L6: Partially Observable MDPs

EECE 571N | Sequential Decision Making | Fall 2025

Cyrus Neary | cyrus.neary@ubc.ca

So far…

Markov Decision Process

$$\mathscr{M} = (S, A, T, R, \gamma, \mu)$$

Bellman's optimality equations

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^*(s') \right]$$

Policy iteration

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*,$$

For k = 0, 1, 2, …

$V_{k+1} \leftarrow T_\pi V_k$

$$\pi'(s) \in \text{argmax}_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^\pi(s') \right]$$

Value iteration

For k = 0, 1, 2, …

$V_{k+1} \leftarrow T_* V_k$

$$\pi^*(s) \in \text{argmax}_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V^*(s') \right]$$

LP solutions

$$\max_{x_{s,a} \in \mathbb{R}^{|S| \cdot |A|}} \sum_{s \in S, a \in A} R(s,a) x_{s,a} \quad \text{s.t.} \quad \sum_{a \in A_{s'}} x_{s',a} = \alpha(s') + \gamma \sum_{s \in S, a \in A_s} T(s' \mid s, a) x_{s,a} \quad \forall a \in A_s, s \in S \qquad x_{s,a} \geq 0, \quad \forall a \in A_s, s \in S$$

What critical assumptions have we been making?

- Markov processes, i.e. current state captures all information needed.
- When do we not have Markov processes? When we can't perfectly observe all the relevant information in the system state.

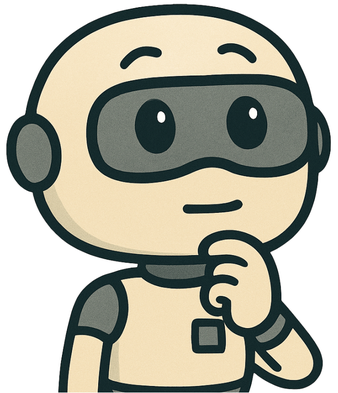
Today POMDPs: → So far our policies are designed with full observation/Markov property in mind.

→ In most situations we cannot fully observe the system state.

→ Control → Noisy sensor

→ Robotics → Only have image observations.

- Time invariance.



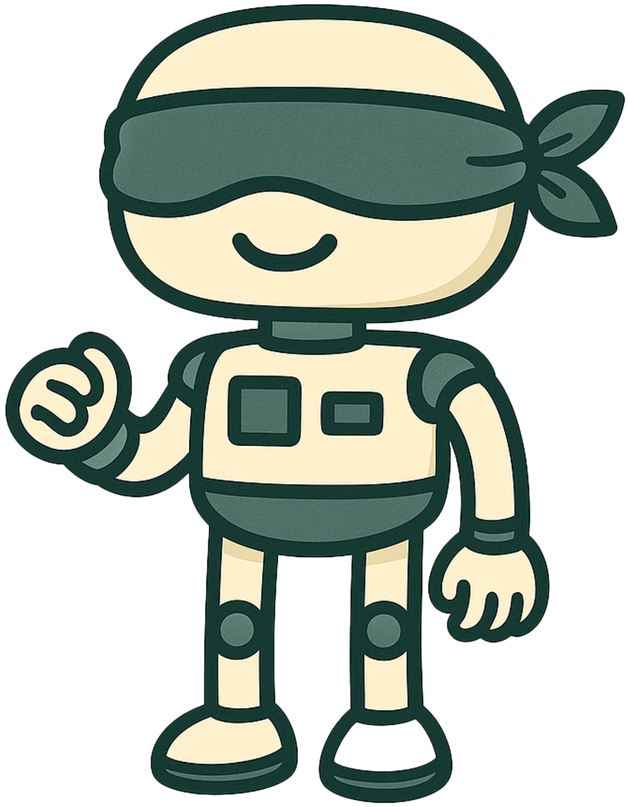
- We have access to a model of the system. i.e. $T(s'|s,a)$ and $R(s,a)$

→ if we remove this assumption, we need to approximate these functions from data while also solving for an optimal policy. Reinforcement Learning setting. Next class.

What if we can’t fully observe the system state?

For example…

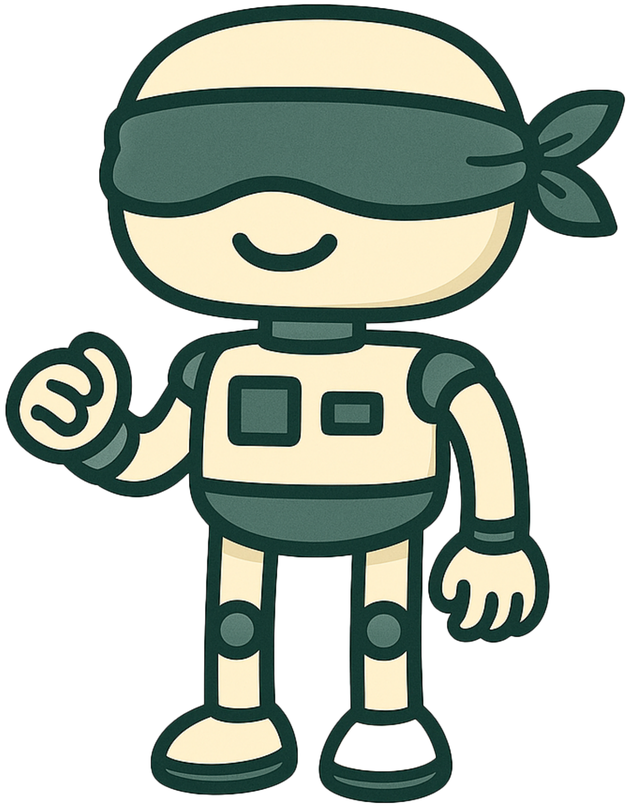
- A robot only observes its environment with egocentric image observations, not a top-down view of the entire world.
- A control policy that relies on velocities only has access to position sensors.
- A poker-playing bot can only see its own cards, and not the cards of its oponents.
- An automated inventory management system can’t measure customer demand, only indirectly observe it through sales.
- Most real world problems ARE partially observable.



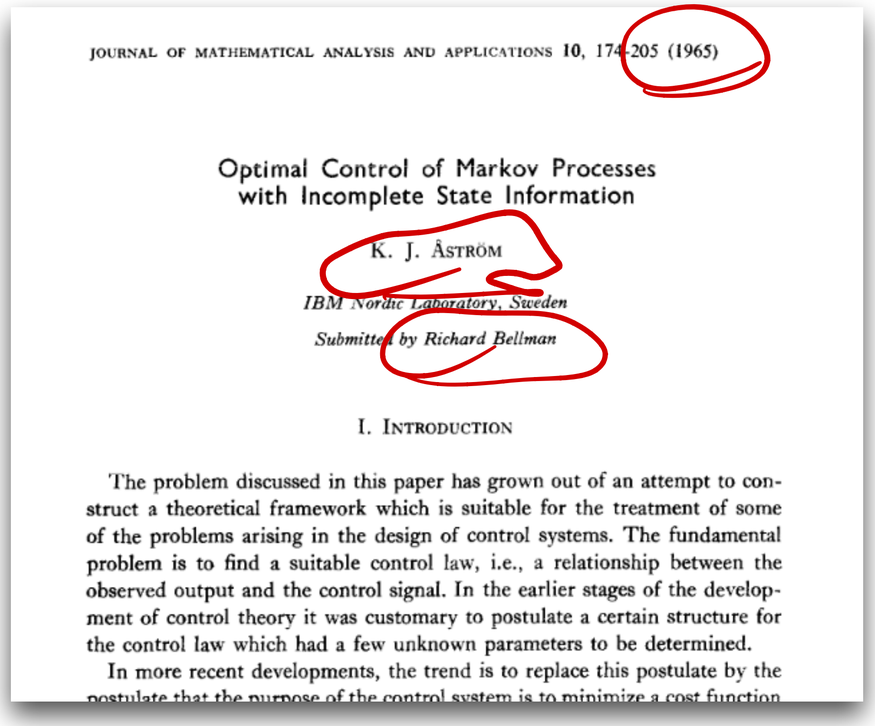
What if we can’t fully observe the system state?

The policies we discussed so far rely on knowing the complete system state at every decision epoch.

$$\pi(a \mid \cancel{s})$$



What if we can’t fully observe the system state?



JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 10, 174-205 (1965)

Optimal Control of Markov Processes with Incomplete State Information

K. J. ÅSTRÖM

IBM Nordic Laboratory, Sweden

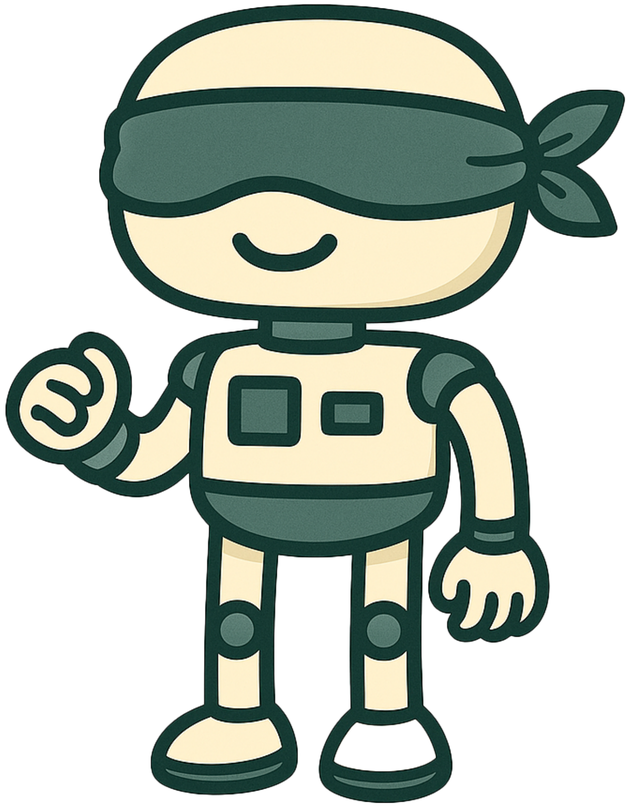
Submitted by Richard Bellman

I. INTRODUCTION

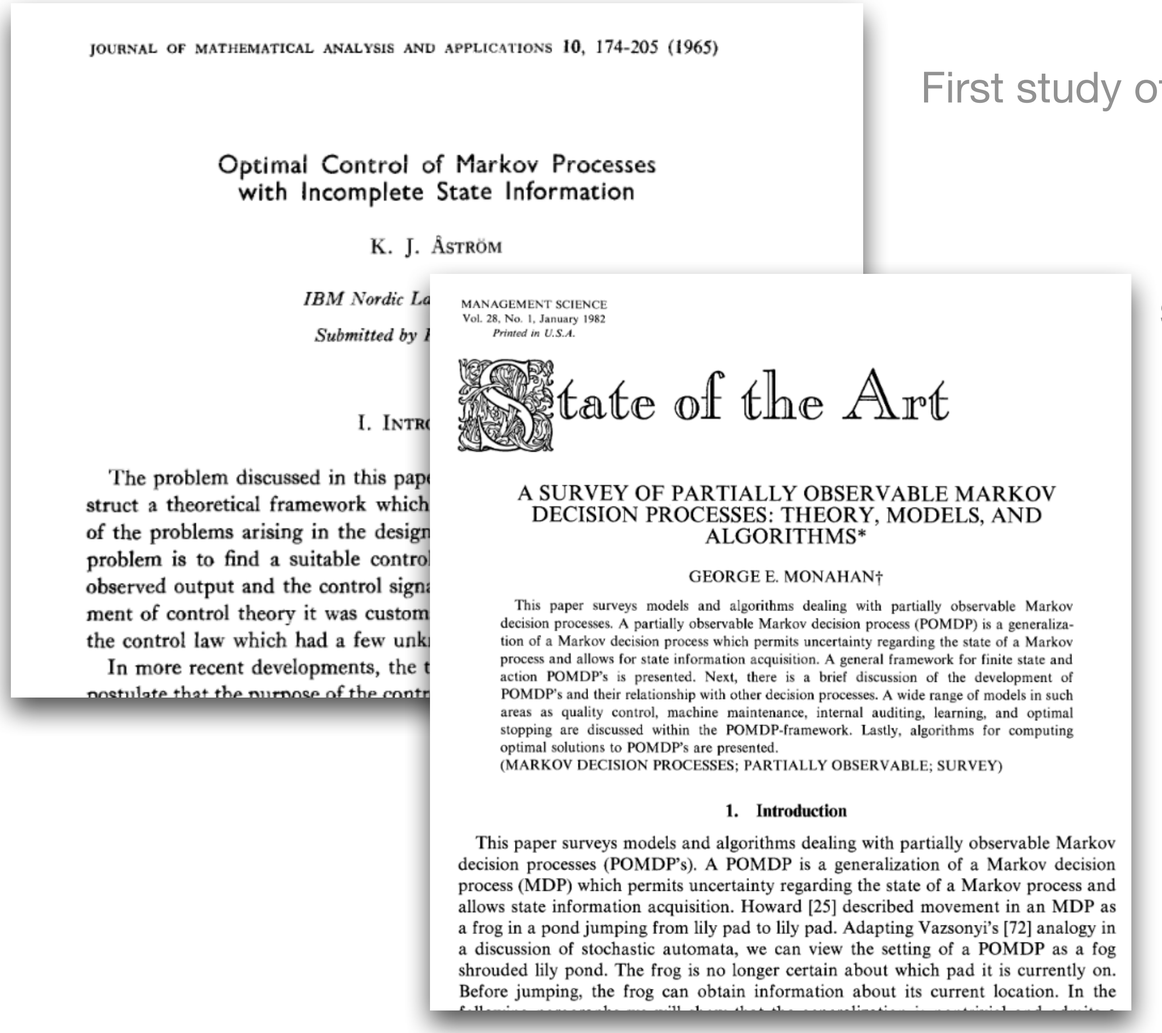
The problem discussed in this paper has grown out of an attempt to construct a theoretical framework which is suitable for the treatment of some of the problems arising in the design of control systems. The fundamental problem is to find a suitable control law, i.e., a relationship between the observed output and the control signal. In the earlier stages of the development of control theory it was customary to postulate a certain structure for the control law which had a few unknown parameters to be determined.

In more recent developments, the trend is to replace this postulate by the postulate that the purpose of the control system is to minimize a cost function

First study of partially observable MDPs (POMDPs)



What if we can't fully observe the system state?



JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 10, 174-205 (1965)

Optimal Control of Markov Processes with Incomplete State Information

K. J. ÅSTRÖM

IBM Nordic La

Submitted by

I. INTRO

The problem discussed in this pape
struct a theoretical framework which
of the problems arising in the design
problem is to find a suitable control
observed output and the control signa
ment of control theory it was custom
the control law which had a few unk
In more recent developments, the t
postulate that the purpose of the contr

MANAGEMENT SCIENCE
Vol. 28, No. 1, January 1982
Printed in U.S.A.

State of the Art

A SURVEY OF PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES: THEORY, MODELS, AND ALGORITHMS*

GEORGE E. MONAHAN†

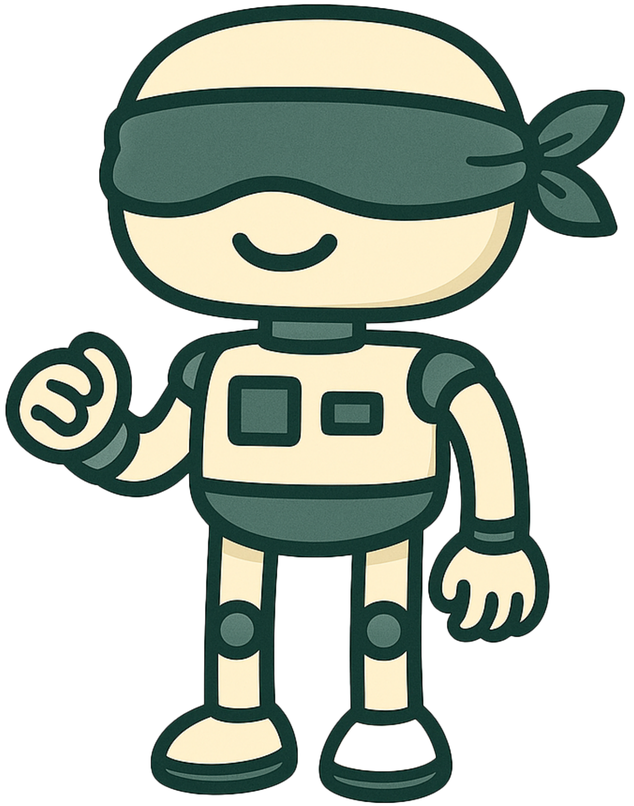
This paper surveys models and algorithms dealing with partially observable Markov decision processes. A partially observable Markov decision process (POMDP) is a generalization of a Markov decision process which permits uncertainty regarding the state of a Markov process and allows for state information acquisition. A general framework for finite state and action POMDP's is presented. Next, there is a brief discussion of the development of POMDP's and their relationship with other decision processes. A wide range of models in such areas as quality control, machine maintenance, internal auditing, learning, and optimal stopping are discussed within the POMDP-framework. Lastly, algorithms for computing optimal solutions to POMDP's are presented.
(MARKOV DECISION PROCESSES; PARTIALLY OBSERVABLE; SURVEY)

1. Introduction

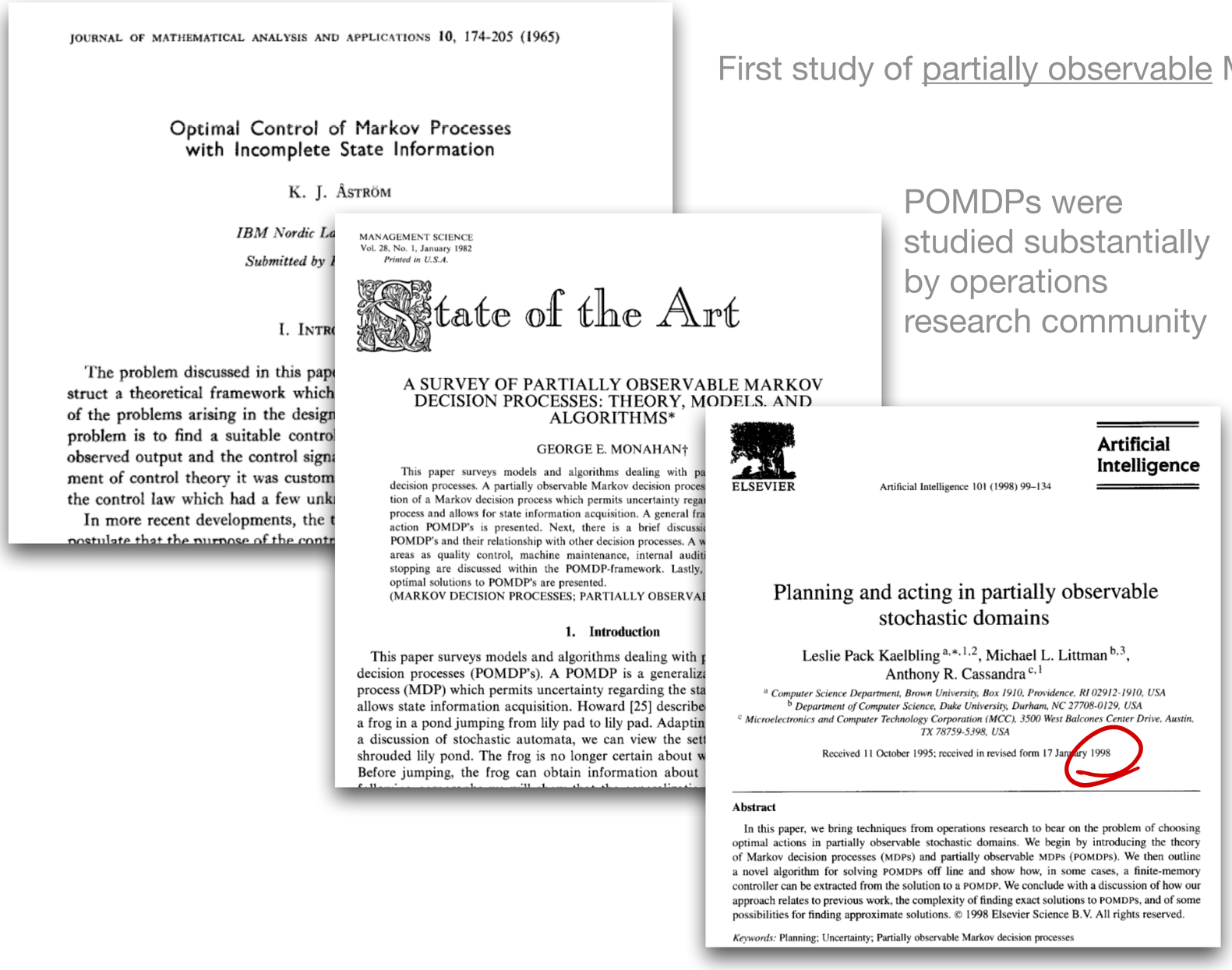
This paper surveys models and algorithms dealing with partially observable Markov decision processes (POMDP's). A POMDP is a generalization of a Markov decision process (MDP) which permits uncertainty regarding the state of a Markov process and allows state information acquisition. Howard [25] described movement in an MDP as a frog in a pond jumping from lily pad to lily pad. Adapting Vazsonyi's [72] analogy in a discussion of stochastic automata, we can view the setting of a POMDP as a fog shrouded lily pond. The frog is no longer certain about which pad it is currently on. Before jumping, the frog can obtain information about its current location. In the

First study of partially observable MDPs (POMDPs)

POMDPs were studied substantially by operations research community



What if we can't fully observe the system state?



JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 10, 174-205 (1965)

Optimal Control of Markov Processes with Incomplete State Information

K. J. ÅSTRÖM

IBM Nordic La

Submitted by

I. INTRO

The problem discussed in this pape
struct a theoretical framework which
of the problems arising in the design
problem is to find a suitable control
observed output and the control signa
ment of control theory it was custom
the control law which had a few unk
In more recent developments, the t
postulate that the purpose of the contr

MANAGEMENT SCIENCE
Vol. 28, No. 1, January 1982
Printed in U.S.A.

State of the Art

A SURVEY OF PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES: THEORY, MODELS, AND ALGORITHMS*

GEORGE E. MONAHAN†

This paper surveys models and algorithms dealing with pa
decision processes. A partially observable Markov decision proces
tion of a Markov decision process which permits uncertainty rega
process and allows for state information acquisition. A general fra
action POMDP's is presented. Next, there is a brief discussi
POMDP's and their relationship with other decision processes. A w
areas as quality control, machine maintenance, internal auditi
stopping are discussed within the POMDP-framework. Lastly,
optimal solutions to POMDP's are presented.
(MARKOV DECISION PROCESSES; PARTIALLY OBSERVA

1. Introduction

This paper surveys models and algorithms dealing with p
decision processes (POMDP's). A POMDP is a generaliz
process (MDP) which permits uncertainty regarding the sta
allows state information acquisition. Howard [25] describe
a frog in a pond jumping from lily pad to lily pad. Adaptin
a discussion of stochastic automata, we can view the sett
shrouded lily pond. The frog is no longer certain about w
Before jumping, the frog can obtain information about

ELSEVIER

Artificial Intelligence 101 (1998) 99–134

Artificial Intelligence

Planning and acting in partially observable stochastic domains

Leslie Pack Kaelbling [a,*,1,2], Michael L. Littman [b,3], Anthony R. Cassandra [c,1]

[a] *Computer Science Department, Brown University, Box 1910, Providence, RI 02912-1910, USA*
[b] *Department of Computer Science, Duke University, Durham, NC 27708-0129, USA*
[c] *Microelectronics and Computer Technology Corporation (MCC), 3500 West Balcones Center Drive, Austin, TX 78759-5398, USA*

Received 11 October 1995; received in revised form 17 January 1998

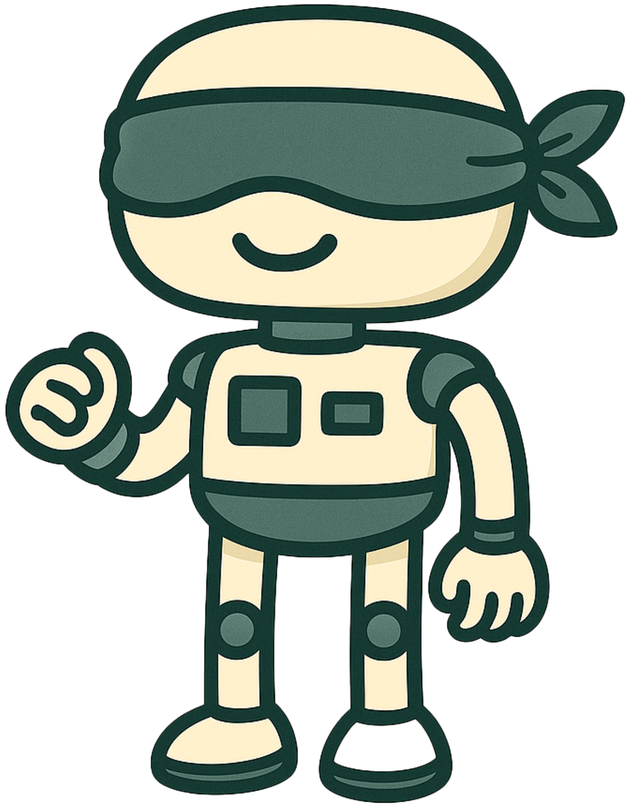
Abstract

In this paper, we bring techniques from operations research to bear on the problem of choosing optimal actions in partially observable stochastic domains. We begin by introducing the theory of Markov decision processes (MDPs) and partially observable MDPs (POMDPs). We then outline a novel algorithm for solving POMDPs off line and show how, in some cases, a finite-memory controller can be extracted from the solution to a POMDP. We conclude with a discussion of how our approach relates to previous work, the complexity of finding exact solutions to POMDPs, and of some possibilities for finding approximate solutions.

Keywords: Planning; Uncertainty; Partially observable Markov decision processes

First study of partially observable MDPs (POMDPs)

POMDPs were studied substantially by operations research community



And also became a general modeling framework for AI/RL researchers.

Partially observable Markov decision processes

$$\mathscr{M} = (S, A, T, R, \gamma, \mu, \Omega, O)$$

exactly the same as an MDP.

Partially observable Markov decision processes

$$\mathscr{M} = (S, A, T, R, \gamma, \mu, \Omega, O)$$

Ω - Observation set

Set of all possible observations
the agent could make.

Partially observable Markov decision processes

$$\mathscr{M} = (S, A, T, R, \gamma, \mu, \Omega, O)$$

Ω - Observation set

Set of all possible observations the agent could make.

eg. could be set of images a robot could observe.

Could be a boolean value representing whether some proposition is true in the environment.

O - Observation function

Function $O(o \mid s', a)$ mapping states and actions to distributions over observations $o \in \Omega$.

Probability that agent makes observation o, given it took action a, and arrived in state s'.

Policies in POMDPs

$$\mathscr{M} = (S, A, T, R, \gamma, \mu, \Omega, O)$$

Before ⇒ we knew s_t $\forall t$.
Now ⇒ we only gain information about s_t through observation o_t.

State feedback policies are no longer enough! $\pi(a \mid s)$

In POMDPs, the agent doesn't have access to $s \in S$. **POMDP policies need memory!**

In general, policy must rely on histories of observations and actions $\pi(a \mid h_t)$, where $h_t = o_0, a_0, o_1, a_1, \ldots, o_t$.

Policies in POMDPs

$$\mathscr{M} = (S, A, T, R, \gamma, \mu, \Omega, O)$$

State feedback policies are no longer enough! $\pi(a \mid s)$

In POMDPs, the agent doesn't have access to $s \in S$. **POMDP policies need memory!**

In general, policy must rely on histories of observations and actions $\pi(a \mid h_t)$, where $h_t = o_0, a_0, o_1, a_1, \ldots, o_t$.

A history dependent policy will, in general, require memory that grows exponentially in the time horizon.

Policies in POMDPs

$$\mathscr{M} = (S, A, T, R, \gamma, \mu, \Omega, O)$$

State feedback policies are no longer enough! $\pi(a \mid s)$

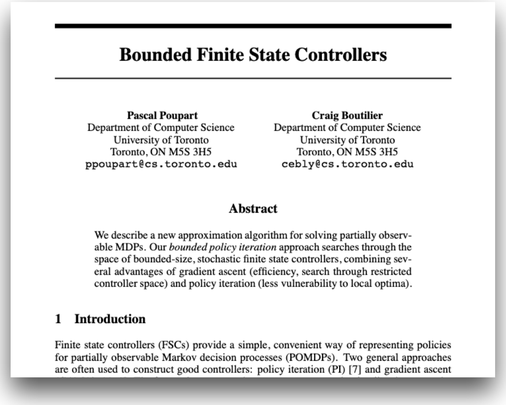
In POMDPs, the agent doesn't have access to $s \in S$. **POMDP policies need memory!**

In general, policy must rely on histories of observations and actions $\pi(a \mid h_t)$, where $h_t = o_0, a_0, o_1, a_1, \ldots, o_t$.

A history dependent policy will, in general, require memory that grows exponentially in the time horizon.

In some cases, one can extract a finite-state controller to represent a controller with a finite amount of memory.

These controllers can approximate optimal controllers, but memory requirements may grow substantially in general cases.



Bounded Finite State Controllers

Pascal Poupart
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
ppoupart@cs.toronto.edu

Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
cebly@cs.toronto.edu

Abstract

We describe a new approximation algorithm for solving partially observable MDPs. Our *bounded policy iteration* approach searches through the space of bounded-size, stochastic finite state controllers, combining several advantages of gradient ascent (efficiency, search through restricted controller space) and policy iteration (less vulnerability to local optima).

1 Introduction

Finite state controllers (FSCs) provide a simple, convenient way of representing policies for partially observable Markov decision processes (POMDPs). Two general approaches are often used to construct good controllers: policy iteration (PI) [7] and gradient ascent

Belief states in POMDPs

A history dependent policy will, in general, require memory that grows exponentially in the time horizon.

Alternatively, define policies and value functions over *belief states* $b \in \Delta(S)$.

- Each belief state represents a *distribution* over states $s \in S$.
  - Intuition: Instead of knowing the current state, we maintain a probability distribution describing what we believe the system state *could* be.

$$b \in \Delta(S), \quad b(s) \geq 0, \quad \sum_{s \in S} b(s) = 1$$

I.e., for finite state spaces, $\mathcal{B} = \{(b(s_1), b(s_2), \ldots, b(s_{|S|}) \in \mathbb{R}^{|S|} \mid b(s_i) \geq 0, \sum_i b(s_i) = 1\}$

Belief state updates are Bayesian filters

Given an initial belief state b_t, an action, an observation $o_{t+1} \in \Omega$, and a POMDP model, we can update our belief state b_{t+1} as:

$$b_{t+1}(s') = \eta O(o_{t+1} \mid s', a_t) \sum_{s \in S} T(s' \mid s, a_t) b_t(s), \quad \forall s' \in S$$

Correction | Prediction

Where η is a normalizing constant. $\eta = \frac{1}{\sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)}$

Define $\tau(b, a, o)$ to denote the Bayesian belief updator, which maps belief b, action a, and observation o to an updated belief vector b'.

Belief state updates are Bayesian filters

Given an initial belief state b_t, an action, an observation $o_{t+1} \in \Omega$, and a POMDP model, we can update our belief state b_{t+1} as:

$$b_{t+1}(s') = \eta O(o_{t+1} \mid s', a_t) \sum_{s \in S} T(s' \mid s, a_t) b_t(s), \quad \forall s' \in S$$

Correction | Prediction

Where η is a normalizing constant. $\eta = \frac{1}{\sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)}$

Define $\tau(b, a, o)$ to denote the Bayesian belief updator, which maps belief b, action a, and observation o to an updated belief vector b'.

Idea: belief states contain all the necessary information (memory) from histories of observations.

Note: an commonly-used example of such belief updates are Kalman filters. These provide closed-form belief updates for POMDPs with linear dynamics, linear observation models, and observations with Gaussian noise.

Deriving belief update $\forall s'$

$$b_{t+1}(s') = \eta O(o_{t+1} \mid s', a_t) \sum_{s \in S} T(s' \mid s, a_t)\, b_t(s)$$

We want $b_{t+1}(s') = Pr(s_{t+1} = s' \mid h_t, a_t, o_{t+1})$

h_t = history up to time t.

Predictor Marginalize over the current state s_t

$$\bar{b}_{t+1}(s') = Pr(s_{t+1} = s' \mid h_t, a_t) = \sum_{s \in S} Pr(s_{t+1} = s', s_t = s \mid h_t, a_t)$$

$$= \sum_{s \in S} \underbrace{Pr(s_{t+1} = s' \mid s_t = s, h_t, a_t)}_{T(s' \mid s, a_t)} \cdot \underbrace{Pr(s_t = s \mid h_t, a_t)}_{b_t(s)}$$

$$\bar{b}_{t+1}(s') = \sum_{s \in S} T(s' \mid s, a_t) \cdot b_t(s)$$

Corrector step Use Baye's rule with the new observation o_{t+1}.

Bayes rule: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

$\Rightarrow$ we want to write $b_{t+1}(s')$ in terms of observation model $O(o_{t+1}|s', a_t)$ and prior $\bar{b}_{t+1}(s')$

$$b_{t+1}(s') = \frac{Pr(o_{t+1}|s', a_t) \cdot Pr(s'|h_t, a_t)}{Pr(o_{t+1}|h_t, a_t)}$$

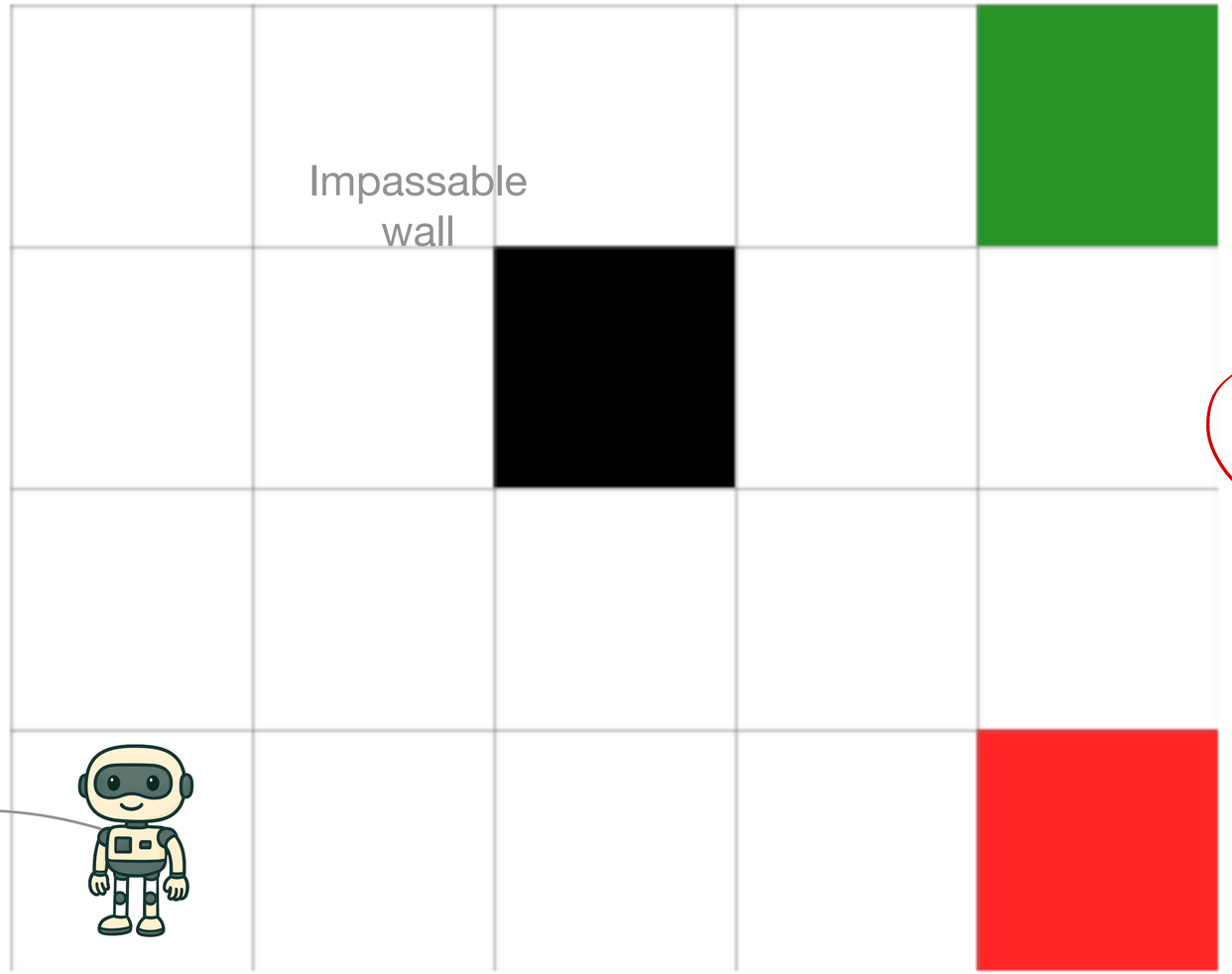
$$= \frac{O(o_{t+1}|s', a_t) \cdot \bar{b}_{t+1}(s')}{\sum_{s \in S} O(o_{t+1}|s', a_t) \bar{b}_{t+1}(s')}$$

and belief update formula

$$b_{t+1}(s') = \eta O(o_{t+1}|s', a_t) \sum_{s \in S} T(s'|s, a) b_t(s)$$

where $\eta = \frac{1}{\sum_{s'} O(o_{t+1}|s', a) \sum_{s} T(s'|s, a) b_t(s)}$

An extreme example of Bayesian filtering: Gridworld with no observations



Goal state
+1 reward
absorbing

Impassable
wall

Every other step:
-0.1 reward

Dynamics:
“Slip probability” of 0.3

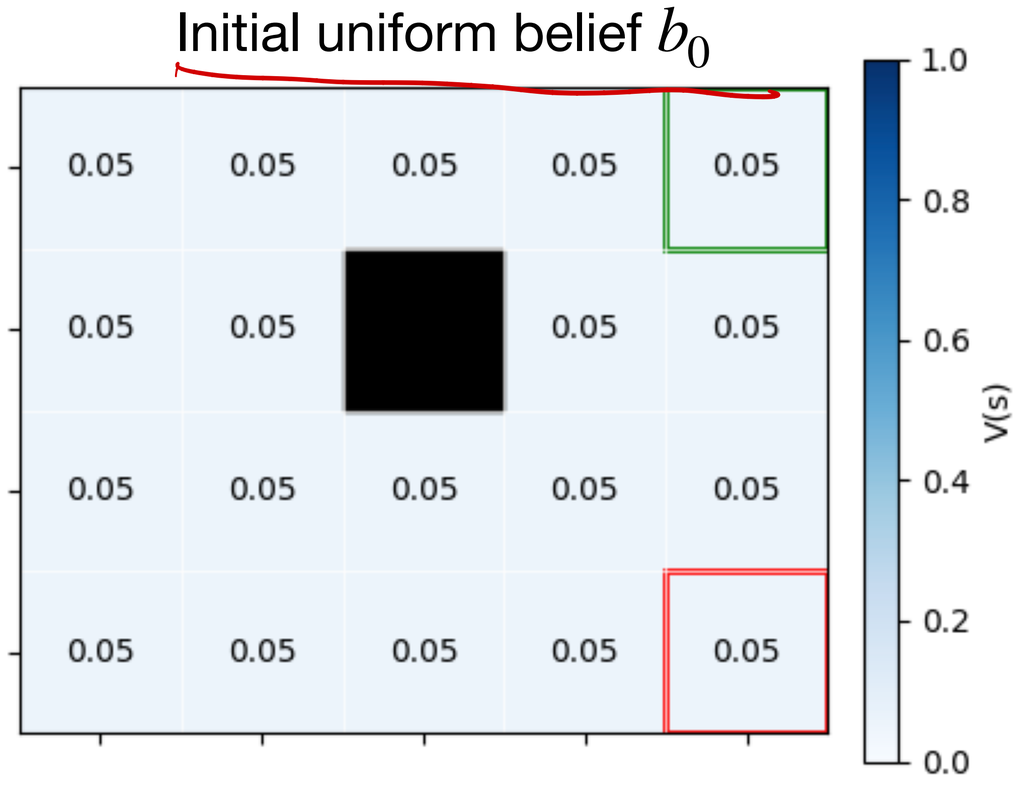
Initial state

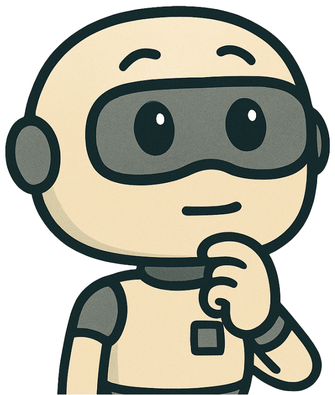
Sink state
-1 reward
absorbing

Consider a meaningless observations and observation function: $\Omega = None$

An extreme example of Bayesian filtering: Gridworld with no observations

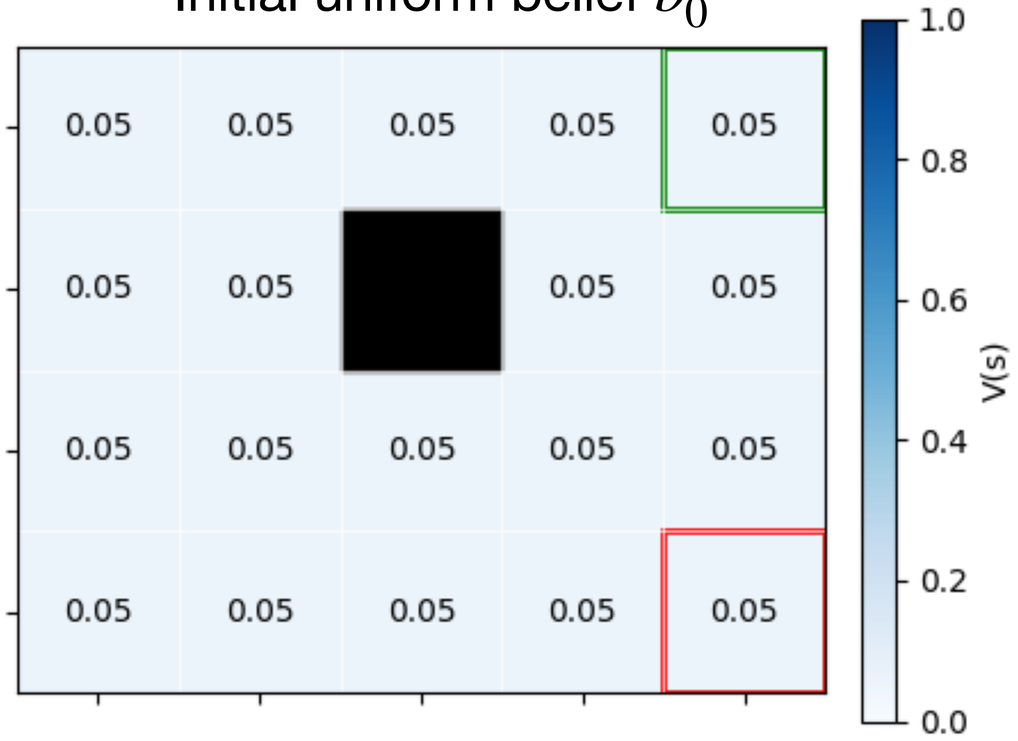
What happens to our belief state if we repeatedly take a particular action in this environment?



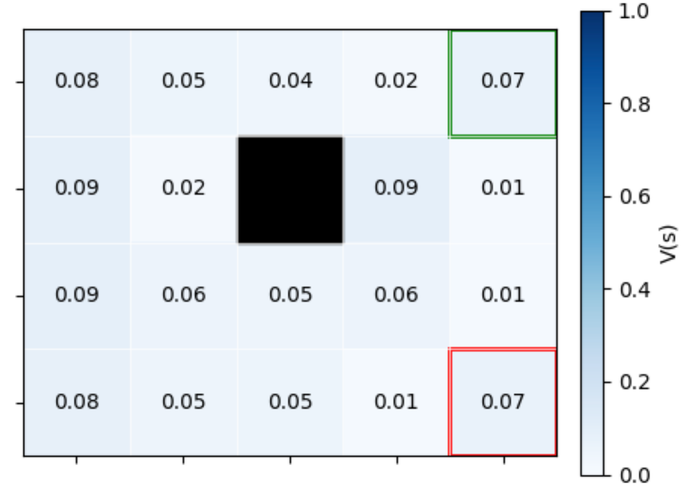


An extreme example of Bayesian filtering: Gridworld with no observations

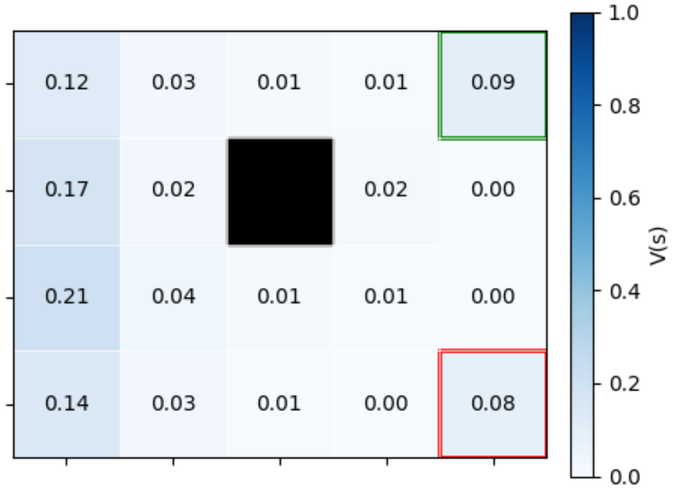
Initial uniform belief b_0



Always take action a = move left



b_1



b_{10}

POMDP solution idea: belief MDP

Given a POMDP $(S, A, T, R, \gamma, \mu, \Omega, O)$, construct an equivalent MDP $\tilde{\mathscr{M}} = (\tilde{S}, A, \tilde{T}, \tilde{R}, \gamma, b_{init})$ over belief states as follows:

POMDP solution idea: belief MDP

Given a POMDP $(S, A, T, R, \gamma, \mu, \Omega, O)$, construct an equivalent MDP $\tilde{\mathscr{M}} = (\tilde{S}, A, \tilde{T}, \tilde{R}, \gamma, b_{init})$ over belief states as follows:

- Belief MDP state space is defined as the POMDP belief space.

$$\tilde{S} = \mathscr{B} = \Delta(S)$$

POMDP solution idea: belief MDP

Given a POMDP $(S, A, T, R, \gamma, \mu, \Omega, O)$, construct an equivalent MDP $\tilde{\mathscr{M}} = (\tilde{S}, A, \tilde{T}, \tilde{R}, \gamma, b_{init})$ over belief states as follows:

- Belief MDP state space is defined as the POMDP belief space.

$$\tilde{S} = \mathscr{B} = \Delta(S)$$

- Action space A remains the same.

POMDP solution idea: belief MDP

Given a POMDP $(S, A, T, R, \gamma, \mu, \Omega, O)$, construct an equivalent MDP $\tilde{\mathscr{M}} = (\tilde{S}, A, \tilde{T}, \tilde{R}, \gamma, b_{init})$ over belief states as follows:

- Belief MDP state space is defined as the POMDP belief space.

$$\tilde{S} = \mathscr{B} = \Delta(S)$$

- Action space A remains the same.
- Transition function $\tilde{T}$ maps belief states to distributions over belief states via Bayesian filtering.

$$\tilde{T}(b' \mid b, a) = \sum_{o \in \Omega} \mathbb{I}\{b' = \tau(b, a, o)\} P(o \mid a, b)$$

Where $P(o \mid b, a) = \sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)$, and $\mathbb{I}\{\}$ is the indicator function.

POMDP solution idea: belief MDP

Given a POMDP $(S, A, T, R, \gamma, \mu, \Omega, O)$, construct an equivalent MDP $\tilde{\mathscr{M}} = (\tilde{S}, A, \tilde{T}, \tilde{R}, \gamma, b_{init})$ over belief states as follows:

- Belief MDP state space is defined as the POMDP belief space.

$$\tilde{S} = \mathscr{B} = \Delta(S)$$

- Action space A remains the same.
- Transition function $\tilde{T}$ maps belief states to distributions over belief states via Bayesian filtering.

$$\tilde{T}(b' \mid b, a) = \sum_{o \in \Omega} \mathbb{I}\{b' = \tau(b, a, o)\} P(o \mid a, b)$$

Where $P(o \mid b, a) = \sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)$, and $\mathbb{I}\{\}$ is the indicator function.

- Reward function $\tilde{R}(b, a)$ is defined as $\tilde{R}(b, a) = \sum_{s \in S} R(s, a) b(s)$

POMDP solution idea: belief MDP

Given a POMDP $(S, A, T, R, \gamma, \mu, \Omega, O)$, construct an equivalent MDP $\tilde{\mathscr{M}} = (\tilde{S}, A, \tilde{T}, \tilde{R}, \gamma, b_{init})$ over belief states as follows:

- Belief MDP state space is defined as the POMDP belief space.

$$\tilde{S} = \mathscr{B} = \Delta(S)$$

- Action space A remains the same.
- Transition function $\tilde{T}$ maps belief states to distributions over belief states via Bayesian filtering.

$$\tilde{T}(b' \mid b, a) = \sum_{o \in \Omega} \mathbb{I}\{b' = \tau(b, a, o)\} P(o \mid a, b)$$

Where $P(o \mid b, a) = \sum_{s' \in S} O(o \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s)$, and $\mathbb{I}\{\}$ is the indicator function.

- Reward function $\tilde{R}(b, a)$ is defined as $\tilde{R}(b, a) = \sum_{s \in S} R(s, a) b(s)$
- $b_{init} \in \Delta(S)$ is an initial belief state.

POMDP solution idea: belief MDP

Solve the belief MDP using standard techniques (e.g., value iteration, policy iteration) to get optimal policy $\pi^*(b)$.

Typo

$$V^*(b) = \max_{a \in A} \left[\tilde{R}(b, a) + \gamma \sum_{o \in \Omega} P(o \mid b, a) V^*(\tau(b, a, o)) \right]$$

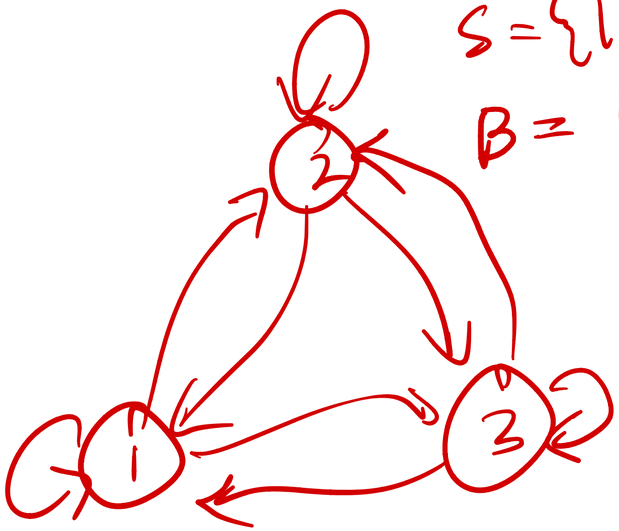
Recall, $\tau(b, a, o)$ to denotes the Bayesian belief updator, which maps belief b, action a, and observation o to an updated belief vector b' via the relationship:

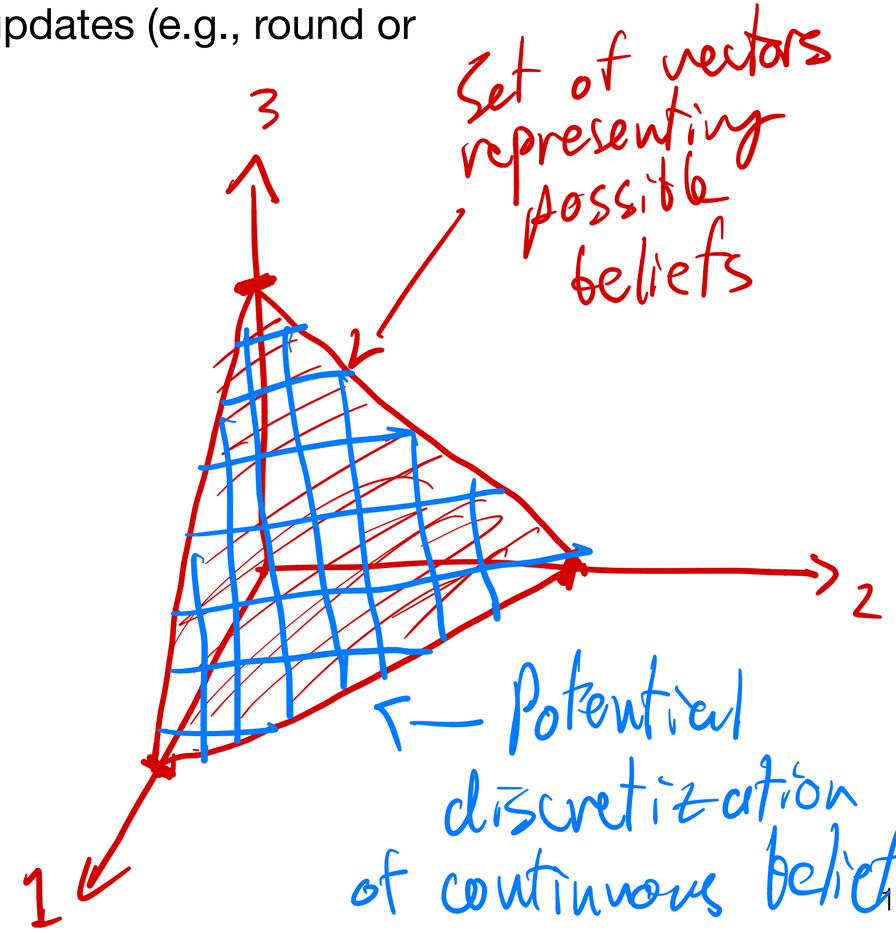
$$b'(s') = \tau(b, a, o)(s') = \eta O(o_{t+1} \mid s', a_t) \sum_{s \in S} T(s' \mid s, a_t) b_t(s), \quad \forall s' \in S$$

challenge: we can't efficiently compute Bellman backups for $V^*(b)$ because the belief state space is continuous!

Approximate POMDP solution idea: discretize the belief MDP

1. Pick a finite set of belief points (e.g., via a grid or sampling).
2. Treat these discretized points as the new “states” of your discrete belief MDP.
3. Define approximations between these points for belief updates (e.g., round or project $\tau(b, a, o)$ to the nearest grid point).
4. Solve the resulting MDP with finite states and actions.





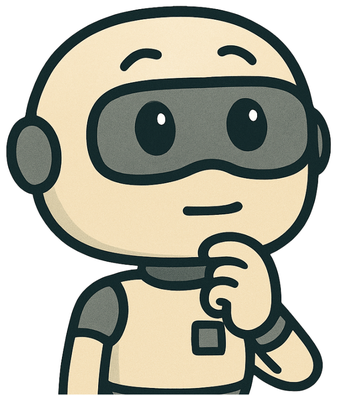
Approximate POMDP solution idea: discretize the belief MDP

Solve the belief MDP using standard techniques (e.g., value iteration, policy iteration) to get optimal policy $\pi^*(b)$.

What's the challenge with this approach?

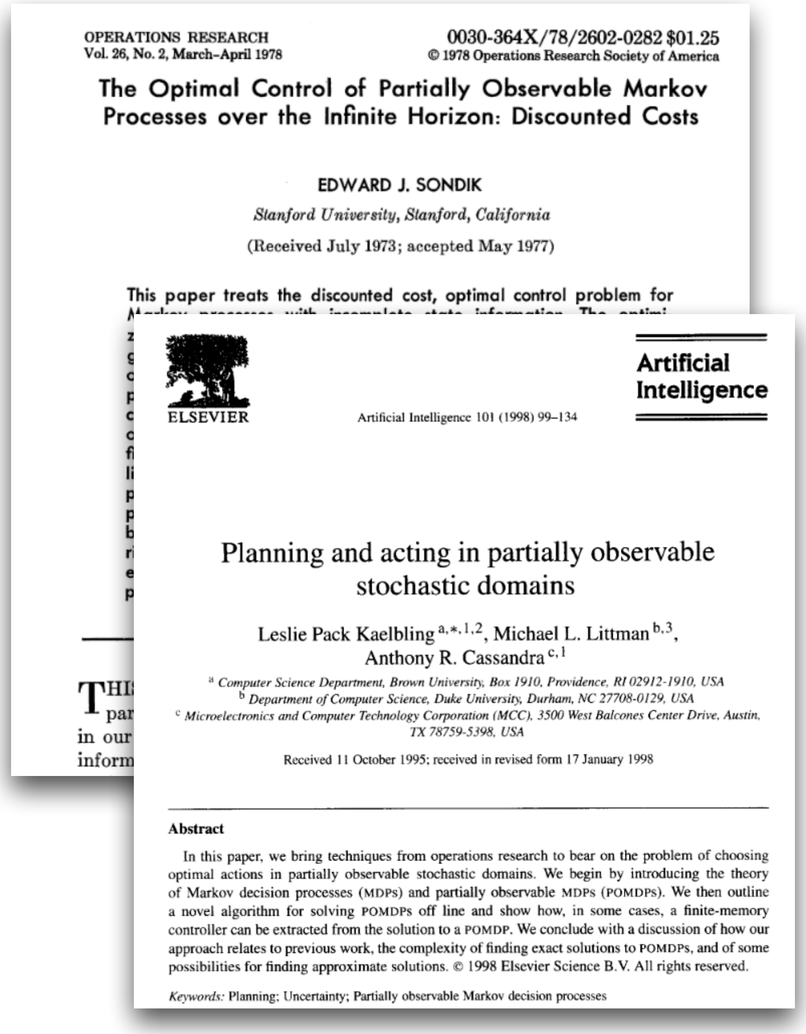
- State space is huge and continuous. For $|S|=n$, belief space is continuous and of dimension $n-1$.

1. Solving discrete MDP is intractable. Grid discretizing the belief space, suffers from curse of dimensionality.
   Grid resolution of m results in $m^{|S|-1}$ points.
   - Consider 19 state gridworld. Discretize each dimension into $k=20$ bins. Resulting discrete belief space has $(k+n+1)C(n-1) \simeq 3.36 \times 10^{10}$ states. (stars and bars combinatorial problem).



2. Backups require summations over every possible observation.
   Large obs spaces (e.g. Images) => Intractable Bayesian filtering.

Offline, exact solutions to POMDPs



OPERATIONS RESEARCH
Vol. 26, No. 2, March–April 1978

0030-364X/78/2602-0282 $01.25
© 1978 Operations Research Society of America

The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs

EDWARD J. SONDIK

Stanford University, Stanford, California

(Received July 1973; accepted May 1977)

This paper treats the discounted cost, optimal control problem for

ELSEVIER

Artificial Intelligence 101 (1998) 99–134

Artificial Intelligence

Planning and acting in partially observable stochastic domains

Leslie Pack Kaelbling [a,*,1,2], Michael L. Littman [b,3], Anthony R. Cassandra [c,1]

[a] *Computer Science Department, Brown University, Box 1910, Providence, RI 02912-1910, USA*
[b] *Department of Computer Science, Duke University, Durham, NC 27708-0129, USA*
[c] *Microelectronics and Computer Technology Corporation (MCC), 3500 West Balcones Center Drive, Austin, TX 78759-5398, USA*

Received 11 October 1995; received in revised form 17 January 1998

Abstract

In this paper, we bring techniques from operations research to bear on the problem of choosing optimal actions in partially observable stochastic domains. We begin by introducing the theory of Markov decision processes (MDPs) and partially observable MDPs (POMDPs). We then outline a novel algorithm for solving POMDPs off line and show how, in some cases, a finite-memory controller can be extracted from the solution to a POMDP. We conclude with a discussion of how our approach relates to previous work, the complexity of finding exact solutions to POMDPs, and of some possibilities for finding approximate solutions.

Keywords: Planning; Uncertainty; Partially observable Markov decision processes

For finite time horizon T, the belief MDP's value function $V_t^*(b)$ is piecewise-linear and convex.

$$V_t(b) = \max_{\alpha \in \Gamma_t} b \cdot \alpha$$

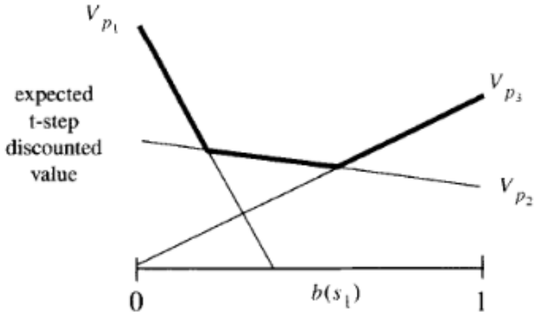
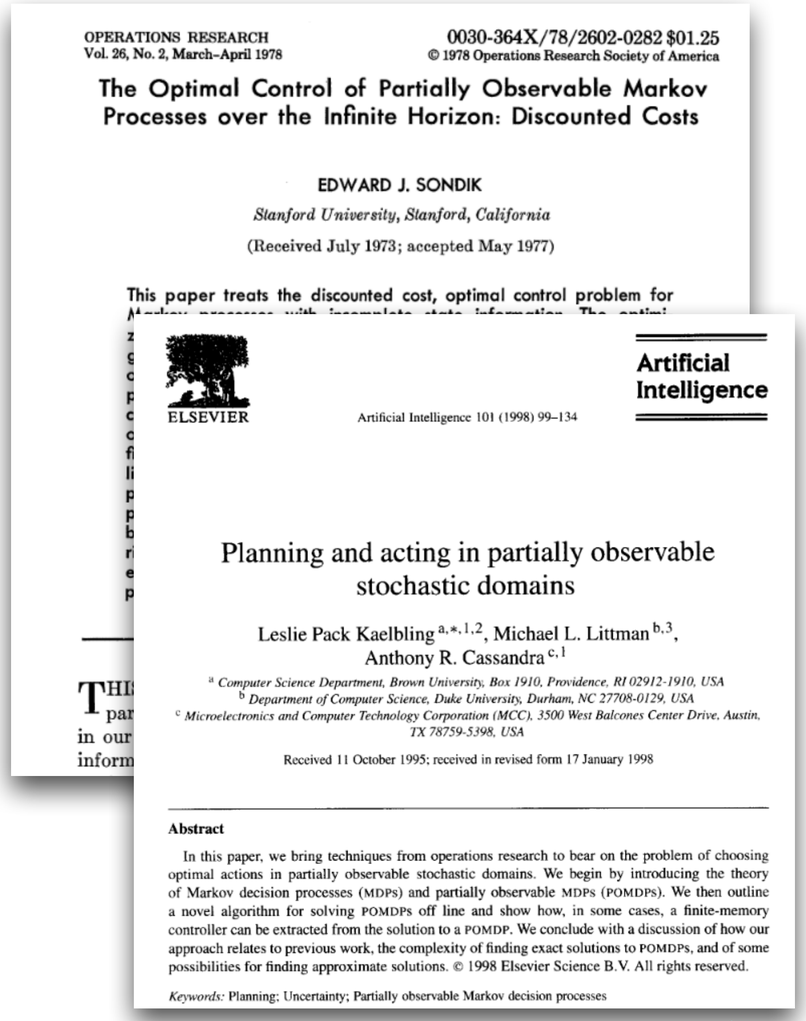


Image courtesy of L.P. Kaelbling et al (1998).

Where $\alpha \in \mathbb{R}^{|S|}$ is a vector representing the expected returns of a paricular finite-horizon plan, with one entry per underlying state.

Offline, exact solutions to POMDPs



OPERATIONS RESEARCH
Vol. 26, No. 2, March–April 1978

0030-364X/78/2602-0282 $01.25
© 1978 Operations Research Society of America

The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs

EDWARD J. SONDIK

Stanford University, Stanford, California

(Received July 1973; accepted May 1977)

This paper treats the discounted cost, optimal control problem for

ELSEVIER

Artificial Intelligence 101 (1998) 99–134

Artificial Intelligence

Planning and acting in partially observable stochastic domains

Leslie Pack Kaelbling [a,*,1,2], Michael L. Littman [b,3], Anthony R. Cassandra [c,1]

[a] *Computer Science Department, Brown University, Box 1910, Providence, RI 02912-1910, USA*
[b] *Department of Computer Science, Duke University, Durham, NC 27708-0129, USA*
[c] *Microelectronics and Computer Technology Corporation (MCC), 3500 West Balcones Center Drive, Austin, TX 78759-5398, USA*

Received 11 October 1995; received in revised form 17 January 1998

Abstract

In this paper, we bring techniques from operations research to bear on the problem of choosing optimal actions in partially observable stochastic domains. We begin by introducing the theory of Markov decision processes (MDPs) and partially observable MDPs (POMDPs). We then outline a novel algorithm for solving POMDPs off line and show how, in some cases, a finite-memory controller can be extracted from the solution to a POMDP. We conclude with a discussion of how our approach relates to previous work, the complexity of finding exact solutions to POMDPs, and of some possibilities for finding approximate solutions. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Planning; Uncertainty; Partially observable Markov decision processes

For finite time horizon T, the belief MDP's value function $V_t^*(b)$ is piecewise-linear and convex.

$$V_t(b) = \max_{\alpha \in \Gamma_t} b \cdot \alpha$$

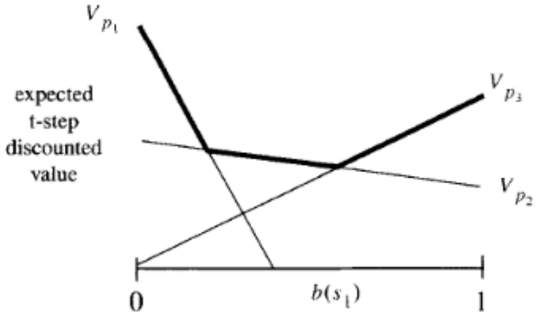


Image courtesy of L.P. Kaelbling et al (1998).

$k = |\Omega|$

Where $\alpha \in \mathbb{R}^{|S|}$ is a vector representing the expected returns of a paricular finite-horizon plan, with one entry per underlying state.

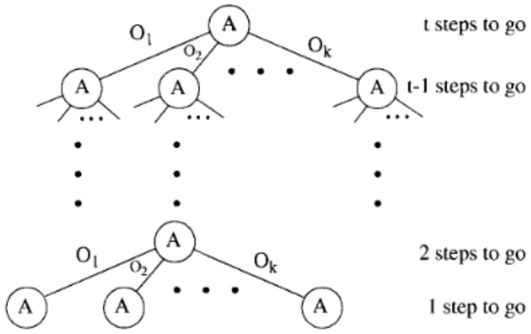
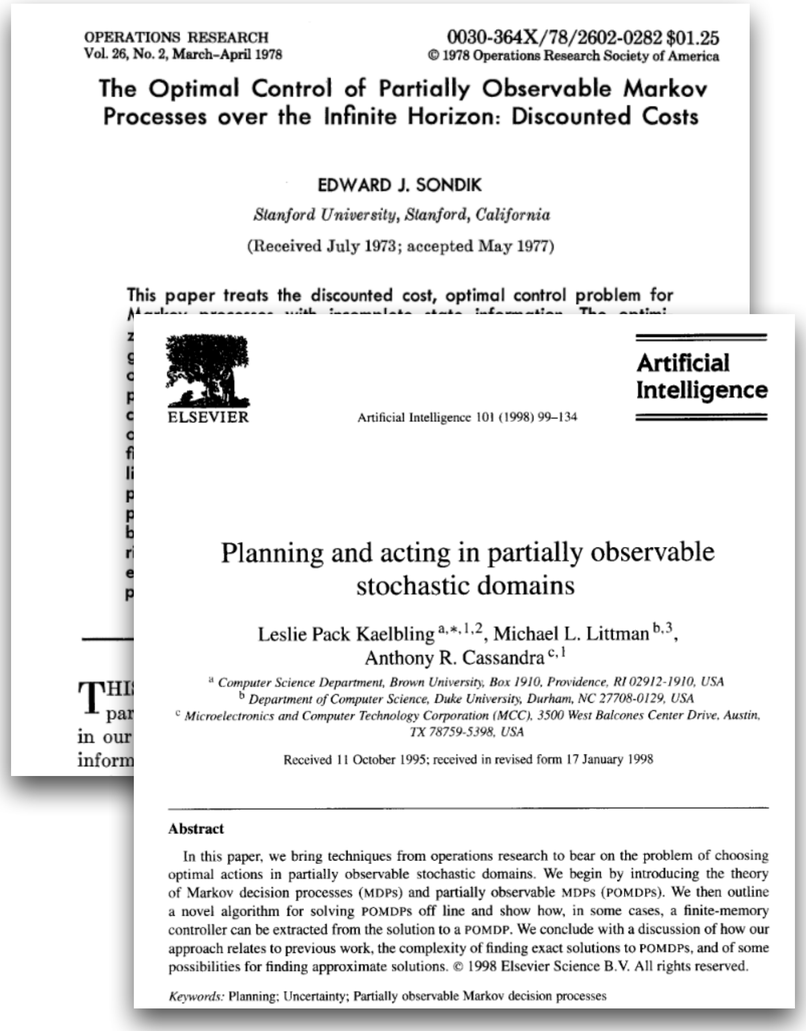


Image courtesy of L.P. Kaelbling et al (1998).

Recall that in finite-horizon settings, a stationary policy is not sufficient and we require a time-dependent policy (or plan).

challenge: Backup still requires consideration of all possible plans of depth T+1. # of plans grows exponentially in T and k.

Offline, exact solutions to POMDPs



OPERATIONS RESEARCH
Vol. 26, No. 2, March–April 1978

0030-364X/78/2602-0282 $01.25
© 1978 Operations Research Society of America

The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs

EDWARD J. SONDIK

Stanford University, Stanford, California

(Received July 1973; accepted May 1977)

This paper treats the discounted cost, optimal control problem for

ELSEVIER

Artificial Intelligence

Artificial Intelligence 101 (1998) 99–134

Planning and acting in partially observable stochastic domains

Leslie Pack Kaelbling [a,*,1,2], Michael L. Littman [b,3], Anthony R. Cassandra [c,1]

[a] *Computer Science Department, Brown University, Box 1910, Providence, RI 02912-1910, USA*
[b] *Department of Computer Science, Duke University, Durham, NC 27708-0129, USA*
[c] *Microelectronics and Computer Technology Corporation (MCC), 3500 West Balcones Center Drive, Austin, TX 78759-5398, USA*

Received 11 October 1995; received in revised form 17 January 1998

Abstract

In this paper, we bring techniques from operations research to bear on the problem of choosing optimal actions in partially observable stochastic domains. We begin by introducing the theory of Markov decision processes (MDPs) and partially observable MDPs (POMDPs). We then outline a novel algorithm for solving POMDPs off line and show how, in some cases, a finite-memory controller can be extracted from the solution to a POMDP. We conclude with a discussion of how our approach relates to previous work, the complexity of finding exact solutions to POMDPs, and of some possibilities for finding approximate solutions.

Keywords: Planning; Uncertainty; Partially observable Markov decision processes

For finite time horizon T, the belief MDP's value function $V_t^*(b)$ is piecewise-linear and convex.

$$V_t(b) = \max_{\alpha \in \Gamma_t} b \cdot \alpha$$

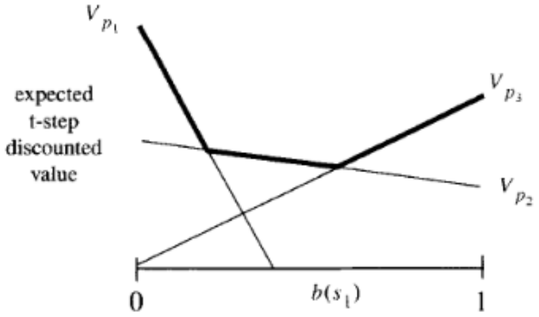


Image courtesy of L.P. Kaelbling et al (1998).

Where $\alpha \in \mathbb{R}^{|S|}$ is a vector representing the expected returns of a paricular finite-horizon plan, with one entry per underlying state.

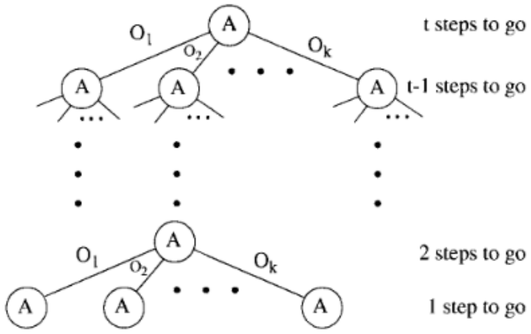


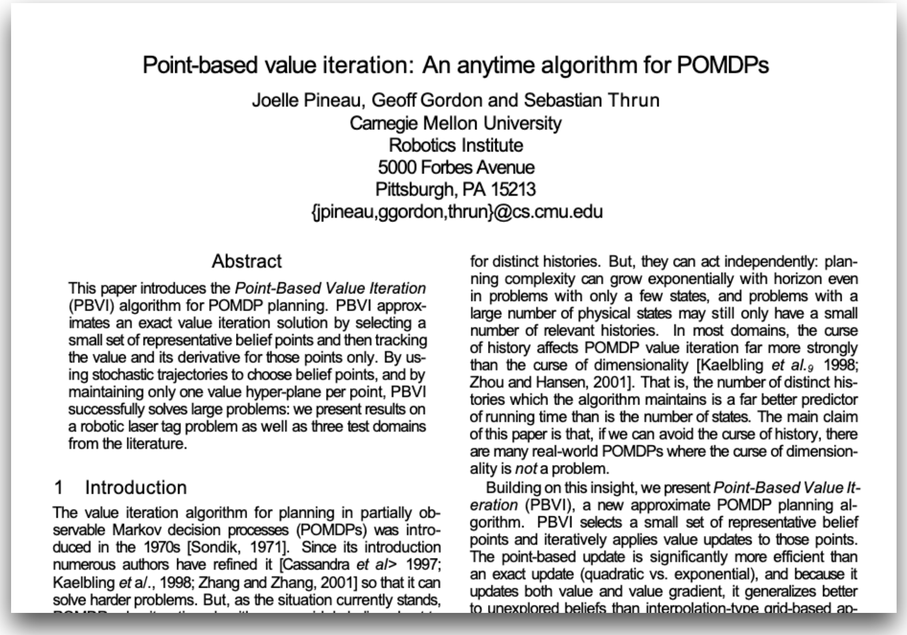
Image courtesy of L.P. Kaelbling et al (1998).

Recall that in finite-horizon settings, a stationary policy is not sufficient and we require a time-dependent policy (or plan).

This property enables algorithms for exact POMDP value iteration.

challenge: Backup still requires consideration of all possible plans of depth T-t. # of plans grows exponentially in T and k.

Offline, approximate solutions to POMDPs



Point-based value iteration: An anytime algorithm for POMDPs

Joelle Pineau, Geoff Gordon and Sebastian Thrun
Carnegie Mellon University
Robotics Institute
5000 Forbes Avenue
Pittsburgh, PA 15213
{jpineau,ggordon,thrun}@cs.cmu.edu

Abstract

This paper introduces the *Point-Based Value Iteration* (PBVI) algorithm for POMDP planning. PBVI approximates an exact value iteration solution by selecting a small set of representative belief points and then tracking the value and its derivative for those points only. By using stochastic trajectories to choose belief points, and by maintaining only one value hyper-plane per point, PBVI successfully solves large problems: we present results on a robotic laser tag problem as well as three test domains from the literature.

1 Introduction

The value iteration algorithm for planning in partially observable Markov decision processes (POMDPs) was introduced in the 1970s [Sondik, 1971]. Since its introduction numerous authors have refined it [Cassandra *et al*> 1997; Kaelbling *et a*/., 1998; Zhang and Zhang, 2001] so that it can solve harder problems. But, as the situation currently stands,

for distinct histories. But, they can act independently: planning complexity can grow exponentially with horizon even in problems with only a few states, and problems with a large number of physical states may still only have a small number of relevant histories. In most domains, the curse of history affects POMDP value iteration far more strongly than the curse of dimensionality [Kaelbling *et al.*, 1998; Zhou and Hansen, 2001]. That is, the number of distinct histories which the algorithm maintains is a far better predictor of running time than is the number of states. The main claim of this paper is that, if we can avoid the curse of history, there are many real-world POMDPs where the curse of dimensionality is *not* a problem.

Building on this insight, we present *Point-Based Value Iteration* (PBVI), a new approximate POMDP planning algorithm. PBVI selects a small set of representative belief points and iteratively applies value updates to those points. The point-based update is significantly more efficient than an exact update (quadratic vs. exponential), and because it updates both value and value gradient, it generalizes better to unexplored beliefs than interpolation-type grid-based ap-

Intuition: Most POMDP problems will never reach every point in the belief simplex. Computing solutions for those points does not help us.

Main idea: Approximate value function only at a selected finite set of belief points.

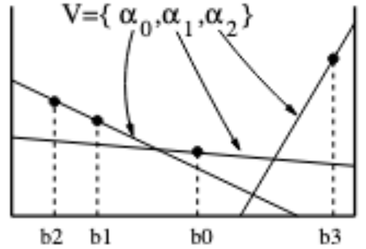
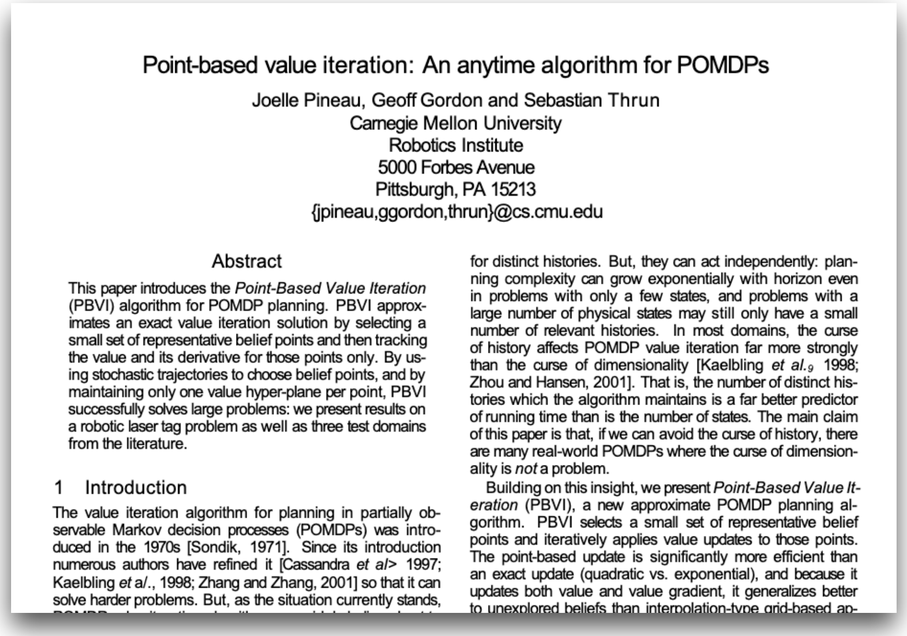


Image courtesy of Pineau et al (2003).

Offline, approximate solutions to POMDPs



Point-based value iteration: An anytime algorithm for POMDPs

Joelle Pineau, Geoff Gordon and Sebastian Thrun
Carnegie Mellon University
Robotics Institute
5000 Forbes Avenue
Pittsburgh, PA 15213
{jpineau,ggordon,thrun}@cs.cmu.edu

Abstract

This paper introduces the *Point-Based Value Iteration* (PBVI) algorithm for POMDP planning. PBVI approximates an exact value iteration solution by selecting a small set of representative belief points and then tracking the value and its derivative for those points only. By using stochastic trajectories to choose belief points, and by maintaining only one value hyper-plane per point, PBVI successfully solves large problems: we present results on a robotic laser tag problem as well as three test domains from the literature.

1 Introduction

The value iteration algorithm for planning in partially observable Markov decision processes (POMDPs) was introduced in the 1970s [Sondik, 1971]. Since its introduction numerous authors have refined it [Cassandra *et al*> 1997; Kaelbling *et a*/., 1998; Zhang and Zhang, 2001] so that it can solve harder problems. But, as the situation currently stands,

for distinct histories. But, they can act independently: planning complexity can grow exponentially with horizon even in problems with only a few states, and problems with a large number of physical states may still only have a small number of relevant histories. In most domains, the curse of history affects POMDP value iteration far more strongly than the curse of dimensionality [Kaelbling *et al.*, 1998; Zhou and Hansen, 2001]. That is, the number of distinct histories which the algorithm maintains is a far better predictor of running time than is the number of states. The main claim of this paper is that, if we can avoid the curse of history, there are many real-world POMDPs where the curse of dimensionality is *not* a problem.

Building on this insight, we present *Point-Based Value Iteration* (PBVI), a new approximate POMDP planning algorithm. PBVI selects a small set of representative belief points and iteratively applies value updates to those points. The point-based update is significantly more efficient than an exact update (quadratic vs. exponential), and because it updates both value and value gradient, it generalizes better to unexplored beliefs than interpolation-type grid-based ap-

Intuition: Most POMDP problems will never reach every point in the belief simplex. Computing solutions for those points does not help us.

Main idea: Approximate value function only at a selected finite set of belief points.

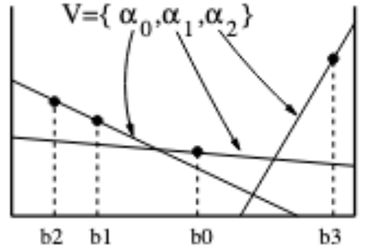
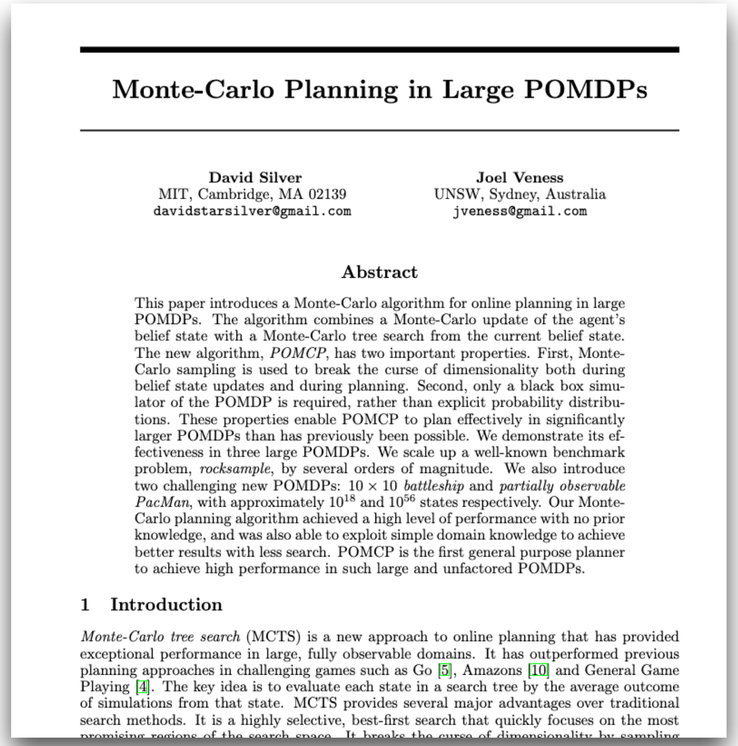


Image courtesy of Pineau et al (2003).

1. Initialize initial belief set $B = \{b_1, b_2, \ldots, b_m\}$.
2. Perform point-based value backups.
   - Perform value backups on the finite set of belief points.
3. Expand the belief set.
   - Add new beliefs by forward simulating trajectories under the current policy.
   - Prune low-importance beliefs.
4. Iterate steps 2-4 until convergence or time budget.

Online, approximate solutions to POMDPs



Monte-Carlo Planning in Large POMDPs

David Silver
MIT, Cambridge, MA 02139
davidstarsilver@gmail.com

Joel Veness
UNSW, Sydney, Australia
jveness@gmail.com

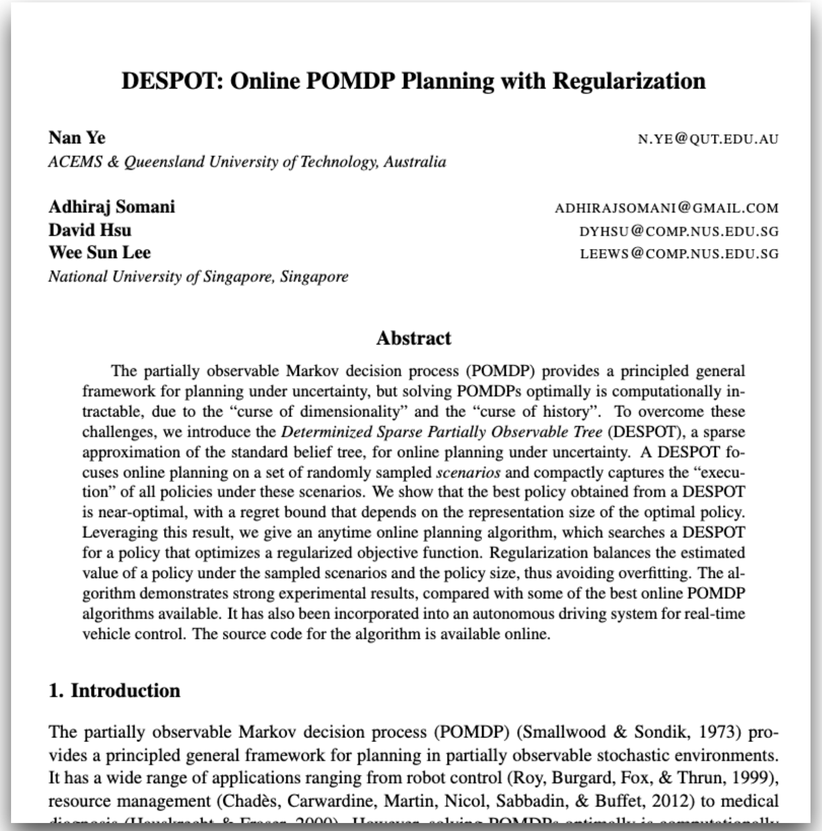
Abstract

This paper introduces a Monte-Carlo algorithm for online planning in large POMDPs. The algorithm combines a Monte-Carlo update of the agent's belief state with a Monte-Carlo tree search from the current belief state. The new algorithm, *POMCP*, has two important properties. First, Monte-Carlo sampling is used to break the curse of dimensionality both during belief state updates and during planning. Second, only a black box simulator of the POMDP is required, rather than explicit probability distributions. These properties enable POMCP to plan effectively in significantly larger POMDPs than has previously been possible. We demonstrate its effectiveness in three large POMDPs. We scale up a well-known benchmark problem, *rocksample*, by several orders of magnitude. We also introduce two challenging new POMDPs: 10×10 *battleship* and *partially observable PacMan*, with approximately 10^{18} and 10^{56} states respectively. Our Monte-Carlo planning algorithm achieved a high level of performance with no prior knowledge, and was also able to exploit simple domain knowledge to achieve better results with less search. POMCP is the first general purpose planner to achieve high performance in such large and unfactored POMDPs.

1 Introduction

Monte-Carlo tree search (MCTS) is a new approach to online planning that has provided exceptional performance in large, fully observable domains. It has outperformed previous planning approaches in challenging games such as Go [5], Amazons [10] and General Game Playing [4]. The key idea is to evaluate each state in a search tree by the average outcome of simulations from that state. MCTS provides several major advantages over traditional search methods. It is a highly selective, best-first search that quickly focuses on the most promising regions of the search space. It breaks the curse of dimensionality by sampling

Key idea: Monte Carlo Tree Search over action-observation histories, combined with particle-based belief updates from a generative model, enabling scalable online planning in large POMDPs.



DESPOT: Online POMDP Planning with Regularization

Nan Ye N.YE@QUT.EDU.AU
ACEMS & Queensland University of Technology, Australia

Adhiraj Somani ADHIRAJSOMANI@GMAIL.COM
David Hsu DYHSU@COMP.NUS.EDU.SG
Wee Sun Lee LEEWS@COMP.NUS.EDU.SG
National University of Singapore, Singapore

Abstract

The partially observable Markov decision process (POMDP) provides a principled general framework for planning under uncertainty, but solving POMDPs optimally is computationally intractable, due to the "curse of dimensionality" and the "curse of history". To overcome these challenges, we introduce the *Determinized Sparse Partially Observable Tree* (DESPOT), a sparse approximation of the standard belief tree, for online planning under uncertainty. A DESPOT focuses online planning on a set of randomly sampled *scenarios* and compactly captures the "execution" of all policies under these scenarios. We show that the best policy obtained from a DESPOT is near-optimal, with a regret bound that depends on the representation size of the optimal policy. Leveraging this result, we give an anytime online planning algorithm, which searches a DESPOT for a policy that optimizes a regularized objective function. Regularization balances the estimated value of a policy under the sampled scenarios and the policy size, thus avoiding overfitting. The algorithm demonstrates strong experimental results, compared with some of the best online POMDP algorithms available. It has also been incorporated into an autonomous driving system for real-time vehicle control. The source code for the algorithm is available online.

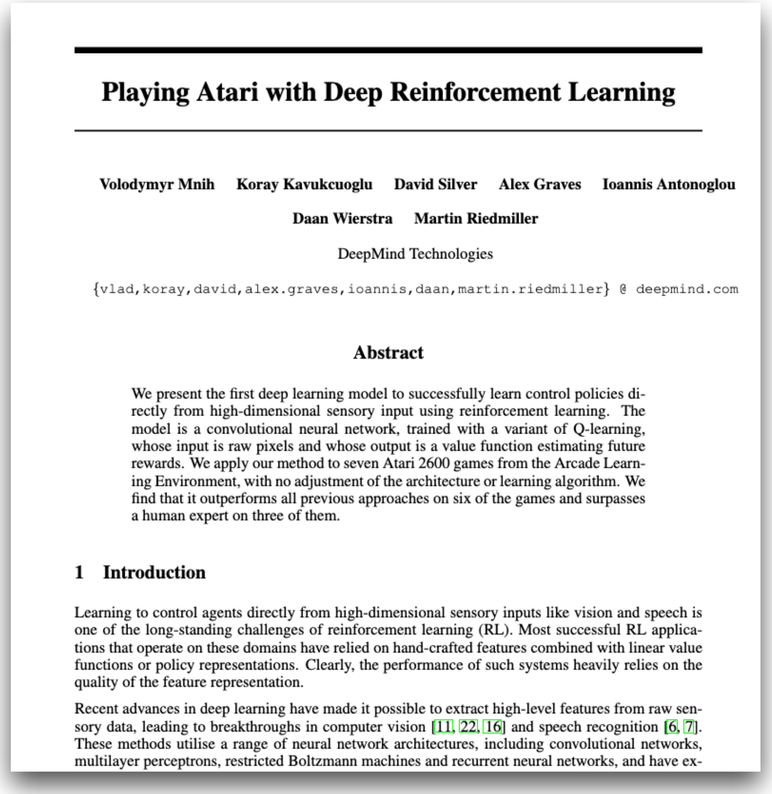
1. Introduction

The partially observable Markov decision process (POMDP) (Smallwood & Sondik, 1973) provides a principled general framework for planning in partially observable stochastic environments. It has a wide range of applications ranging from robot control (Roy, Burgard, Fox, & Thrun, 1999), resource management (Chadès, Carwardine, Martin, Nicol, Sabbadin, & Buffet, 2012) to medical diagnosis (Hauskrecht & Fraser, 2000). However, solving POMDPs optimally is computationally

Key idea: Determinized Sparse Partially Observable Tree, a sparse approximation method for online POMDP planning that samples scenarios to avoid the curse of dimensionality and history.

Hint! Cool class project.

Deep RL algorithms as implicit solutions to POMDPs



Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**
Daan Wierstra **Martin Riedmiller**
DeepMind Technologies
{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

Abstract

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

1 Introduction

Learning to control agents directly from high-dimensional sensory inputs like vision and speech is one of the long-standing challenges of reinforcement learning (RL). Most successful RL applications that operate on these domains have relied on hand-crafted features combined with linear value functions or policy representations. Clearly, the performance of such systems heavily relies on the quality of the feature representation.

Recent advances in deep learning have made it possible to extract high-level features from raw sensory data, leading to breakthroughs in computer vision [11, 22, 16] and speech recognition [6, 7]. These methods utilise a range of neural network architectures, including convolutional networks, multilayer perceptrons, restricted Boltzmann machines and recurrent neural networks, and have ex-

The goal of this paper was not focused on solving POMDPs, but was the first to demonstrate that deep neural networks trained on fixed-length histories of observations could implicitly learn performant policies and value functions in partially observable environments.

Many modern deep RL algorithms are applied to POMDP problems, and address the issue of partial observability by relying on the neural network (MLP, CNN, RNN, etc.) to implicitly learn to extract information on a latent belief state from observation histories.