Artificial Intelligence Techniques
CS4375
Lecture 8: Planning under sensing uncertainty

Matthijs Spaan

Delft University of Technology

Delft, The Netherlands

September 28, 2023

Outline for today

1. Partially Observable Markov Decision Processes
   - Model
   - Beliefs
   - Algorithms
2. Online planning
   - MDP
   - POMDP

Partially Observable Markov Decision Processes

Beyond MDPs

- Real agents cannot directly observe the state.
- Sensors provide partial and noisy information about the world.

Beyond MDPs

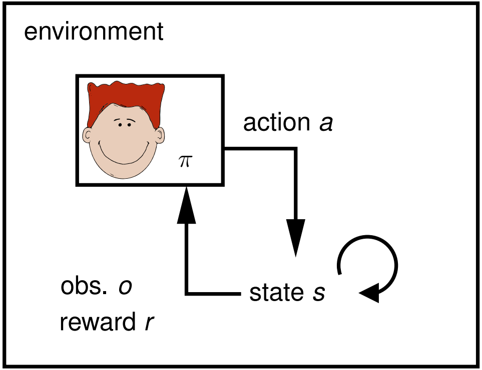
- MDPs have been very successful, but they require an observable Markovian state.
- Many domains this is impossible (or expensive) to obtain:
  - Diagnosis (medical, maintenance)
  - Robot navigation
  - Tutoring
  - Dialog systems
  - Vision-based robotics
  - Fault recovery

Model

Observation model

- Imperfect sensors.
- Partially observable environment:
  - Sensors are **noisy**.
  - Sensors have a **limited view**.
- $p(o|s', a)$ is the probability the agent receives observation o in state s' after taking action a.

POMDP Agent



POMDPs

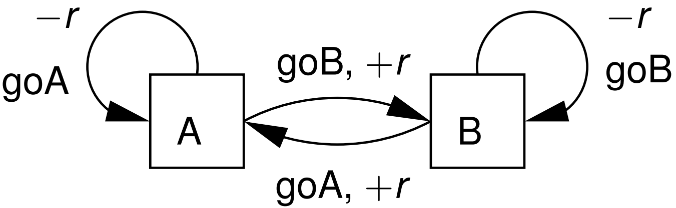
Partially observable Markov decision processes (POMDPs) (Kaelbling et al., 1998):

- Framework for agent planning under uncertainty.
- Typically assumes discrete sets of states S, actions A and observations O.
- Transition model $p(s'|s, a)$: models the effect of **actions**.
- Observation model $p(o|s', a)$: relates **observations** to states.
- Task is defined by a **reward** model $R(s, a)$.
- A planning horizon h (finite or ∞).
- A discount rate $0 \leq \gamma < 1$.
- Goal is to compute plan, or **policy** π, that maximizes long-term reward.

Beliefs

Memory

- In POMDPs memory is required for optimal decision making.
- In this non-observable example (Singh et al., 1994):



Policy	Value
MDP: optimal policy	$V = \sum_{t=0}^{\infty} \gamma^t r = \frac{r}{1-\gamma}$
POMDP: memoryless deterministic	$V_{\max} = r - \frac{\gamma r}{1-\gamma}$
POMDP: memoryless stochastic	$V = 0$
POMDP: memory-based (optimal)	$V_{\min} = \frac{\gamma r}{1-\gamma} - r$

Beliefs

Beliefs:

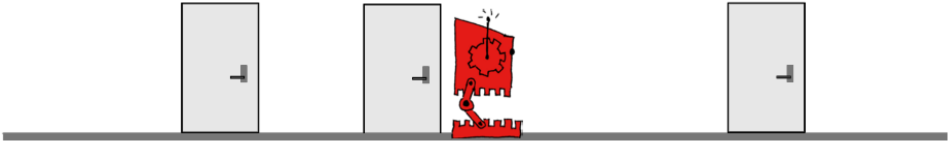
- The agent maintains a **belief** $b(s)$ of being at state s.
- After action $a \in A$ and observation $o \in O$ the belief $b(s)$ can be updated using Bayes' rule:

$$b'(s') = \frac{p(o|s', a) \sum_s p(s'|s, a) b(s)}{p(o|b, a)}$$

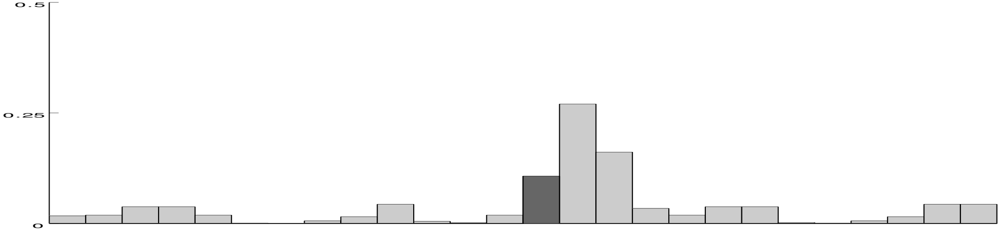
- The belief vector is a **Markov** signal for the planning task.

Belief update example

True situation:



Robot's belief:

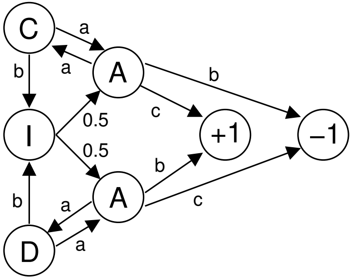


- Observations: *door* or **corridor**, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

Algorithms

MDP-based algorithms

- Exploit belief state, and use the MDP solution as a heuristic.
- Most likely state (Cassandra et al., 1996): $\pi_{MLS}(b) = \pi^*(\arg\max_s b(s))$.
- Q_{MDP} (Littman et al., 1995): $\pi_{Q_{\text{MDP}}}(b) = \arg\max_a \sum_s b(s) Q^*(s, a)$.



(Parr and Russell, 1995)

POMDPs as continuous-state MDPs

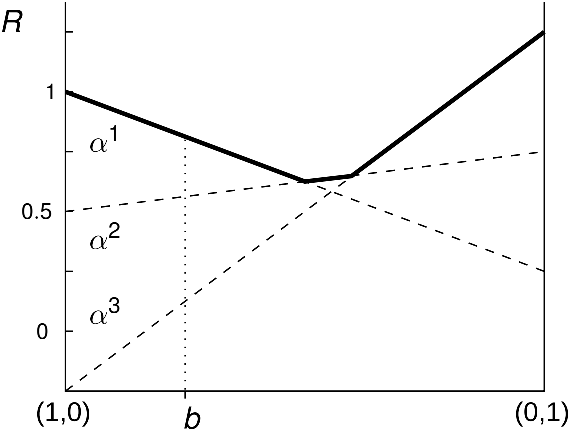
A POMDP can be treated as a continuous-state (belief-state) MDP:

- Continuous state space Δ: a simplex in $[0, 1]^{|S|-1}$.
- Stochastic Markovian transition model $p(b_a^o|b, a) = p(o|b, a)$. This is the normalizer of Bayes' rule.
- Reward function $R(b, a) = \sum_s R(s, a)b(s)$. This is the average reward with respect to $b(s)$.
- The agent fully 'observes' the new belief-state b_a^o after executing a and observing o.

Solving POMDPs

- A solution to a POMDP is a **policy**, i.e., a mapping $\pi : \Delta \mapsto A$ from beliefs to actions.
- The optimal value V^* of a POMDP satisfies the Bellman optimality equation $V^* = HV^*$:
$$V^*(b) = \max_a \left[R(b, a) + \gamma \sum_o p(o|b, a) V^*(b_a^o) \right]$$
- Value iteration repeatedly applies $V_{n+1} = HV_n$ starting from an initial V_0.
- Computing the optimal value function is a hard problem (PSPACE-complete for finite horizon, undecidable for infinite horizon).

Example V_0



$R(s, a)$	a_1	a_2	a_3
s_1	1.00	0.50	−0.25
s_2	0.25	0.75	1.25

PWLC shape of V_n

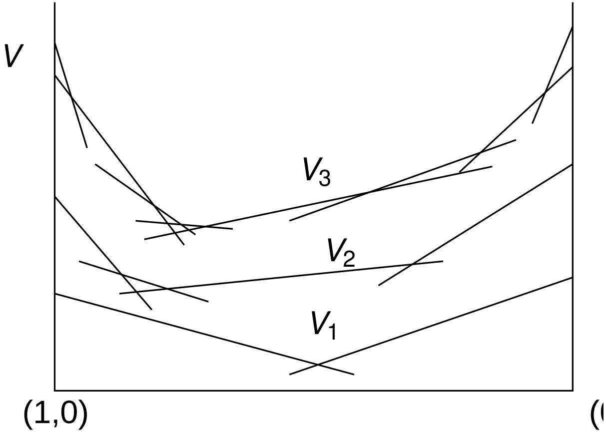
- Like V_0, V_n is as well piecewise linear and convex.
- Rewards $R(b, a) = b \cdot R(s, a)$ are linear functions of b. Note that the value of a point b satisfies:

$$V_{n+1}(b) = \max_a \left[b \cdot R(s, a) + \gamma \sum_o p(o|b, a) V_n(b_a^o) \right]$$

  which involves a maximization over (at least) the vectors $R(s, a)$.
- Intuitively: less uncertainty about the state (low-entropy beliefs) means better decisions (thus higher value).

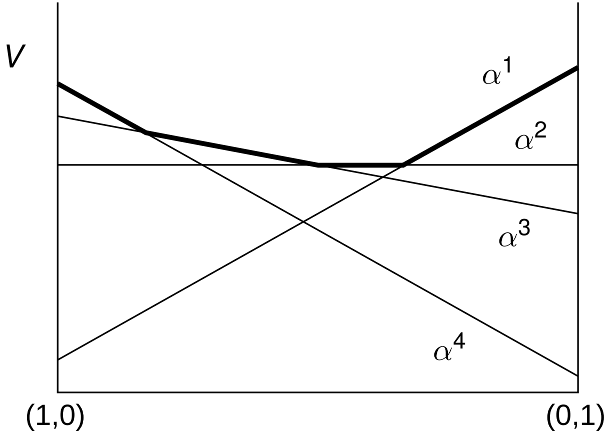
Exact value iteration

Value iteration computes a sequence of value function estimates $V_1, V_2, \ldots, V_n$, using the POMDP backup operator H, $V_{n+1} = HV_n$.

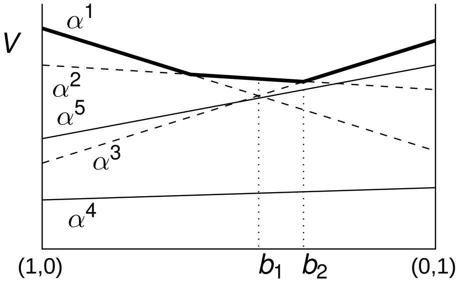


Optimal value functions

The optimal value function of a (finite-horizon) POMDP is piecewise linear and convex:
$V(b) = \max_{\alpha} b \cdot \alpha$.



Vector pruning



Linear program for pruning:

variables: $\forall s \in S, b(s); x$
maximize: x
subject to:

$$b \cdot (\alpha - \alpha') \geq x, \forall \alpha' \in V, \alpha' \neq \alpha$$
$$b \in \Delta(S)$$

Optimal POMDP methods

Enumerate and prune:

- Most straightforward: Monahan (1982)'s enumeration algorithm. Generates a maximum of $|A||V_n|^{|O|}$ vectors at each iteration, hence requires pruning.
- Incremental pruning (Zhang and Liu, 1996; Cassandra et al., 1997; Walraven and Spaan, 2017).

Search for witness points:

- One Pass (Sondik, 1971; Smallwood and Sondik, 1973).
- Relaxed Region, Linear Support (Cheng, 1988).
- Witness (Cassandra et al., 1994).

Sub-optimal techniques

- Grid-based approximations

  (Drake, 1962; Lovejoy, 1991; Brafman, 1997; Zhou and Hansen, 2001; Bonet, 2002).
- Optimizing finite-state controllers

  (Platzman, 1981; Hansen, 1998b; Poupart and Boutilier, 2004).
- Heuristic search in the belief tree

  (Satia and Lave, 1973; Hansen, 1998a).
- Compression or clustering

  (Roy et al., 2005; Poupart and Boutilier, 2003; Virin et al., 2007).
- Point-based techniques

  (Pineau et al., 2003; Smith and Simmons, 2004; Spaan and Vlassis, 2005; Shani et al., 2007; Kurniawati et al., 2008).
- Monte Carlo tree search

  (Silver and Veness, 2010).

Point-based backup

- For finite horizon V^* is piecewise linear and convex, and for infinite horizons V^* can be approximated arbitrary well by a PWLC value function (Smallwood and Sondik, 1973).
- Given value function V_n and a particular belief point b we can easily compute the vector α_{n+1}^b of HV_n such that

$$\alpha_{n+1}^b = \underset{\{\alpha_{n+1}^k\}_k}{\arg\max}\, b \cdot \alpha_{n+1}^k,$$

where $\{\alpha_{n+1}^k\}_{k=1}^{|HV_n|}$ is the (unknown) set of vectors for HV_n. We will denote this operation $\alpha_{n+1}^b = \mathtt{backup}(b)$.

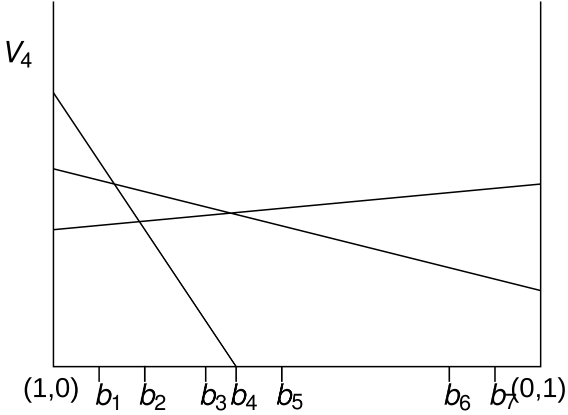
Point-based (approximate) methods

Point-based (approximate) value iteration plans only on a limited set of **reachable** belief points:

1. Let the robot explore the environment.
2. Collect a set B of belief points.
3. Run approximate value iteration on B.

PERSEUS: randomized point-based VI

Idea: at every backup stage improve the value of all $b \in B$.



(Spaan and Vlassis, 2005)

POMDPs in action

- Intention-aware online POMDP planning (Bai et al., 2015)
- ACAS X: Airborne Collision Avoidance System X (Kochenderfer et al., 2012)

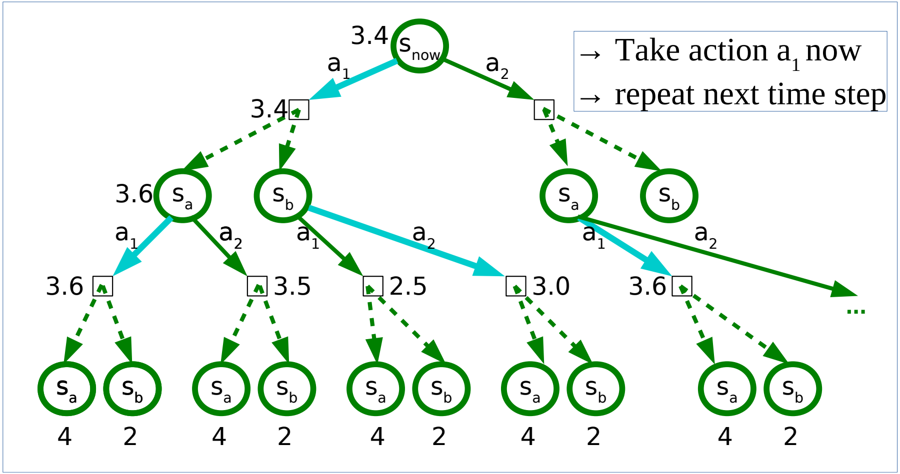
Online planning

Beyond off-line planning

- What if we interleave planning and execution?
- Basic understanding of a powerful technique: Monte Carlo Tree Search (MCTS)
  - critical component in AlphaGo
- Two variations:
  - MDPs: UCT (Kocsis and Szepesvári, 2006)
  - POMDPs: POMCP (Silver and Veness, 2010)

Dynamic Programming in trees

- Construct a plan for h time steps into the future



Beyond Dynamic Programming

Dynamic Programming

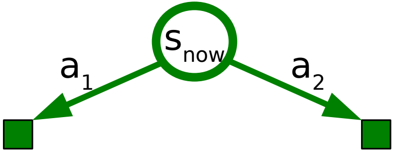
- Problem: trees get huge
  - Not practical

Monte Carlo Tree Search (MCTS)

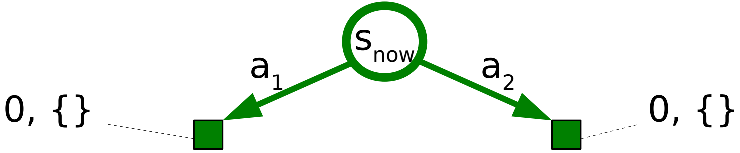
- Provides leverage by
  - incrementally constructing a sampled version of the tree
  - focusing on promising regions
- Monte Carlo updates instead of Bellman updates

MDP

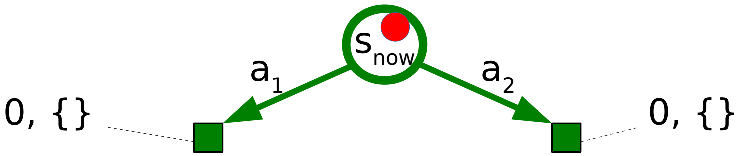
Monte Carlo Tree Search – MDP Example



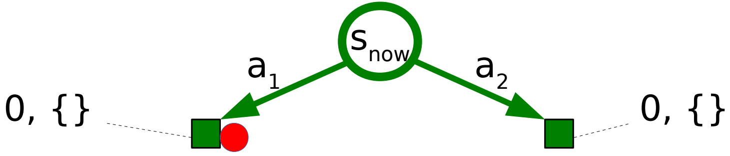
Monte Carlo Tree Search – MDP Example



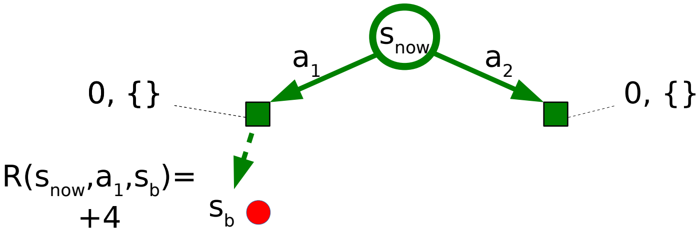
Monte Carlo Tree Search – MDP Example



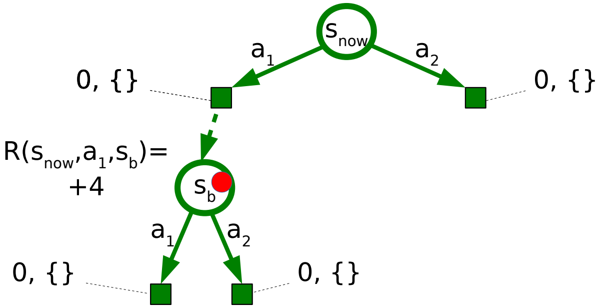
Monte Carlo Tree Search – MDP Example



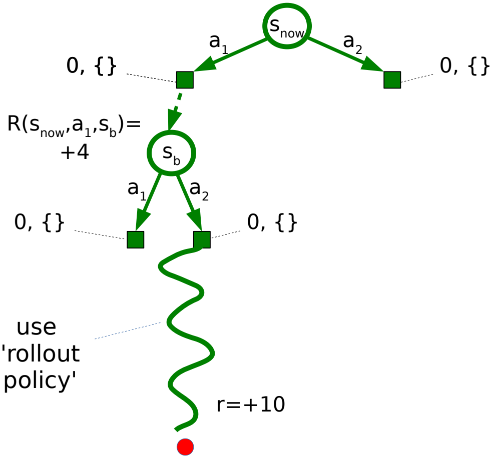
Monte Carlo Tree Search – MDP Example



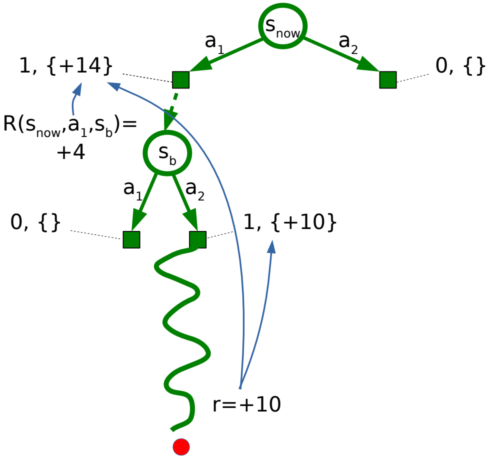
Monte Carlo Tree Search – MDP Example



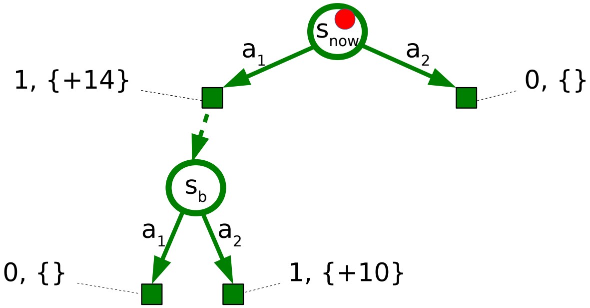
Monte Carlo Tree Search – MDP Example



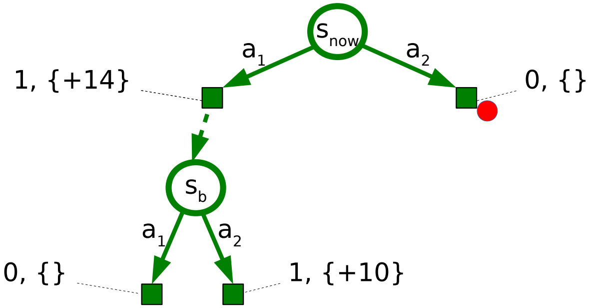
Monte Carlo Tree Search – MDP Example



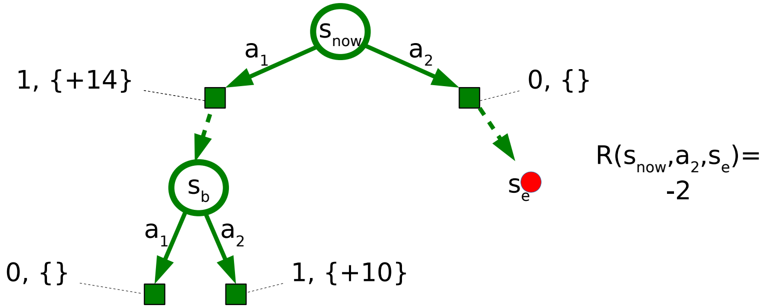
Monte Carlo Tree Search – MDP Example



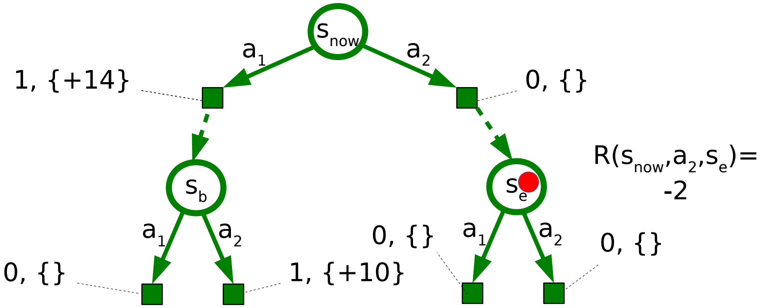
Monte Carlo Tree Search – MDP Example



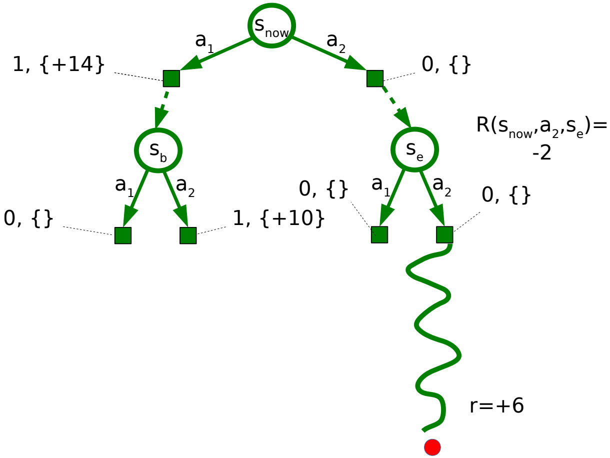
Monte Carlo Tree Search – MDP Example



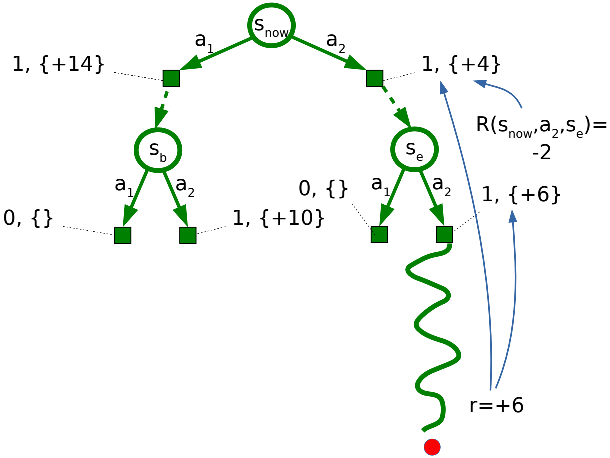
Monte Carlo Tree Search – MDP Example



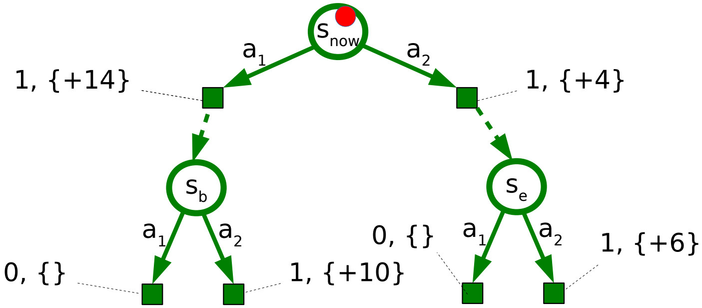
Monte Carlo Tree Search – MDP Example



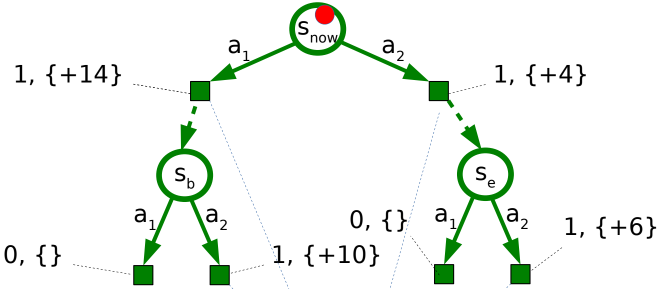
Monte Carlo Tree Search – MDP Example



Monte Carlo Tree Search – MDP Example

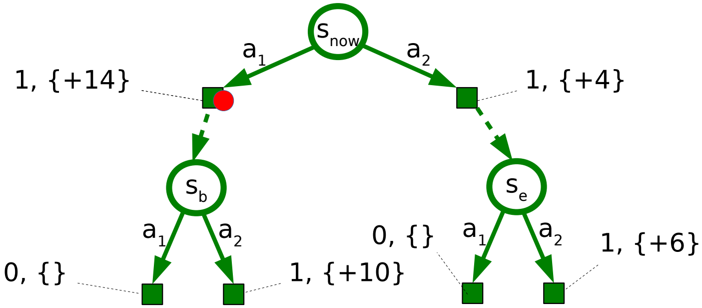


Monte Carlo Tree Search – MDP Example

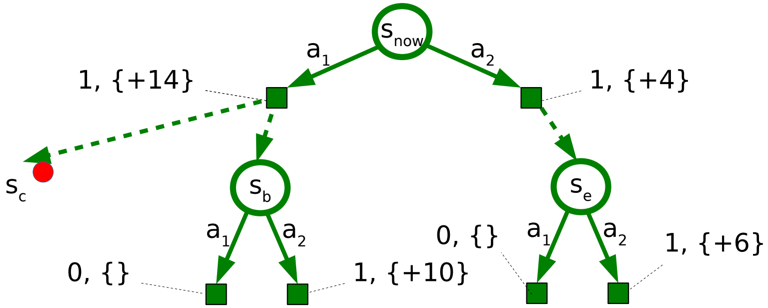


NOTE: the statistics maintained, represent an estimate of **Q(s,a)**

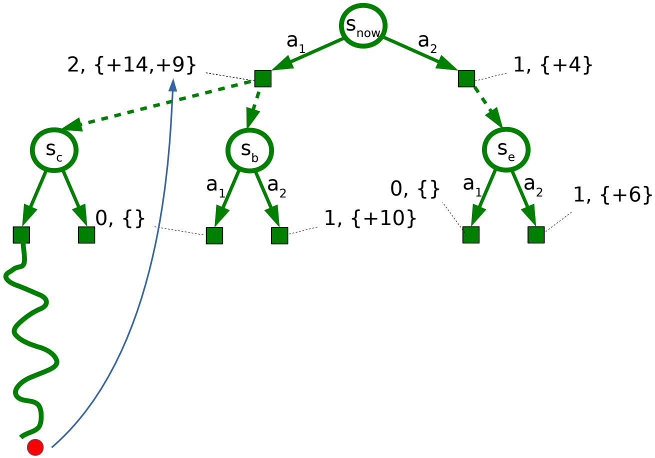
Monte Carlo Tree Search – MDP Example



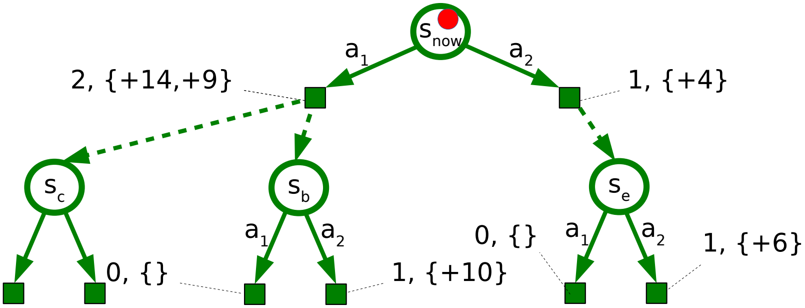
Monte Carlo Tree Search – MDP Example



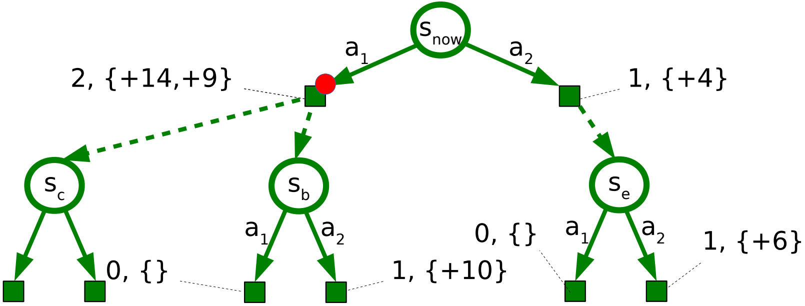
Monte Carlo Tree Search – MDP Example



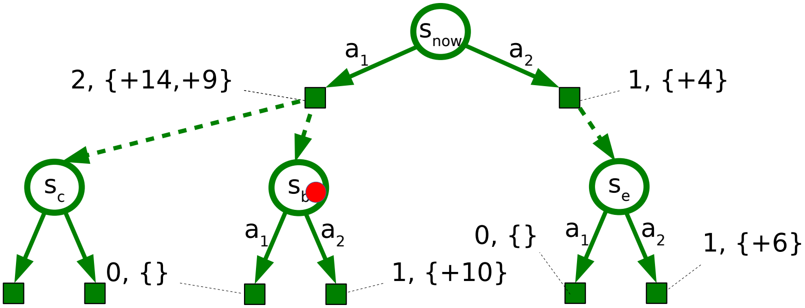
Monte Carlo Tree Search – MDP Example



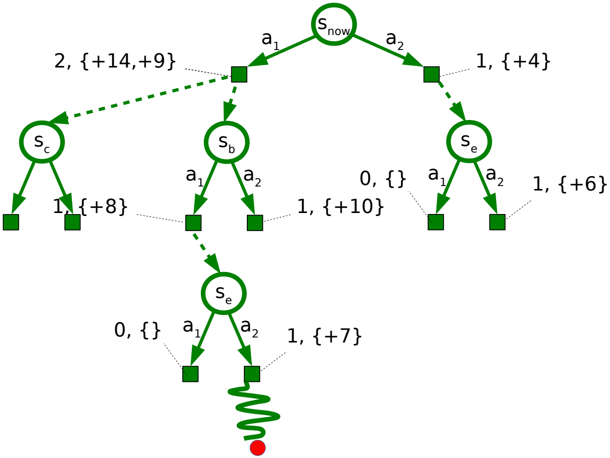
Monte Carlo Tree Search – MDP Example



Monte Carlo Tree Search – MDP Example

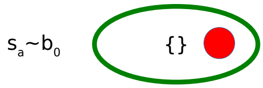


Monte Carlo Tree Search – MDP Example



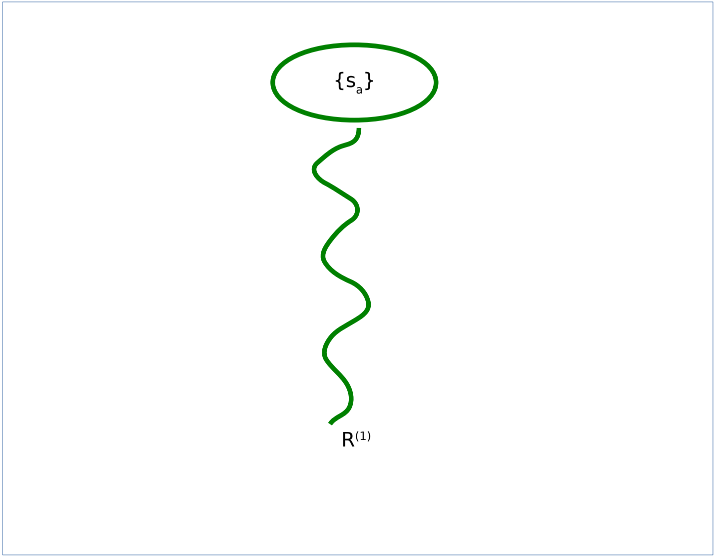
POMDP

Monte Carlo Tree Search – POMDP Example

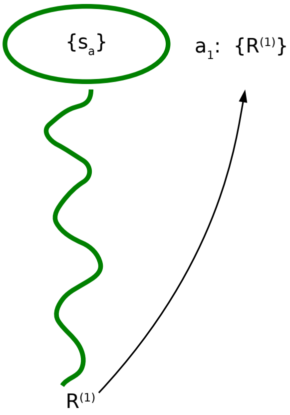


Monte Carlo Tree Search – POMDP Example

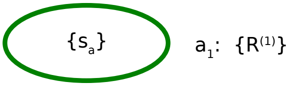
Monte Carlo Tree Search – POMDP Example



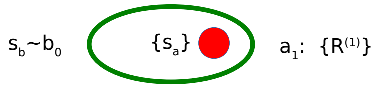
Monte Carlo Tree Search – POMDP Example



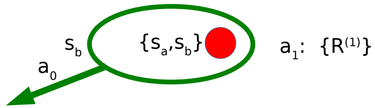
Monte Carlo Tree Search – POMDP Example



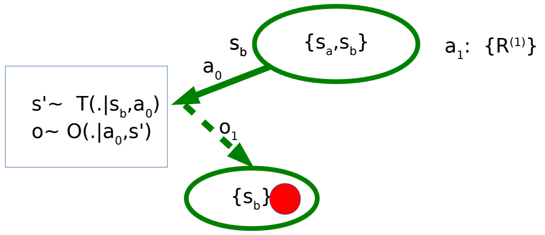
Monte Carlo Tree Search – POMDP Example



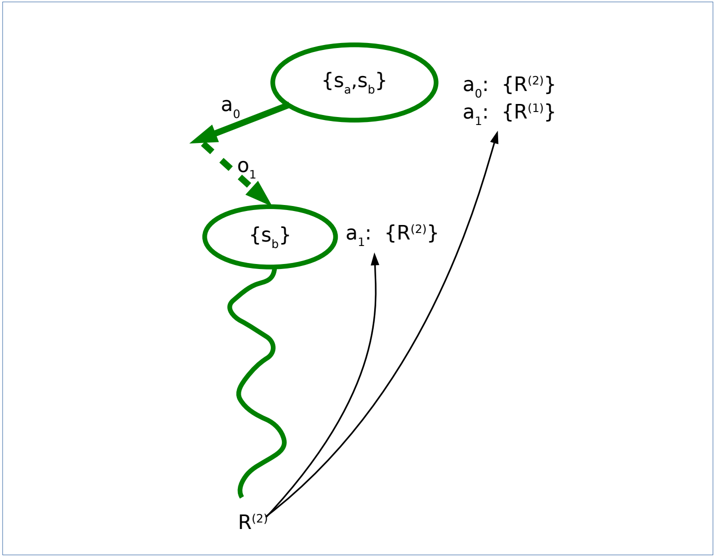
Monte Carlo Tree Search – POMDP Example



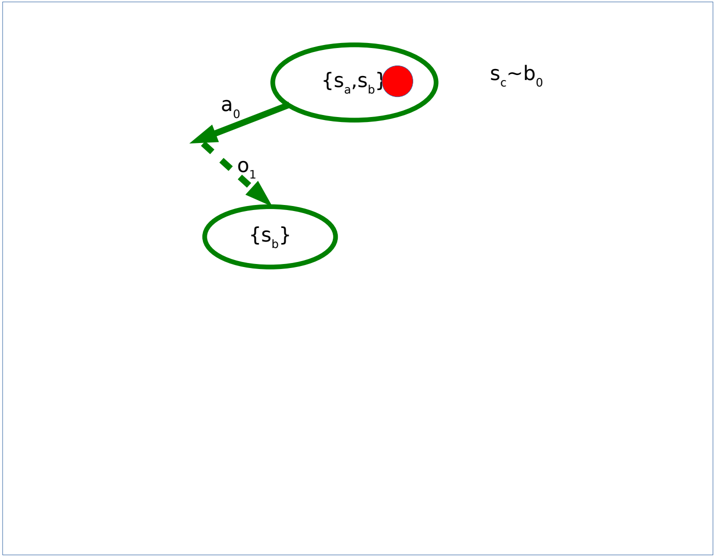
Monte Carlo Tree Search – POMDP Example



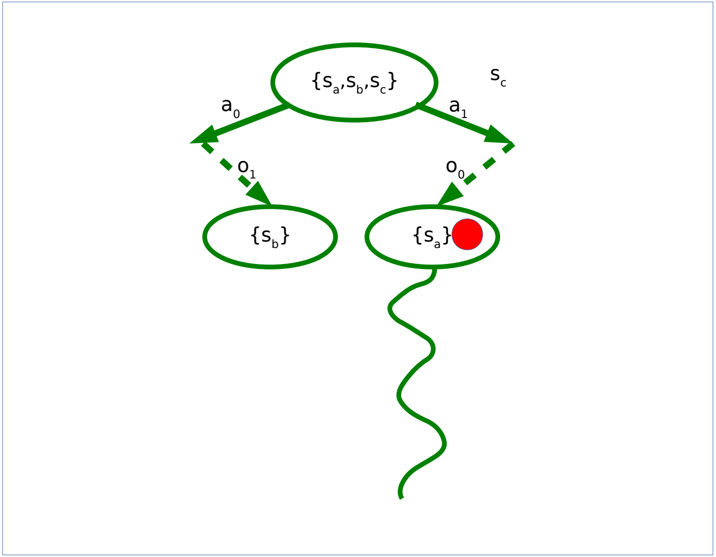
Monte Carlo Tree Search – POMDP Example



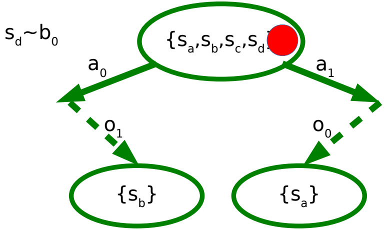
Monte Carlo Tree Search – POMDP Example



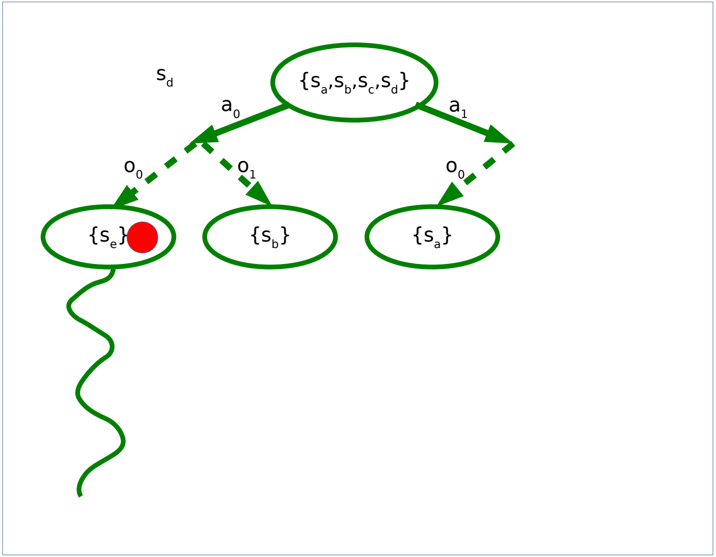
Monte Carlo Tree Search – POMDP Example



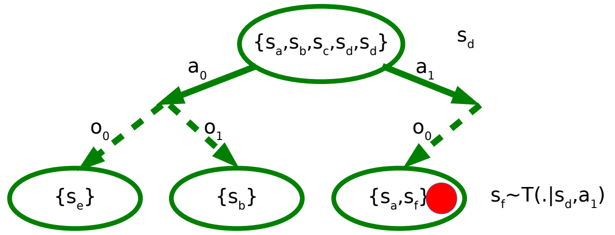
Monte Carlo Tree Search – POMDP Example



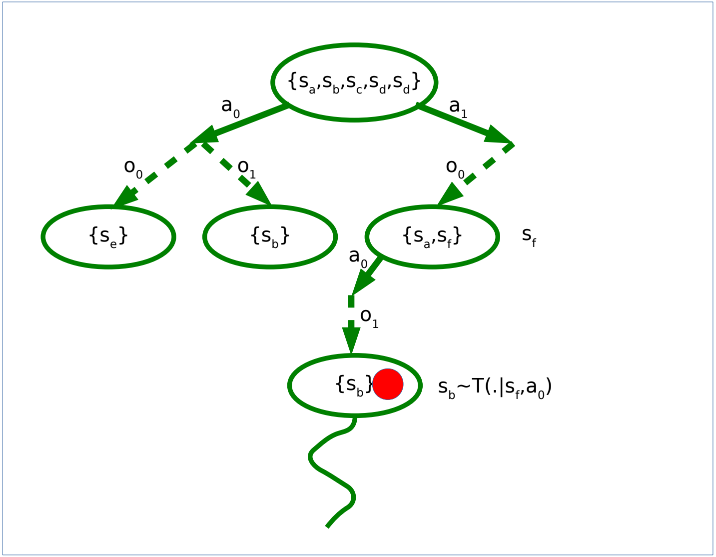
Monte Carlo Tree Search – POMDP Example



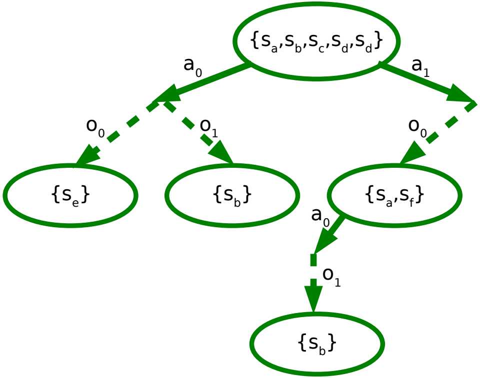
Monte Carlo Tree Search – POMDP Example



Monte Carlo Tree Search – POMDP Example



Monte Carlo Tree Search – POMDP Example



MCTS – Convergence

- Does this converge?
- Yes, but not trivial, conflicting requirements
  - accurate value estimates $\rightarrow$ try all actions infinitely often
  - estimates of an optimal policy $\rightarrow$ be greedy in sub-tree

MCTS – Action selection in the tree

- What actions to select?
- Balance:
  - exploitation: focus on good branches
  - exploration: see if there could be better branches
- Typical approach: exploration bonus
  - For instance, UCT algorithm (Kocsis and Szepesvári, 2006)

$$U(h, a) = Q(h, a) + c\sqrt{\frac{\log(N_h + 1)}{N_a}}$$

  - Upper confidence bound = mean return + exploration bonus

MCTS – Rollout policies

- Another important component: what rollout policy?
- In theory:
  - as long as it gives positive probability to any action
- In practice:
  - huge effect!
  - use domain knowledge
- Perspective: MCTS as a “policy improvement operator”
  - you give it a policy, and MCTS makes it better by applying additional search

MCTS – Pros/cons

Benefits:

- rapidly zooms in on promising regions
- can be used to improve policies
- basis of many successful applications

Limitations:

- needle in the hay-stack problems
- problems with high branching factor

References

H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee. Intention-aware online pomdp planning for autonomous driving in a crowd. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2015.

B. Bonet. An epsilon-optimal grid-based algorithm for partially observable Markov decision processes. In *International Conference on Machine Learning*, 2002.

R. I. Brafman. A heuristic variable grid solution method for POMDPs. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997.

A. R. Cassandra, L. P. Kaelbling, and M. L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1994.

A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile robot navigation. In *Proc. of International Conference on Intelligent Robots and Systems*, 1996.

A. R. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proc. of Uncertainty in Artificial Intelligence*, 1997.

H. T. Cheng. *Algorithms for partially observable Markov decision processes*. PhD thesis, University of British Columbia, 1988.

A. W. Drake. *Observation of a Markov process through a noisy channel*. Sc.D. thesis, Massachusetts Institute of Technology, 1962.

E. A. Hansen. *Finite-memory control of partially observable systems*. PhD thesis, University of Massachusetts, Amherst, 1998a.

E. A. Hansen. Solving POMDPs by searching in policy space. In *Proc. of Uncertainty in Artificial Intelligence*, 1998b.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos. Next-generation airborne collision avoidance system. *Lincoln Laboratory Journal*, 19(1), 2012.

L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*, pages 282–293. Springer, 2006.

H. Kurniawati, D. Hsu, and W. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, 2008.

M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning*, 1995.

W. S. Lovejoy. Computationally feasible bounds for partially observed Markov decision processes. *Operations Research*, 39(1):162–175, 1991.

G. E. Monahan. A survey of partially observable Markov decision processes: theory, models and algorithms. *Management Science*, 28(1):1–16, 1982.

R. Parr and S. Russell. Approximating optimal policies for partially observable stochastic domains. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 1995.

J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 2003.

L. K. Platzman. A feasible computational approach to infinite-horizon partially-observed Markov decision problems. Technical Report J-81-2, School of Industrial and Systems Engineering, Georgia Institute of Technology, 1981. Reprinted in working notes AAAI 1998 Fall Symposium on Planning with POMDPs.

P. Poupart and C. Boutilier. Value-directed compression of POMDPs. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

P. Poupart and C. Boutilier. Bounded finite state controllers. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

N. Roy, G. Gordon, and S. Thrun. Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research*, 23:1–40, 2005.

J. K. Satia and R. E. Lave. Markovian decision processes with probabilistic observation of states. *Management Science*, 20(1):1–13, 1973.

G. Shani, R. I. Brafman, and S. E. Shimony. Forward search value iteration for POMDPs. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 2007.

D. Silver and J. Veness. Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems 23*, 2010.

S. Singh, T. Jaakkola, and M. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *International Conference on Machine Learning*, 1994.

R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov decision processes over a finite horizon. *Operations Research*, 21: 1071–1088, 1973.

T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proc. of Uncertainty in Artificial Intelligence*, 2004.

E. J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

Y. Virin, G. Shani, S. E. Shimony, and R. Brafman. Scaling up: Solving POMDPs through value based clustering. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 2007.

E. Walraven and M. T. J. Spaan. Accelerated vector pruning for optimal POMDP solvers. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3672–3678, 2017.

N. L. Zhang and W. Liu. Planning in stochastic domains: problem characteristics and approximations. Technical Report HKUST-CS96-31, Department of Computer Science, The Hong Kong University of Science and Technology, 1996.

R. Zhou and E. A. Hansen. An improved grid-based approximation algorithm for POMDPs. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 2001.