Optimally Solving Dec-POMDPs as Continuous-State MDPs

Jilles Dibangoye [1], **Chris Amato** [2], Olivier Buffet [1] and François Charpillet [1]

[1] Inria, Université de Lorraine — France

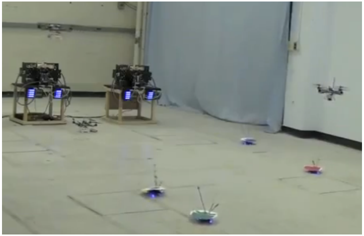
[2] MIT, CSAIL — USA

IJCAI — August 8, 2013

Outline

General overview

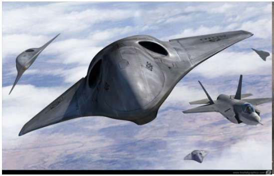
- Agents situated in a world, receiving information and choosing actions
    - Uncertainty about outcomes and sensors
    - Sequential domains
    - Cooperative multi-agent
    - Decision-theoretic approach
- Developing approaches that scale to real-world domains

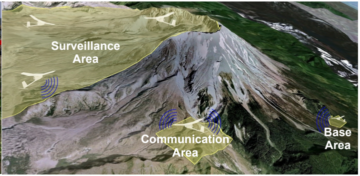


Cooperative multiagent problems

- Each agent's choice affects all others, but must be made using only local information
- Communication may be costly, slow or noisy

Domains of interest — robotics, disaster response, networks, . . .



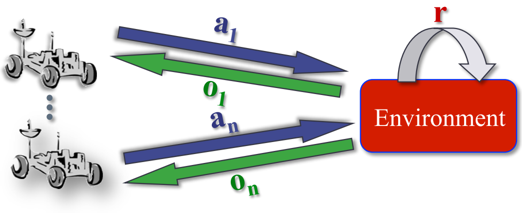


Massachusetts Institute of Technology

Multi-Agent Decision Making Under Uncertainty

Decentralized partially observable Markov decision process (Dec-POMDP)

- Sequential decision-making
  - At each stage, each agent takes an action and receives:
    - A local observation
    - A joint immediate reward



Multi-Agent Decision Making Under Uncertainty

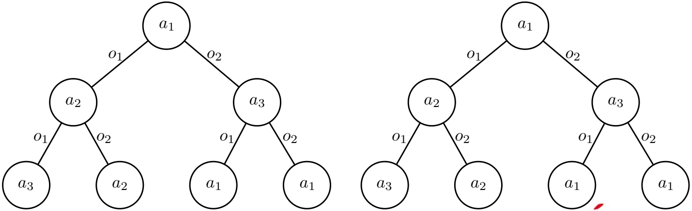
Dec-POMDP definition

Dec-POMDP — $\langle I, S, \{A_i\}, \{Z_i\}, p, r, o, b_0, T \rangle$

- I, a finite set of agents
- S, a finite set of states
- A_i, each agent's finite set of actions
- Z_i, each agent's finite set of observations
- p, the state transition model: $\Pr(s'|s, \vec{a})$
- o, the observation model: $\Pr(\vec{o}|s', \vec{a})$
- r, the reward model: $R(s, \vec{a})$
- b_0, initial state distribution
- T, planning horizon

Dec-POMDP solutions

- History $\theta_i^t = \langle a_i^0, o_i^1, \ldots, a_i^{t-1}, o_i^t \rangle$
- **Local policy**: each agent maps histories to actions, $\pi_i : \Theta_i \rightarrow A_i$
  - State is unknown, so beneficial to remember history
- π_i, a sequence of **decision rules** $\pi_i = \pi_i^0, \ldots, \pi_i^{T-1}$ mapping histories to actions, $\pi_i^t(\theta_i^t) = a_i$
- **Joint policy** $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ with individual (local) agent policies π_i
- Goal is to maximize expected cumulative reward over a finite horizon

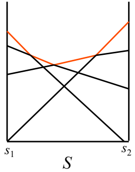


POMDPs

- Subclass of Dec-POMDPs with only one agent
- Agent maintain's **belief** state (distributions over states)
- Policy = mapping from histories or belief states

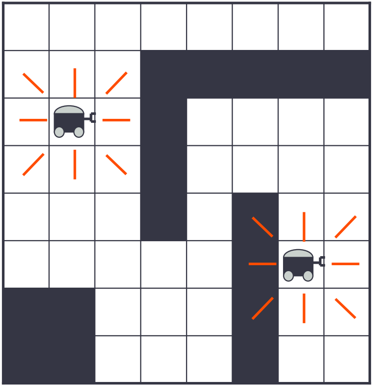
$$\pi : B \rightarrow A$$

- Can solve a POMDP as a continuous-state "belief" MDP
- $V^{\pi}(b) = R(b,a) + \sum_{o} \Pr(b'|b,a,o)\Pr(o|b',a)V^{\pi}(b')$
- Structure: piecewise linear convex (PWLC) value function



Example: 2-Agent Navigation

Meeting in a grid



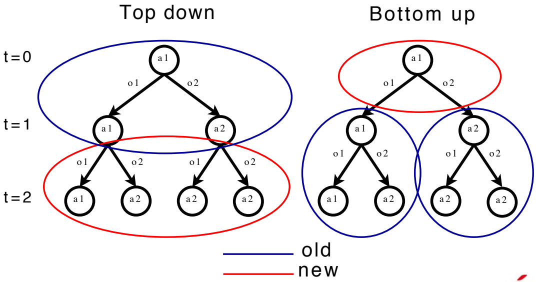
- **States:** grid cell pairs
- **Actions:** move $\uparrow$, $\downarrow$, $\leftarrow$, $\rightarrow$, stay
- **Transitions:** noisy
- **Observations:** red lines
- **Rewards:** negative unless sharing the same square

Challenges in solving Dec-POMDPs

- Partial observability makes the problem difficult to solve
- No common state estimate (centralized belief state) or concise sufficient statistic
  - Each agent depends on the others
  - Can't directly transform Dec-POMDPs into a continuous-state MDP from a single agent's perspective
- Therefore, Dec-POMDPs are fundamentally different and more complex (NEXP instead of PSPACE)

Current methods

- Assume an offline planning phase that is centralized
- Generate explicit policy representations (trees) for each agent
- Search bottom up (DP) or top down (heuristic search)
- Often use game-theoretic ideas from the perspective of a single agent
- Search in the space of policies for the optimal set



Overview of our approach

Current methods don't take full advantage of **centralized planning phase**

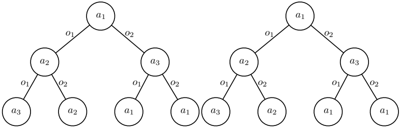
Overview

- Push common information into an **occupancy state**
- Move local information into action selection as **decision rules**
- Formalize Dec-POMDPs as **continuous-state MDPs** with a **PWLC** value function
- Exploit **multiagent structure** in representation, making it scalable

This **doesn't use explicit policy representations** or construct policies from a single agent's perspective

Centralized Sufficient Statistic

- Policy π, sequence of decentralized decision rules, $\pi = \langle \pi^0, \ldots, \pi^{T-1} \rangle$
- Joint history $\theta^t = \langle \theta_1^t, \ldots, \theta_n^t \rangle$, with $\pi^t(\theta^t) = \langle a_1, \ldots, a_n \rangle$



- An **occupancy state** is a distribution $\eta(s, \theta^t) = \Pr(s, \theta^t | \pi^{0:t-1}, b_0)$
- The occupancy state is a **sufficient statistic**:
  Can optimize *future* policy $\pi^{t:T}$ over η
  rather than initial belief and past joint policies

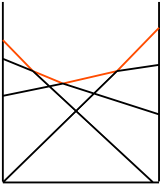
Dec-POMDPs as continuous-state MDPs

- Occupancy state $\eta^t(s, \theta^t) = \Pr(s, \theta^t | \pi^{0:t-1}, \eta_0)$ with $\eta_0 = b_0$
- Transform Dec-POMDP into a continuous-state MDP
  - $s_{MDP} : \eta$
  - $a_{MDP} : \pi^t$ (decentralized decision rules)
  - $T_{MDP} : \Pr(\eta^t | \pi^{t-1}, \eta^{t-1})$ — Deterministic with $\mathbf{P}(\eta^t, \pi^t) = \eta^{t+1}$
  - $R_{MDP} : \sum_{s,\theta^t} \eta^t(s, \theta^t) R(s, \pi^t(\theta^t))$
- Centralized sufficient statistic (the occupancy state)
- **Decision rules ensure decentralization**

Piecewise linear convexity

- Bellman optimality operator:

$$V_t^*(\eta^t) = \max_{\pi^t \in D} R_{MDP}(\eta^t, \pi^t) + V_{t+1}^*(\mathbf{P}(\eta^t, \pi^t))$$



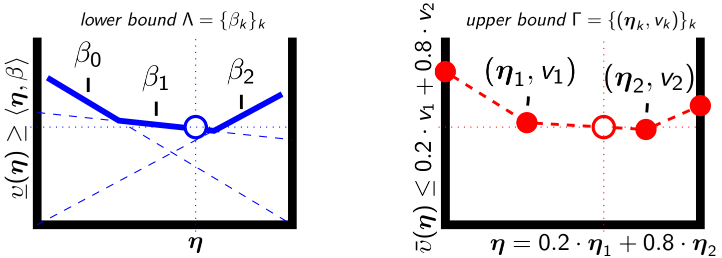
- 1- Operator preserves PWLC property
  (piecewise linearity and convexity)
  2- $R_{MDP}(\eta^t, \pi^t)$ is linear
  $\Rightarrow$ **PWLC value function**

- POMDP algorithms can be used!

Solving the occupancy MDP

Feature-based heuristic search value iteration (FB-HSVI)

- Based on heuristic search value iteration (Smith and Simmons, UAI 04)
- Sample occupancy distributions starting from the initial occupancy
- Update upper bounds based on decision rules (on the way down)
- Update lower bounds (on the way back up)
- Stop when bounds converge for initial occupancy



Scaling up

The occupancy MDP has very large action and state spaces

Two key ideas to deal with these combinatorial explosions:

1. State reduction through history compression
   - Compress histories of the same length (Oliehoek et al., JAIR 13)
   - Reduce history length without loss
2. More efficient action selection
   - Generating a greedy decision rule for an occupancy state as a weighted constraint satisfaction problem

Experiments

Tested 3 versions of our algorithm

Algorithm 0: HSVI with occupancy MDP

Algorithm 1: HSVI with efficient action selection

Algorithm 2: HSVI with efficient action selection
+ feature-based state space

Comparison algorithms

Forward search: GMAA*-ICE (Spaan et al., IJCAI 2011)

Dynamic programming: IPG (Amato et al., ICAPS 2009),
LPC (Boularias and Chaib-draa, ICAPS 2008)

Optimization: MILP (Aras and Dutech, JAIR 2010)

Experiments

Optimal υ within $\varepsilon = 0.01$

T	MILP	LPC	IPG	ICE	FB-HSVI(ρ) 0	FB-HSVI(ρ) 1	FB-HSVI(ρ) 2	$\upsilon_\epsilon(\boldsymbol{\eta}^0)$
The multi-agent tiger problem ($\|S\| = 2, \|Z\| = 4, \|A\| = 9, K = 3$)								
2	–	0.17	0.32	0.01	0.05	0.03	**0.03**	−4.00
3	4.9	1.79	55.4	0.01	2.17	0.06	**0.40**	5.1908
4	72	534	2286	108	9164	2.66	**1.36**	4.8027
5				347		22.2	**9.65**	7.0264
6						171.3	**24.42**	10.381
7							**33.11**	**9.9935**
8							**41.21**	**12.217**
9							**58.51**	**15.572**
10							**65.57**	**15.184**
The recycling-robot problem ($\|S\| = 4, \|Z\| = 4, \|A\| = 9, K = 1$)								
2	–	–	0.30	36	0.03	0.02	**0.01**	7.000
3	–	–	1.07	36	0.05	0.47	**0.10**	10.660
4	–	–	42.0	72	0.85	0.65	**0.30**	13.380
5	–	–	1812	72	1.52	0.87	**0.34**	16.486
10					5.06	2.83	**0.52**	31.863
30					62.8	37.9	**1.13**	93.402
70						78.1	**2.13**	216.47
100						259	**2.93**	**308.78**
The mars-rovers problem ($\|S\| = 256, \|Z\| = 81, \|A\| = 36, K = 3$)								
2	–	–	83	1.0	0.21	0.09	**0.10**	5.80
3	–	–	389	1.0	2.84	0.21	**0.23**	9.38
4				103	104.2	1.73	**0.47**	10.18
5						6.38	**0.82**	**13.26**
6						8.16	**3.97**	**18.62**
7						11.13	**5.81**	**20.90**
8						35.49	**22.8**	**22.47**
9						57.47	**26.5**	**24.31**
10						316.2	**62.7**	**26.31**

- Time and value on benchmarks
- Blank space = algorithm over time (200s)
- Red for fastest and previously unsolvable horizons
- K is the largest history window used

Conclusion

Summary

- Dec-POMDPs are powerful multiagent models
- Formulated Dec-POMDPs as continuous-state MDPs with PWLC value function
- POMDP (and continuous MDP) methods can now be applied
- Can also take advantage of multiagent structure in the problem
- Our approach shows significantly improved scalability

Future work

- Approximate solutions (bounds on the solution quality)
- More concise statistics
  - Subclasses like TI Dec-MDPs in our AAMAS-13 paper
  - Just observation histories as in Oliehoek, IJCAI 13