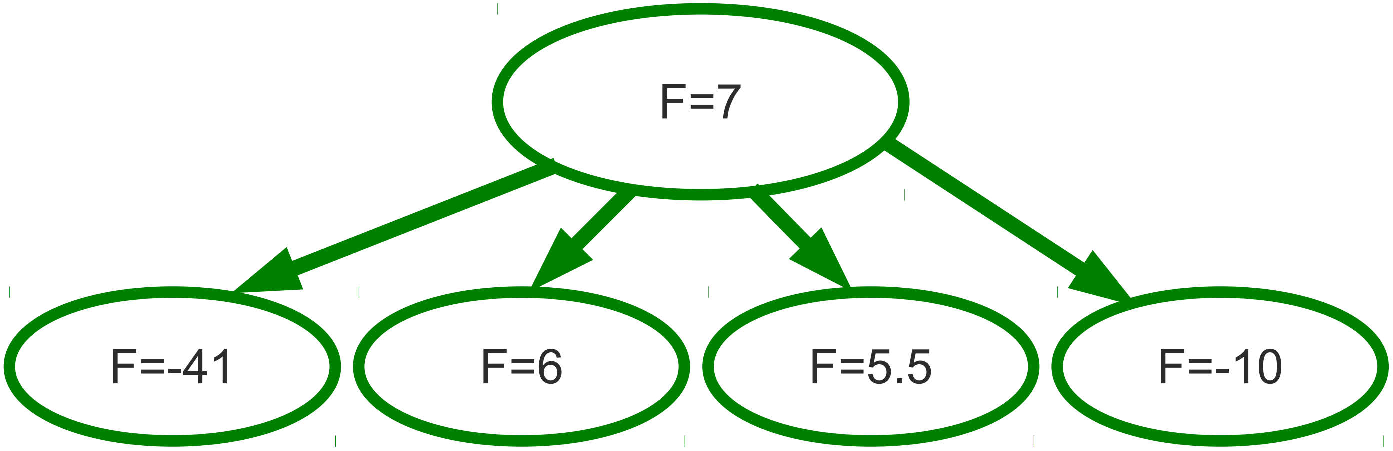
- No need to cluster from scratch
- Probabilistic equivalence 'extends forwards'
  - identical extensions of two PE histories are also PE

    → can bootstrap from CBG of the previous stage
  - 'Incremental clustering'



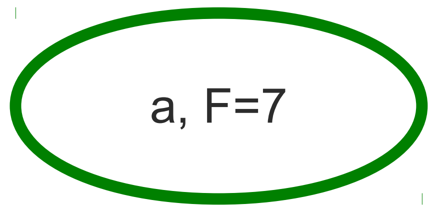
Incremental Expansion

- Key idea: nodes have many children, but only few are useful.
  - i.e., only few will be selected for further expansion
  - others will have too low heuristic value



- if we can generate the nodes in decreasing heuristic order
  → can avoid expansion of redundant nodes

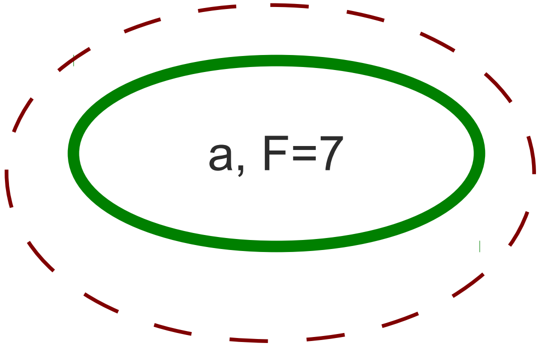
Incremental Expansion



Open list
a – 7

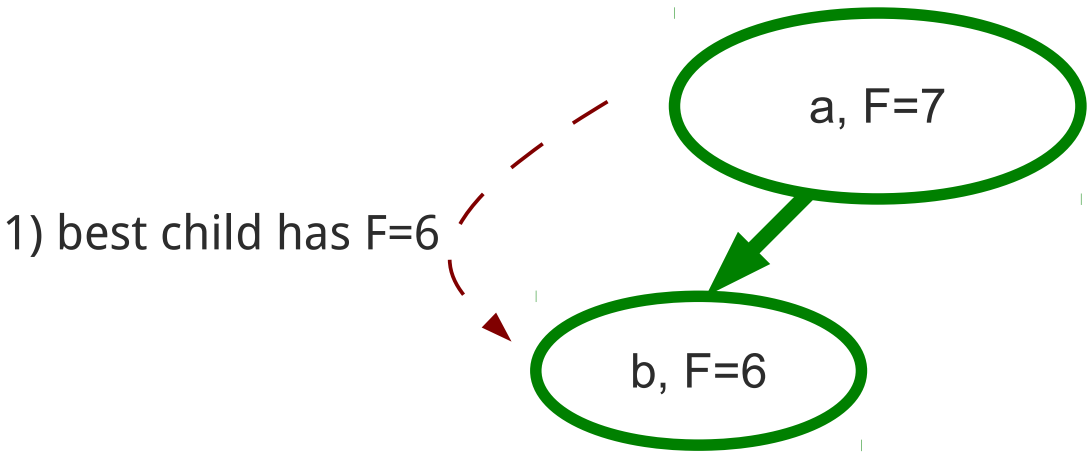
Incremental Expansion

Select for expansion →



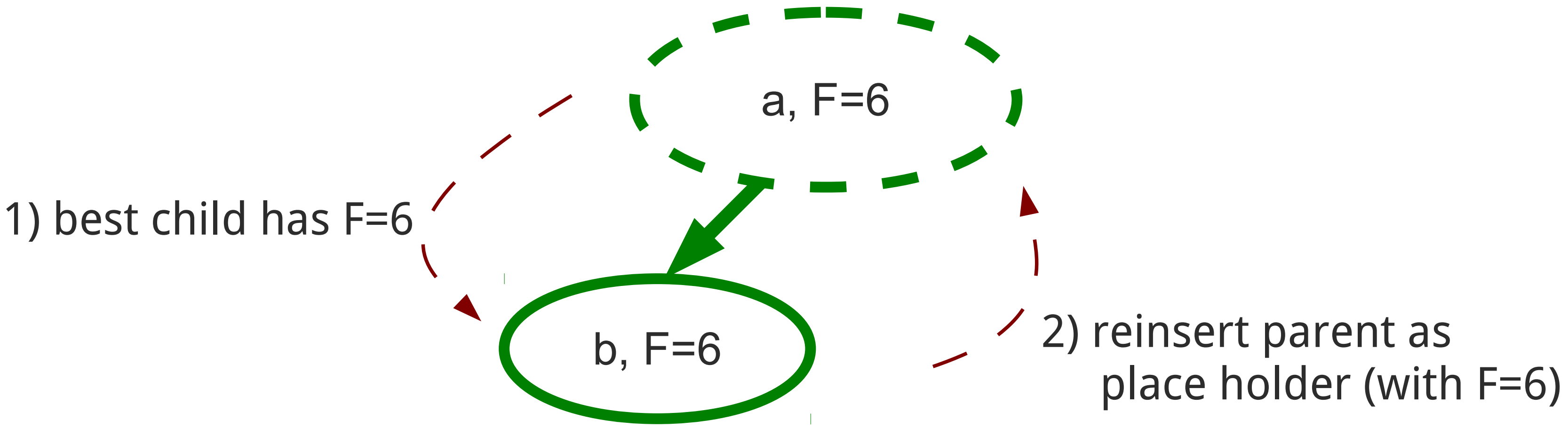
Open list
a – 7

Incremental Expansion



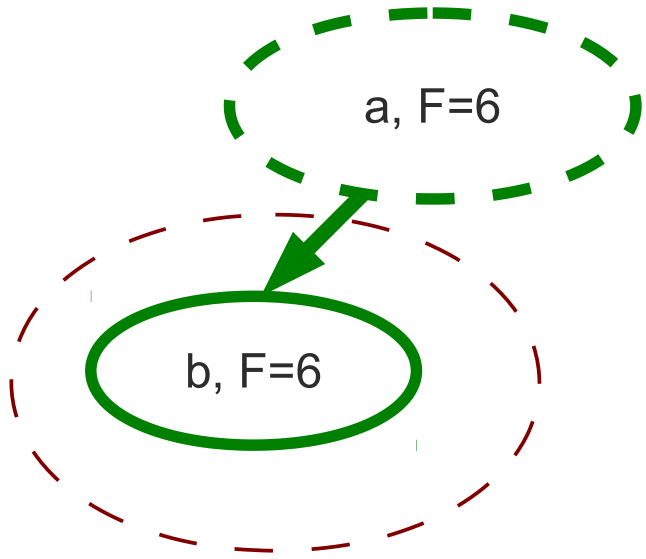
Open list
b – 6

Incremental Expansion



Open list
b – 6
a – 6

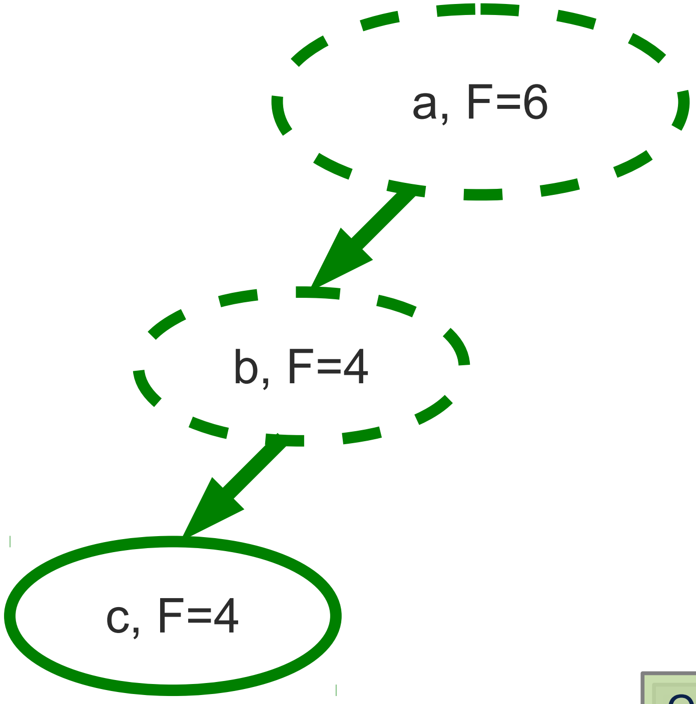
Incremental Expansion



Select for expansion →

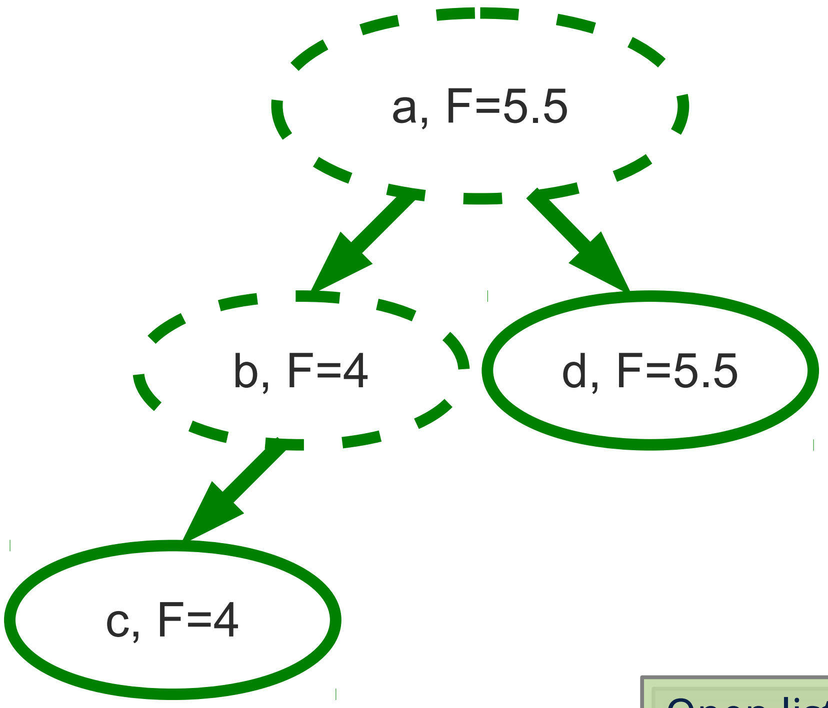
Open list
b – 6
a – 6

Incremental Expansion



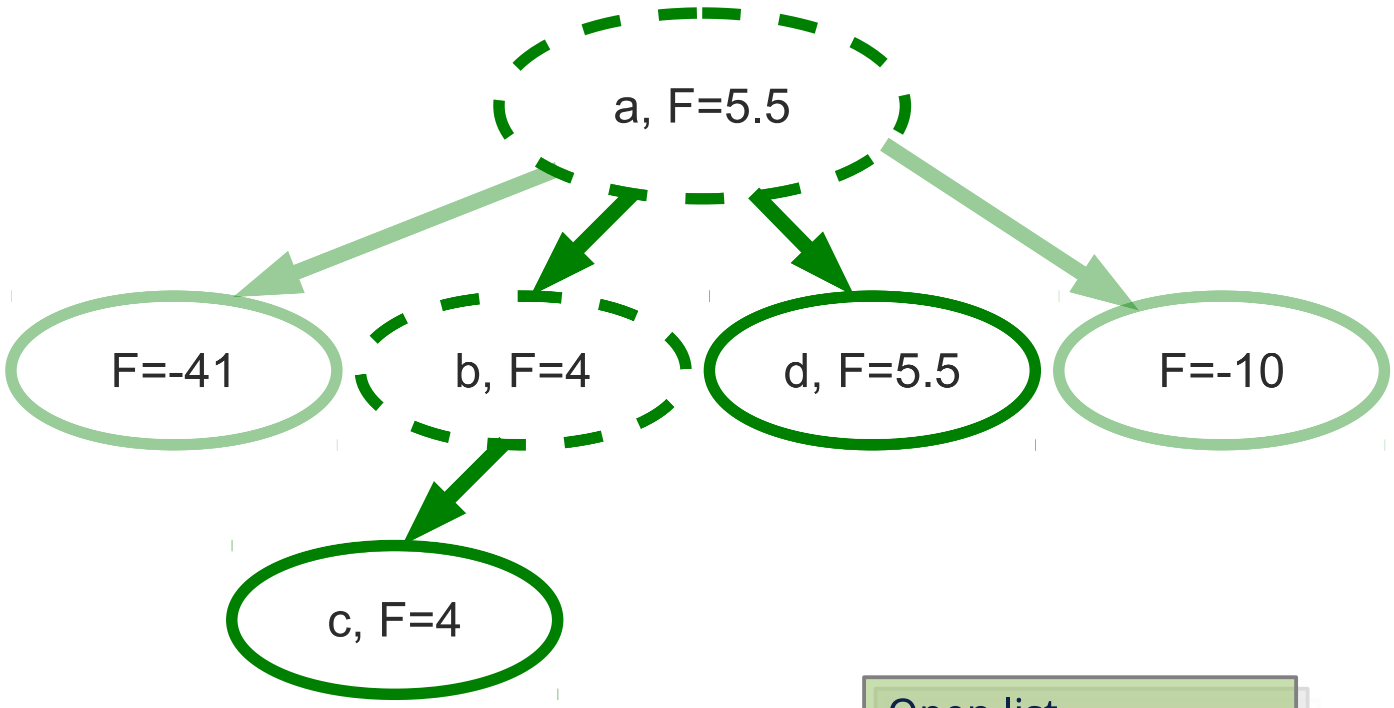
Open list
a – 6
c – 4
b – 4

Incremental Expansion



Open list
d – 5.5
a – 5.5
c – 4
b – 4

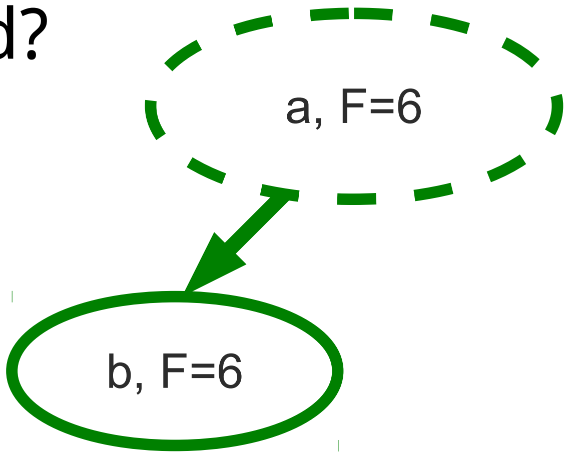
Incremental Expansion



Open list
d – 5.5
a – 5.5
c – 4
b – 4

Incremental Expansion: How?

- How do we generate the next-best child?



- Node ↔ BG, so...
  - find the solutions of the BG
    - in decreasing order of value
    - i.e., 'incremental BG solver'
  - Modification of BaGaBaB [Oliehoek et al. 2010]
    - stop searching when next solution found
    - save search tree for next time visited.
  - Nested A*!

Results

GMAA*-ICE can solve higher horizons than listed

incremental expansion complements incr. clustering

	problem primitives			
	n	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{A}_i\rvert$	$\lvert\mathcal{O}_i\rvert$
Dec-Tiger	2	2	3	2
BroadcastChannel	2	4	2	2
GridSmall	2	16	5	2
Cooperative Box Pushing	2	100	4	5
Recycling Robots	2	4	3	2
Hotel 1	2	16	3	4
FireFighting	2	432	3	2

'–' memory limit violations
'*' time limit overruns
'#' heuristic bottleneck

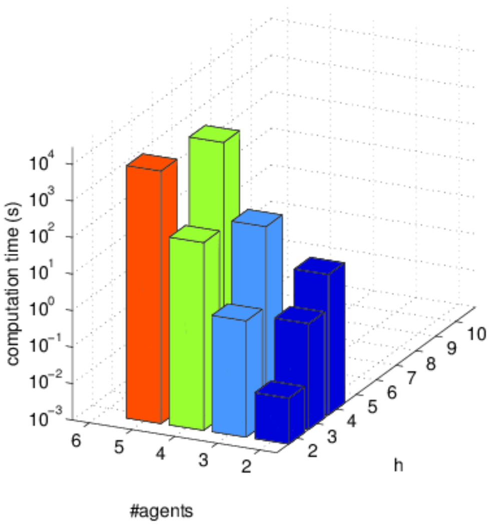
h	MILP	DP-LPC	DP-IPG	GMAA — Q_{BG} IC	GMAA — Q_{BG} ICE	GMAA — Q_{BG} heur
BroadcastChannel, ICE solvable to $h = 900$						
2	0.38	≤ 0.01	0.09	≤ 0.01	≤ 0.01	≤ 0.01
3	1.83	0.50	56.66	≤ 0.01	≤ 0.01	≤ 0.01
4	34.06	*	*	≤ 0.01	≤ 0.01	≤ 0.01
5	48.94			≤ 0.01	≤ 0.01	≤ 0.01
Dec-Tiger, ICE solvable to $h = 6$						
2	0.69	0.05	0.32	≤ 0.01	≤ 0.01	≤ 0.01
3	23.99	60.73	55.46	≤ 0.01	≤ 0.01	≤ 0.01
4	*	–	2286.38	0.27	≤ 0.01	0.03
5			–	21.03	0.02	0.09
FireFighting (2 agents, 3 houses, 3 firelevels), ICE solvable to $h \gg 1000$						
2	4.45	8.13	10.34	≤ 0.01	≤ 0.01	≤ 0.01
3	–	–	569.27	0.11	0.10	0.07
4			–	950.51	1.00	0.65
GridSmall, ICE solvable to $h = 6$						
2	6.64	11.58	0.18	0.01	≤ 0.01	≤ 0.01
3	*	–	4.09	0.10	≤ 0.01	0.42
4			77.44	1.77	≤ 0.01	67.39
Recycling Robots, ICE solvable to $h = 70$						
2	1.18	0.05	0.30	≤ 0.01	≤ 0.01	≤ 0.01
3	*	2.79	1.07	≤ 0.01	≤ 0.01	≤ 0.01
4		2136.16	42.02	≤ 0.01	≤ 0.01	0.02
5		–	1812.15	≤ 0.01	≤ 0.01	0.02
Hotel 1, ICE solvable to $h = 9$						
2	1.92	6.14	0.22	≤ 0.01	≤ 0.01	0.03
3	315.16	2913.42	0.54	≤ 0.01	≤ 0.01	1.51
4	–	–	0.73	≤ 0.01	≤ 0.01	3.74
5			1.11	≤ 0.01	≤ 0.01	4.54
9			8.43	0.02	≤ 0.01	20.26
10			17.40	#	#	
15			283.76			
Cooperative Box Pushing (Q_{POMDP}), ICE solvable to $h = 4$						
2	3.56	15.51	1.07	≤ 0.01	≤ 0.01	≤ 0.01
3	2534.08	–	6.43	0.91	0.02	0.15
4	–		1138.61	*	328.97	0.63

Results

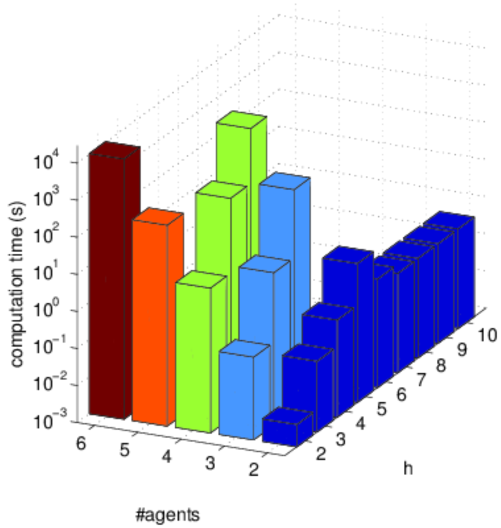
h	V^*	T_{GMAA*}(s)	T_{IC}(s)	T_{ICE}(s)
Recycling Robots				
3	10.660125	≤ 0.01	≤ 0.01	≤ 0.01
4	13.380000	713.41	≤ 0.01	≤ 0.01
5	16.486000	–	≤ 0.01	≤ 0.01
6	**19.554200**		≤ 0.01	≤ 0.01
10	**31.863889**		≤ 0.01	≤ 0.01
15	**47.248521**		≤ 0.01	≤ 0.01
20	**62.633136**		≤ 0.01	≤ 0.01
30	**93.402367**		0.08	0.05
40	**124.171598**		0.42	0.25
50	**154.940828**		2.02	1.27
70	**216.479290**		–	28.66
80			–	–
BroadcastChannel				
4	3.890000	≤ 0.01	≤ 0.01	≤ 0.01
5	4.790000	1.27	≤ 0.01	≤ 0.01
6	**5.690000**	–	≤ 0.01	≤ 0.01
7	**6.590000**		≤ 0.01	≤ 0.01
10	**9.290000**		≤ 0.01	≤ 0.01
25	**22.881523**		≤ 0.01	≤ 0.01
50	**45.501604**		≤ 0.01	≤ 0.01
100	**90.760423**		≤ 0.01	≤ 0.01
250	**226.500545**		0.06	0.07
500	**452.738119**		0.81	0.94
700	**633.724279**		0.52	0.63
800			–	–
900	**814.709393**		9.57	11.11
1000			–	–

Cases that compress well

* excluding heuristic



GMAA*



GMAA*-ICE

Sufficient Plan-Time Statistics [Oliehoek 2013]

- Optimal decision rule
  depends on past joint policy φ^t → search tree

- In fact possible to give an expression for the optimal value function based on φ^t [Oliehoek et al. 2008]

- Recent insight:
  reformulation based on a **sufficient statistic**
  - compact formulation of Q*
  - search tree → DAG ("suff. stat-based pruning")

Optimal Value Functions

2 parts:

- Value propagation:

- Value optimization:

Optimal Value Functions

2 parts:

- Value propagation:
  - last stage t=h-1

(past Pol, AOH, decis. rule)

expected reward

$$Q^*(\varphi^{h-1},\vec{\theta}^{h-1},\delta^{h-1}) = R(\vec{\theta}^{h-1},\delta^{h-1}(\vec{\theta}^{h-1}))$$

$$\delta^t(\vec{\theta}^t) = \langle \delta_1^t(\vec{\theta}_1^t),\ldots,\delta_n^t(\vec{\theta}_n^t)\rangle$$

- Value optimization:

Optimal Value Functions

2 parts:

- Value propagation:
  - last stage t=h-1 $Q^*(\varphi^{h-1}, \vec{\theta}^{h-1}, \delta^{h-1}) = R(\vec{\theta}^{h-1}, \delta^{h-1}(\vec{\theta}^{h-1}))$
  - t<h-1

$$Q^*(\varphi^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\varphi^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

$$\varphi^{t+1} = (\varphi^t, \delta^t)$$

- Value optimization:

Optimal Value Functions

2 parts:

- Value propagation:
  - last stage t=h-1 $Q^*(\varphi^{h-1},\vec{\theta}^{h-1},\delta^{h-1})=R(\vec{\theta}^{h-1},\delta^{h-1}(\vec{\theta}^{h-1}))$
  - t<h-1

$$Q^*(\varphi^t,\vec{\theta}^t,\delta^t)=R(\vec{\theta}^t,\delta^t(\vec{\theta}^t))+\sum_o P(o|\vec{\theta}^t,\delta^t(\vec{\theta}^t))Q^*(\varphi^{t+1},\vec{\theta}^{t+1},\delta^{*t+1})$$

$$\varphi^{t+1}=(\varphi^t,\delta^t)$$

- Value optimization:

$$\delta^{*t+1}=arg\,max_{\delta^{t+1}}\sum_{\vec{\theta}^{t+1}} P(\vec{\theta}^{t+1}|b^0,\varphi^{t+1})Q^*(\varphi^{t+1},\vec{\theta}^{t+1},\delta^{t+1})$$

(need to do 'stage-wise' maximization)

Optimal Value Functions

> **Q vs V ?**
>
> → we can interpret it as a 'plan-time' MDP
>
> - ►state: φ
> - ►actions: δ
>
> $$V(\varphi^t) = max_{\delta^t} \; Q^*(\varphi^t, \delta^t)$$
>
> $$Q^*(\varphi^t, \delta^t) = \sum_{\vec{\theta}^t} P(\vec{\theta}^t | b^0, \varphi^t) Q^*(\varphi^t, \vec{\theta}^t, \delta^t)$$

2 parts:

- Value propagation:
  - last stage t=h-1
  - t<h-1

$$Q^*(\varphi^{h-1}, \vec{\theta}^{h-1}, \delta^{h-1}) = R(\vec{\theta}^{h-1}, \delta^{h-1}(\vec{\theta}^{h-1}))$$

$$Q^*(\varphi^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\varphi^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

$$\varphi^{t+1} = (\varphi^t, \delta^t)$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} P(\vec{\theta}^{t+1} | b^0, \varphi^{t+1}) Q^*(\varphi^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

(need to do 'stage-wise' maximization)

Optimal Value Functions

2 parts:

- Value propagation:
  - last stage t=h-1 $Q^*(\varphi^{h-1}, \vec{\theta}^{h-1}, \delta^{h-1}) = R(\vec{\theta}^{h-1}, \delta^{h-1}(\vec{\theta}^{h-1}))$
  - t<h-1

$$Q^*(\varphi^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\varphi^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

$$\varphi^{t+1} = (\varphi^t, \delta^t)$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} P(\vec{\theta}^{t+1}|b^0, \varphi^{t+1}) Q^*(\varphi^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

(need to do 'stage-wise' maximization)

Optimal Value Functions

2 parts:

- Value propagation:
  - last stage t=h-1 $\quad Q^*(\varphi^{h-1},\vec{\theta}^{h-1},\delta^{h-1})=R(\vec{\theta}^{h-1},\delta^{h-1}(\vec{\theta}^{h-1}))$
  - t<h-1

$$Q^*(\varphi^t,\vec{\theta}^t,\delta^t)=R(\vec{\theta}^t,\delta^t(\vec{\theta}^t))+\sum_o P(o|\vec{\theta}^t,\delta^t(\vec{\theta}^t))Q^*(\varphi^{t+1},\vec{\theta}^{t+1},\delta^{*t+1})$$

$$\varphi^{t+1}=(\varphi^t,\delta^t)$$

- Value optimization:

$$\delta^{*t+1}=\arg\max_{\delta^{t+1}}\sum_{\vec{\theta}^{t+1}} P(\vec{\theta}^{t+1}|b^0,\varphi^{t+1})Q^*(\varphi^{t+1},\vec{\theta}^{t+1},\delta^{t+1})$$

(need to do 'stage-wise' maximization)

Optimal Value Functions

2 parts:

- Value propagation:
  - last stage t=h-1 $Q^*(\varphi^{h-1},\vec{\theta}^{h-1},\delta^{h-1}) = R(\vec{\theta}^{h-1},\delta^{h-1}(\vec{\theta}^{h-1}))$
  - t<h-1

$$Q^*(\varphi^t,\vec{\theta}^t,\delta^t) = R(\vec{\theta}^t,\delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t,\delta^t(\vec{\theta}^t))\, Q^*(\varphi^{t+1},\vec{\theta}^{t+1},\delta^{*t+1})$$

$$\varphi^{t+1} = (\varphi^t,\delta^t)$$

- Value optimization:

$$\delta^{*t+1} = \arg\max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} P(\vec{\theta}^{t+1}|b^0,\varphi^{t+1})\, Q^*(\varphi^{t+1},\vec{\theta}^{t+1},\delta^{t+1})$$

(need to do 'stage-wise' maximization)

Optimal Value Functions

2 parts:

- Value propag
  - last stage t
  - t<h-1

$, \delta^{h-1}(\vec{\theta}^{h-1}))$

But: initial dependence only through this probability term!

$Q^*(\varphi^t, \vec{\theta}^t, \delta^t) = R$ … $, \vec{\theta}^{t+1}, \delta^{*t+1})$

$\varphi^{t+1} = (\varphi^t, \delta^t)$

- Value optimization:

$$\delta^{*t+1} = \arg\max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} P(\vec{\theta}^{t+1} | b^0, \varphi^{t+1}) Q^*(\varphi^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

(need to do 'stage-wise' maximization)

Sufficient Statistic – 1

2 parts:

- Value propagation:

$$Q^*(\sigma^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} \sigma^{t+1}(\vec{\theta}^{t+1}) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

Sufficient Statistic – 1

2 parts:

- Value propagation:

$$Q^*(\sigma^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} \sigma^{t+1}(\vec{\theta}^{t+1}) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

Limited use: every **deterministic** past joint policy induces a different σ !

Sufficient Statistic – 2

use: $\sigma^t(s, \vec{o}^t)$

2 parts:

- Value propagation:

$$Q^*(\sigma^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} \sigma^{t+1}(\vec{\theta}^{t+1}) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

Sufficient Statistic – 2

use: $\sigma^t(s, \vec{o}^t)$

2 parts:

- Value propagation:

$$Q^*(\sigma^t, \vec{\theta}^t, \delta^t) = R(\vec{\theta}^t, \delta^t(\vec{\theta}^t)) + \sum_o P(o|\vec{\theta}^t, \delta^t(\vec{\theta}^t)) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{*t+1})$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{\theta}^{t+1}} \sigma^{t+1}(\vec{\theta}^{t+1}) Q^*(\sigma^{t+1}, \vec{\theta}^{t+1}, \delta^{t+1})$$

▶substitute AOH → OH
▶but then → also adapt R(..) and P(o|...)

Sufficient Statistic – 2

use: $\sigma^t(s, \vec{o}^t)$

2 parts:

- Value propagation:

$$Q^*(\sigma^t, \vec{o}^t, \delta^t) = R(\sigma^t, \vec{o}^t, \delta^t) + \sum_o P(o|\sigma^t, \vec{o}^t, \delta^t) Q^*(\sigma^{t+1}, \vec{o}^{t+1}, \delta^{*t+1})$$

- Value optimization:

$$\delta^{*t+1} = arg\,max_{\delta^{t+1}} \sum_{\vec{o}^{t+1}} \sigma^t(\vec{o}^{t+1}) Q^*(\sigma^{t+1}, \vec{o}^{t+1}, \delta^{t+1})$$

Results –1

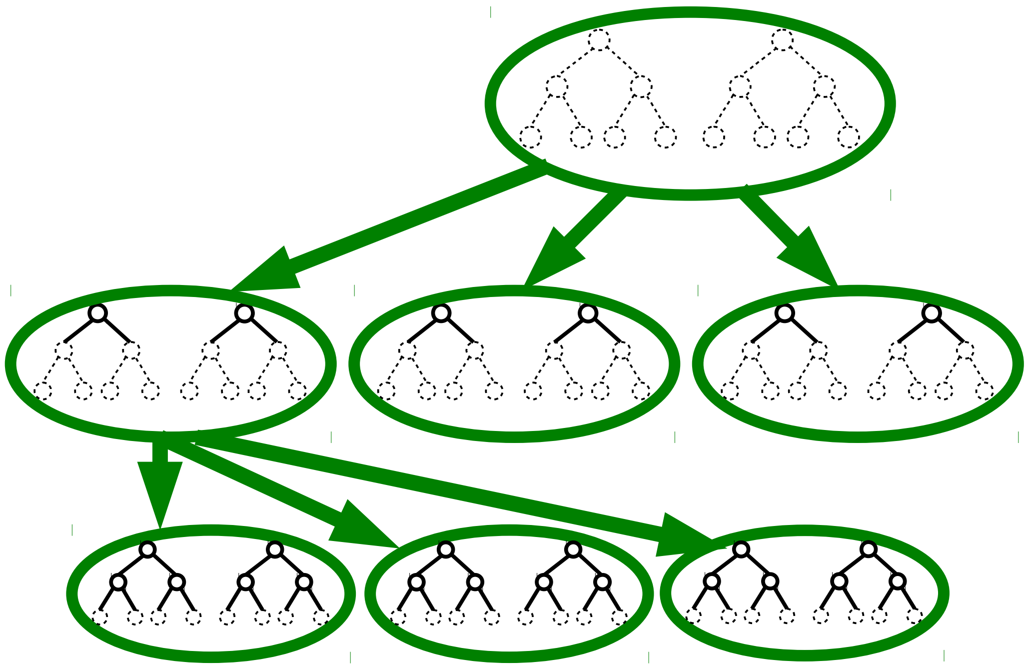
- Reduction in size of Q*

	$t=1$		$t=2$		$t=3$	
	φ_1	σ_1	φ_2	σ_2	φ_3	σ_3
tiger	9	2	729	20	$4.78e6$	4520
broadcast	4	4	64	56	$1.63e4$	$1.16e4$
recycling	9	9	729	441	$4.78e6$	X
FF	9	9	729	729	$4.78e6$	X
gridsmall	25	16	$1.56e4$	4096	$6.10e9$	X
hotel1	9	1	$5.90e4$	4	$1.7e19$	–

Table 1: Number of σ_t vs. number of φ_t .

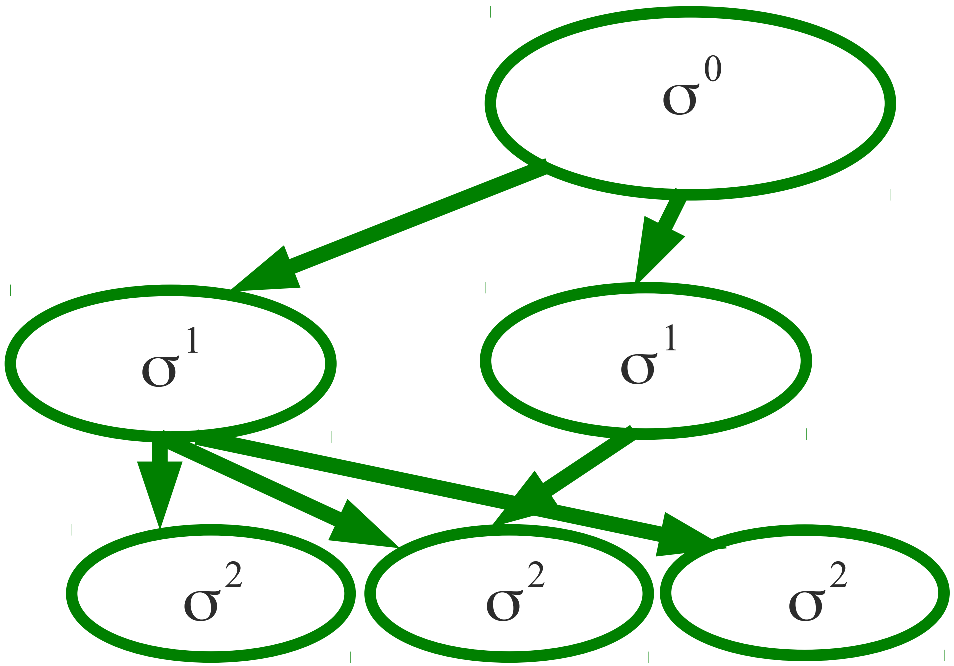
Sufficient statistic-based pruning

- Before



Sufficient statistic-based pruning

- Now
  - many φ ↔ same σ



- GMAA*-ICE with SSBP:
  - perform GMAA*-ICE, but at each node compute σ
  - if same σ but lower G-value → prune branch

Results – 2

- Speed-up GMAA*-ICE due to SSBP

	SSBP	nodes created at depth t 1	2	3	4	5	6
tiger							
QMDP, h5	yes	1	10	615	28475	4	
	no	9	69	2319	41130	4	
QBG,h6	yes	1	2	8	18	162	1
	no	9	2	8	18	166	1
hotel1							
QMDP, h4	yes	1	4	6	3		
	no	9	252	11178	10935		
QMDP, h5	yes	1	4	12	15	7	
	no	not solvable (out of 2GB mem.)					
QBG, h5	no	9	4	3	3	1	

Table 2: Number of created child nodes in GMAA-ICE, when using sufficient statistic-based pruning (SSBP).

promising, but does not address the current bottleneck...

References

- Most references can be found in

  Frans A. Oliehoek. **Decentralized POMDPs**. In Wiering, Marco and van Otterlo, Martijn, editors, *Reinforcement Learning: State of the Art*, Adaptation, Learning, and Optimization, pp. 471–503, Springer Berlin Heidelberg, Berlin, Germany, 2012.

- Other:
  - Dibangoye, Amato, Buffet, & Charpillet. Optimally Solving Dec-POMDPs as Continuous-State MDPs. *IJCAI*, 2013.
  - Oliehoek, Spaan, Amato, & Whiteson. Incremental Clustering and Expansion for Faster Optimal Planning in Decentralized POMDPs. *JAIR,* 2013.
  - Oliehoek. Sufficient Plan-Time Statistics for Decentralized POMDPs. *IJCAI*, 2013.