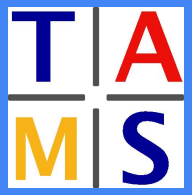
Decision making in Multiagent settings DEC-POMDP



December 7

Mohammad Ali Asgharpour Setyani

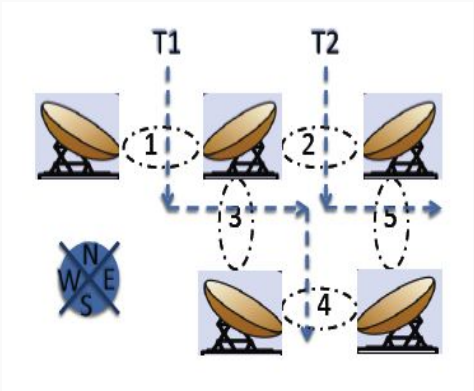
Outline

- Motivation
- Models
- Algorithms
- Starcraft Project
- Conclusion

Motivation

Application Domains :

- Multi-robot coordination
  - Space exploration rovers (Zilberstein et al., 2002)
  - Helicopter flights (Pynadath and Tambe, 02)
  - Navigation (Emery-Montemerlo et al., 05; Spaan and Melo 08)
- Sensor networks (e.g. target tracking from multiple viewpoints) (Nair et al., 05, Kumar and Zilberstein 09-AAMAS)
- Multi-access broadcast channels (Ooi and Wornell, 1996)

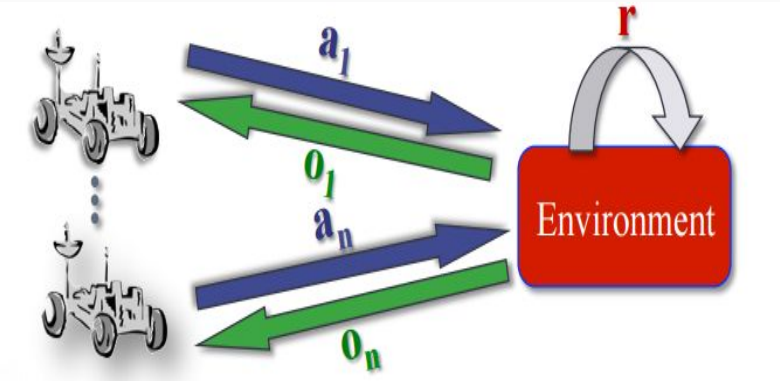


In all these problem multiple decision makers jointly control a process, but cannot share all of their information in every time step.

Models

Decentralized POMDPs

- Decentralized partially observable Markov decision process (**DecPOMDP**) also called multiagent team decision problem (**MTDP**).
- Extension of the single agent POMDP
- Decentralized process with two agents
  - At each stage, each agent takes an action and receives:
    - A local observation
    - A joint immediate reward
- each agent receives an individual observation, but the reward generated by the environment is the same for all agents.



Schematic view of a decentralized process with 2 agents, a global reward function and private observation functions (courtesy of Christopher Amato)

The DEC-POMDP model

A Dec-POMDP can be defined with the tuple: $\langle I, S, \{A_i\}, P, \{\Omega_i\}, O, R, h \rangle$ where

- I *is a finite set of agents indexed* $1, \ldots, n$.
- S *is a finite set of states, with distinguished initial state* s_0.
- A_i *is a finite set of actions available to agent* i, *and* $\vec{A} = \otimes_{i \in I} A_i$ *is the set of joint actions.*
- $P : S \times \vec{A} \to \Delta S$ *is a Markovian transition function.* $P(s'|s, \vec{a})$ *denotes the probability that after taking joint action* $\vec{a}$ *in state* s *a transition to state* s' *occurs.*
- Ω_i *is a finite set of observations available to agent* i, *and* $\vec{\Omega} = \otimes_{i \in I} \Omega_i$ *is the set of joint observations.*
- $O : \vec{A} \times S \to \Delta\vec{\Omega}$ *is an observation function.* $O(\vec{o}|\vec{a}, s')$ *denotes the probability of observing joint observation* $\vec{o}$ *given that joint action* $\vec{a}$ *was taken and led to state* s'.
- $R : \vec{A} \times S \to \mathbb{R}$ *is a reward function.* $R(\vec{a}, s')$ *denotes the reward obtained after joint action* $\vec{a}$ *was taken and a state transition to* s' *occurred.*
- *If the DEC-POMDP has a finite horizon, that horizon is represented by a positive integer* h.

Dec-POMDP solutions

- A **local policy** for each agent is a mapping from its observation sequences to actions, $\Omega^* \longrightarrow A$
  - State is unknown, so beneficial to remember history
- A **joint policy** is a local policy for each agent

- Goal is to maximize expected cumulative reward over a finite or infinite horizon
  - For infinite-horizon cannot remember the full observation history

$$V^{\pi}(s_0) = E\left[\sum_{t=0}^{h-1} R(\vec{a}_t, s_t) | s_0, \pi\right]$$

  - In infinite case, a discount factor, γ, is used

$$V^{\pi}(s_0) = E\left[\sum_{t=0}^{\infty} \gamma^t R(\vec{a}_t, s_t) | s_0, \pi\right]$$

Multi access broadcast channels

Multi-access broadcast channels (Ooi and Wornell, 1996)

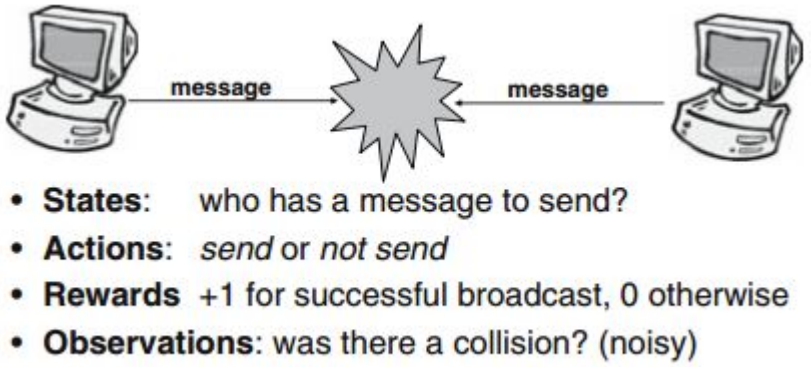
Two agents are controlling a message channel on which only one message per time step can be sent, otherwise a collision occurs.

The agents have the same goal of maximizing the global throughput of the channel.

Every time step the agents have to decide whether to send a message or not.

At the end of a time step, every agent observes information about its own message buffer about a possible collision and about a possible successful message broadcast.

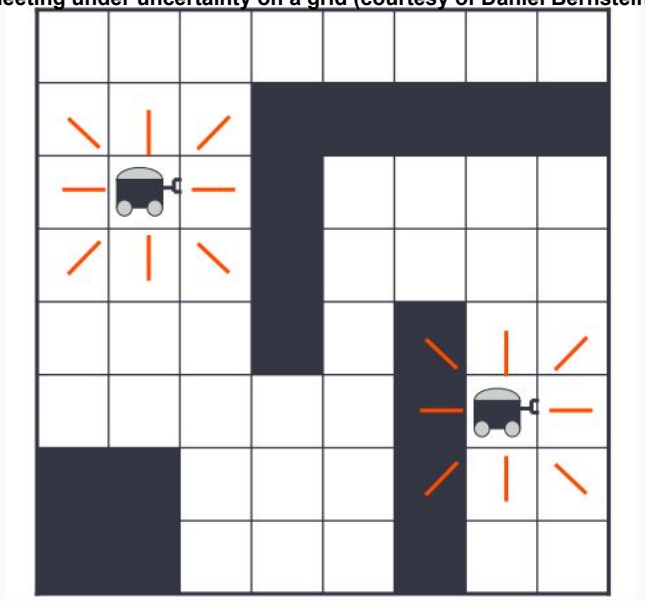
The challenge of this problem is that the observations of possible collisions are noisy and thus the agents can only build up potentially uncertain beliefs about the outcome of their actions.



Multi-access broadcast channel
(courtesy of Daniel Bernstein)

2 agent grid world Navigation

Meeting under uncertainty on a grid (courtesy of Daniel Bernstein)



States : Grid cell pairs
Action : move Up, Down, Left, Right
Transitions: noisy
Observations: red lines
Rewards: negative unless sharing the same square

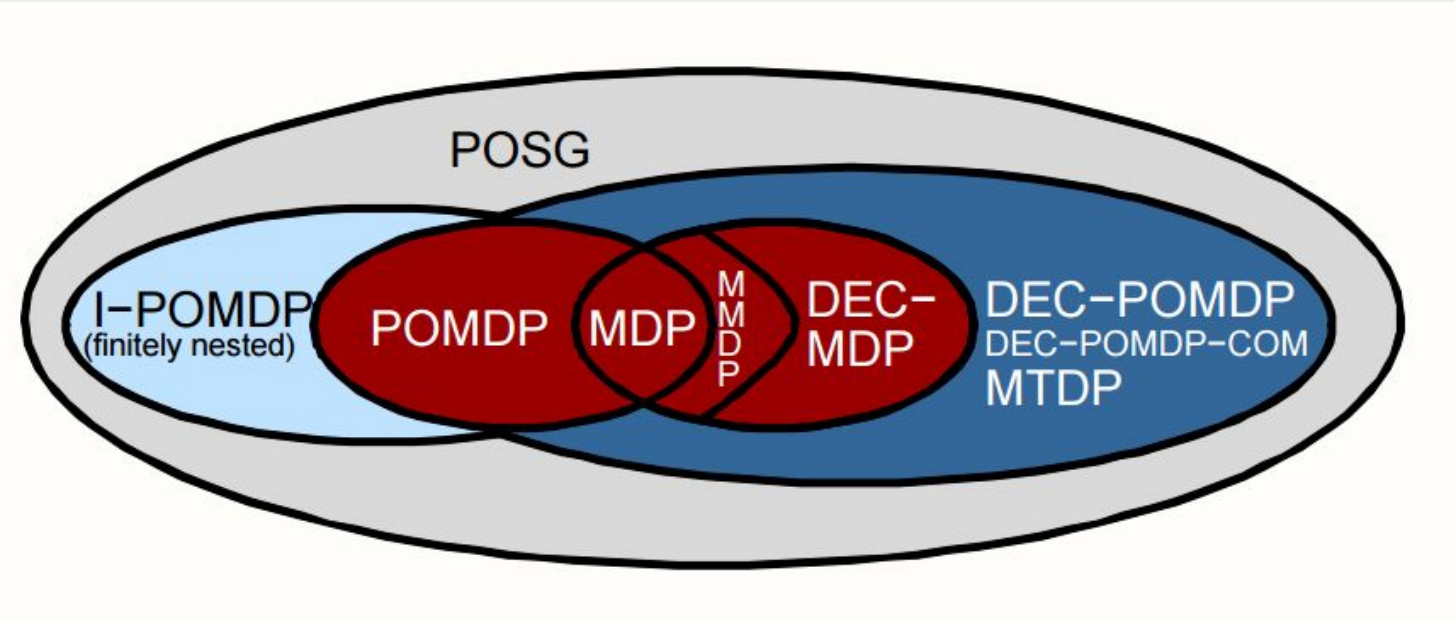
Two agents have to meet as soon as possible on a 2D grid where obstacles are blocking some parts of the environment.

Every time step the agents make a noisy transition, that is, with some probability Pi, agent i arrives at the desired location and with probability 1 − Pi the agent remains at the same location.

Due to the uncertain transitions of the agents, the optimal solution is not easy to compute, as every agent's strategy can only depend on some belief about the other agent's location.

An optimal solution for this problem is a sequence of moves for each agent such that the expected time taken to meet is as low as possible. (Goldman and Zilberstein, 2004)

Relationships among the models

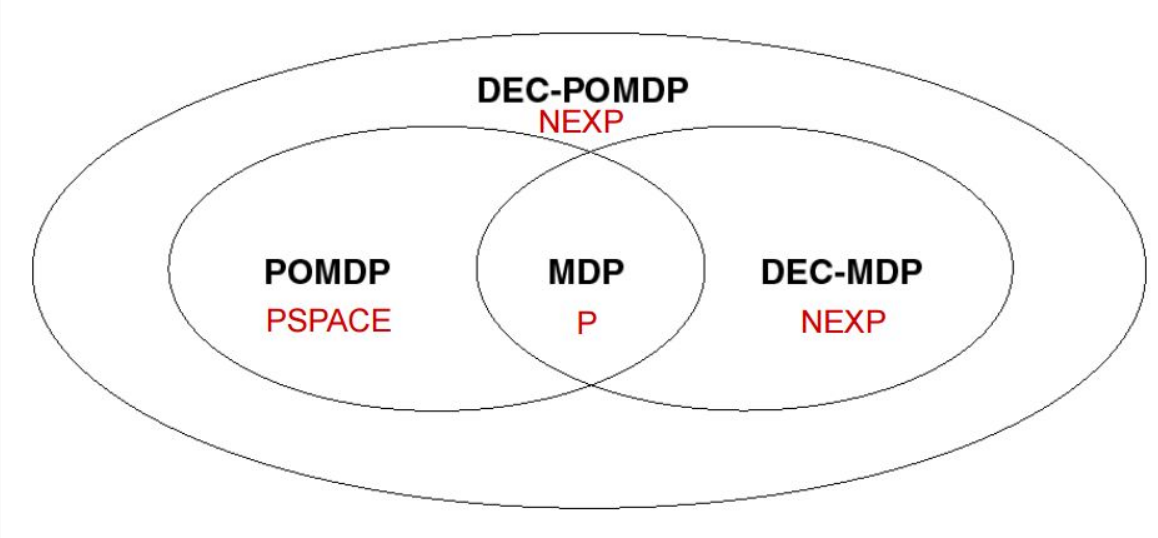


(Goldman and Zilberstein, 2004)

Challenges in solving Dec-POMDPs

- Agents must consider the choices of all others in addition to the state and action uncertainty present in POMDPs.
- This makes DEC-POMDPs much harder to solve (NEXP-complete).
- No common state estimate (centralized belief state)
  - Each agent depends on the others
  - This requires a belief over the possible policies of the other agents
  - Can’t transform Dec-POMDPs into a continuous state MDP (how POMDPs are typically solved)

General complexity results



(Goldman and Zilberstein, 2004)

Algorithms

Algorithms

- How do we produce a solution for these models?
  - **Joint Equilibrium Search for Policies (JESP)**
  - **Multiagent A***
  - **Summary and comparison of algorithms**

Joint Equilibrium Search for Policies (JESP)

(Nair et al., 03)

Instead of exhaustive search, find best response

Algorithm: JESP (Nair et al., 2003)

Start with (full) policy for each agent
while not converged do
for i=1 to n
Fix other agent policies
Find a best response policy for agent i

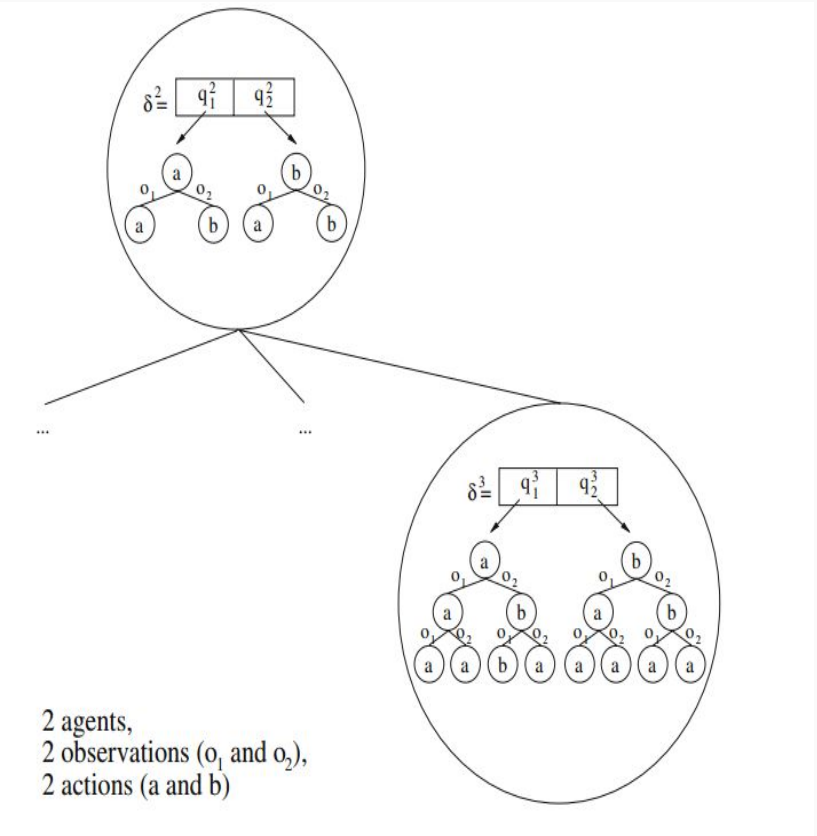
JESP summary (Nair et al., 03)

- Find a locally **optimal** set of policies
- Worst case **complexity** is the same as **exhaustive search**, but in practice is much faster
- Can also incorporate **dynamic programming** to speed up finding best **responses**
  - Fix policies of other agents
  - Generate reachable belief states from initial state
  - Build up policies from last step to first
  - At each step, choose subtrees that maximize value at reachable belief states

Multiagent A* : A heuristic search algorithm for DEC-POMDPs (Szer et al., 05)

MAA* : Top-down heuristic policy search

- The algorithm is based on the widely used A∗ algorithm
- performs best-first search in the space of possible joint policies.
- can build up policies for agents from the first step
- Use heuristic search over joint policies
- Like brute force search, searches in the space of policy vectors
- But not all nodes at every level are fully expanded. Instead, a **heuristic function** is used to evaluate the leaf nodes of the search tree.



A section of the multi-agent A∗ search tree, showing a horizon 2 policy vector with one of its expanded horizon 3 child nodes (courtesy of Daniel Szer)

Multiagent A*

- Requires an admissible heuristic function.
- A* -like search over partially specified joint policies:

$$\varphi^t = (\delta^0, \delta^1, \ldots, \delta^{t-1}),$$
$$\delta^t = (\delta_0^t, \ldots, \delta_n^t) \qquad \delta_i^t : \vec{\mathcal{O}}_i^t \to A_i$$

Heuristic value for φ^t:

$$\underbrace{\widehat{V}(\varphi^t)}_{F} = \underbrace{V^{0\ldots t-1}(\varphi^t)}_{G} + \underbrace{\widehat{V}^{t\ldots h-1}}_{H}$$

if $\widehat{V}^{t\ldots h-1}$ is admissible (overestimation), so is $\widehat{V}(\varphi^t)$.

summary and comparison of algorithms

Algorithm :	Joint equilibrium-based search for policies
Authors :	Nair et al
Solution quality :	Approximate
Solution technique :	Computation of reachable belief states, dynamic programming,improving policy of one agent while holding the others fixed
Advantages :	Avoids unnecessary computation by only considering reachable belief states
Disadvantages :	Suboptimal solutions: finds only local optima

summary and comparison of algorithms

Algorithm :	Multi-agent A*: heuristic search for DEC-POMDPs
Authors :	Szer et al
Solution quality :	Optimal
Solution technique :	Top-down A∗-search in the space of joint policies
Advantages :	Can exploit an initial state, could use domain-specific knowledge for the heuristic function (when available)
Disadvantages :	Cannot exploit specific DEC-POMDP structure, can at best solve problems with horizon 1 greater than problems that can be solved via brute force search (independent of heuristic)

StarCraft Project

Starcraft

- Starcraft released in 1998, is a military sci-fi real time strategy video game developed by blizzard entertainment.
- selling over 9.5 million copies worldwide. starcraft has become a very successful game for a over a decade and continues to receive support from blizzard.
- As a result of this popularity the modding community has spent countless hour reverse-engineering the starcraft code producing the BroodWar API (BWAPI)
- Student Starcraft AI competition : enabling modders to develop Custom AI bots for the game

Approach :

The MDAP toolbox is a free c++ software toolbox.

The toolbox includes most algorithms like JESP and MAA* which will be used to test.

Evaluation and Demonstration

We can measure how well an algorithm performs base on two criterion:

1. How many members from the team survived
2. How many enemy combatants were eliminated

Gameplay presentation

Conclusion

What problems Dec-POMDPs are good for :

- Sequential (not “one shot” or greedy)
- Cooperative (not single agent or competitive)
- Decentralized (not centralized execution or free, instantaneous communication)
- Decision-theoretic (probabilities and values)

Resources

- Dec-POMDP webpage
  - Papers, talks, domains, code, results
  - http://rbr.cs.umass.edu/~camato/decpomdp/
- Matthijs Spaan's Dec-POMDP page
  - Domains, code, results
  - http://users.isr.ist.utl.pt/~mtjspaan/decpomdp/index_en.html
- USC's Distributed POMDP page
  - Papers, some code and datasets
  - http://teamcore.usc.edu/projects/dpomdp/

References

Incremental Policy Generation for Finite-Horizon DEC-POMDPs. Christopher Amato, Jilles Steeve Dibangoye and Shlomo Zilberstein. Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS-09), Thessaloniki, Greece, September, 2009.
Policy Iteration for Decentralized Control of Markov Decision Processes. Daniel S. Bernstein, Christopher Amato, Eric A. Hansen and Shlomo Zilberstein. Journal of AI Research (JAIR), vol. 34, pages 89-132, February, 2009.
Multiagent planning under uncertainty with stochastic communication delays. Matthijs T. J. Spaan, Frans A. Oliehoek, and Nikos Vlassis. In Int. Conf. on Automated Planning and Scheduling, pages 338–345, 2008.
Formal Models and Algorithms for Decentralized Decision Making Under Uncertainty. Sven Seuken and Shlomo Zilberstein. Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS). 17:2, pages 190-250, 2008.
MAA*: A Heuristic Search Algorithm for Solving Decentralized POMDPS. Daniel Szer, Francois Charpillet, Shlomo Zilberstein. Proceedings of the 21st International Conference on Uncertainty in Artificial Intelligence (UAI-05), Edinburgh, Scotland, July 2005
A framework for sequential planning in multi-agent settings. Piotr J. Gmytrasiewicz and Prashant Doshi. Journal of Artificial Intelligence Research, 24:49–79, 2005.
Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis. Claudia V. Goldman and Shlomo Zilberstein. Journal of Artificial Intelligence Research Volume 22, pages 143-174, 2004.
Decentralized control of a multiple access broadcast channel: Performance bounds. James M. Ooi and Gregory W. Wornell. In Proceedings of the 35th Conference on Decision and Control, 1996.
Formal models and algorithms for decentralized decision making under uncertainty Sven Seuken · Shlomo Zilberstein

Towards Realistic Decentralized Modelling for Use in a Real-World Personal Assistant Agent Scenario. Christopher Amato, Nathan Schurr and Paul Picciano. In Proceedings of the Workshop on Optimisation in Multi-Agent Systems (OptMAS) at the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-11), Taipei, Taiwan, May 2011.