Decision Making in Multiagent Settings:
Team Decision Making

Matthijs Spaan    Christopher Amato    Shlomo Zilberstein

AAMAS10 Tutorial, May 10, 2010

Part Z: Beyond Two Agents
Beyond Two Agents

**Challenge:** How to scale up decision-theoretic planning to multi-agent systems with *many* agents?

- Which existing algorithms scale up?
- What are the main difficulties?
Key Problem Characteristics

- A **group** of agents interact in a stochastic environment with **no central controller** of their operation.
- Agents try to achieve some **joint objectives** in a **cooperative setting**.
- Each planning problem involves a **sequence** of decisions over **finite or infinite horizon**.
- The environment changes **stochastically** based on the **current state** and the **set of actions** taken by the agents.
- Each agent has **different partial knowledge** about the global state.
Decentralized Partially Observable MDP

- A framework for multiagent sequential decision making
- Multiple agents control the Markov process → interaction
- Each agent has a local view → decentralization
- Joint reward → cooperation
- Goal: a policy optimizing expected reward
A DEC-POMDP is defined by a tuple:

\[ M = \langle S, \vec{A}, P, R, \vec{\Omega}, O \rangle \]

- \( S \), a finite set of states with initial state distribution \( b_0 \)
- \( \vec{A} = A_1 \times \ldots \times A_n \), each agent’s finite set of actions
- \( P \), the state transition model: \( P(s'|s, \vec{a}) \)
- \( R \), the reward model: \( R(s, \vec{a}) \)
- \( \vec{\Omega} = \Omega_1 \times \ldots \times \Omega_n \), each agent’s finite set of observations
- \( O \), the observation model: \( O(\vec{o}|s', \vec{a}) \)

Is it feasible to describe an arbitrary 15-agent DEC-POMDP with 4 actions and 4 observations per agent?

What about representing agent policies in this case?

What can be done about this?
Exploiting Structured Interactions

Tracking a target using a sensor network [Nair et al. ’05]

- Independent targets move stochastically between locations
- Sensors do not affect each other or target movement
- Sensors provide noisy observation
- Reward obtained when two sensors track a target jointly
- Find a joint policy optimizing reward over a finite horizon
- Corresponding DEC-POMDP is intractable.

Can we do better?
Exploiting Structured Interactions

Tracking a target using a sensor network [Nair et al. ’05]

- Independent targets move stochastically between locations
- Sensors do not affect each other or target movement
- Sensors provide noisy observation
- Reward obtained when two sensors track a target jointly
- Find a joint policy optimizing reward over a finite horizon
- Corresponding DEC-POMDP is intractable.
Can we do better?
ND-POMDP: Model Specification

Networked Distributed POMDP (ND-POMDP [Nair et al. ’05])
Can model such structured interactions succinctly

\[ S = \times_{1 \leq i \leq n} S_i \times S_u \]
\[ A = \times_{1 \leq i \leq n} A_i \]
\[ \Omega = \times_{1 \leq i \leq n} \Omega_i \]

\[ P(s' | s, a) = P_u(s'_u | s_u) \cdot \prod_{1 \leq i \leq n} P_i(s'_i | s_i, s_u, a_i), \text{ transition ind.} \]
\[ O(\omega | s, a) = \prod_{1 \leq i \leq n} O_i(\omega_i | s_i, s_u, a_i), \text{ observation ind.} \]
\[ R(s, a) = \Sigma_l R_l(s_l, s_u, a_l), \text{ reward decomposition along links} \]
ND-POMDP: Interaction Hypergraph

- Vertices are the agents
- An edge connects agents sharing a reward component
  \[ R = R_{12} + R_{13} + R_{24} + R_{34} + R_{45} \]
- Graph becomes useful in establishing the complexity and designing algorithms
ND-POMDPs: Decomposable Value Function

- **Key Observation:** The expected value of a joint policy is decomposable along the interaction links.
- This is an important advantage of transition and observation independence in ND-POMDPs.

**Theorem**

*Given transition and observation independence, and a decomposable reward function, the expected value of a joint policy \( \pi \) is decomposable along the interaction links.*

\[
V_t^{\pi}(s^t, \omega^t) = \sum_l V_t^{\pi_l}(s^t_l, s^t_u, \omega^t_l).
\]

- Originally suggested and proved by [Nair et al. ’05].
Algorithms for ND-POMDP

- Early algorithms were based on policy search:
  - GOA [Nair et al. ’05]
  - LID-JESP [Nair et al. ’05]
  - SPIDER [Pradeep et al. ’07]
  - FANS [Marecki et al. ’08]

- Scalability with planning horizon and number of agents remained very limited

- A recent point-based approach called CBDP is more promising [Kumar & Zilberstein, ’09]
- New approach provides strong theoretical guarantees with respect to runtime and scalability
Algorithms for ND-POMDP

- Early algorithms were based on policy search:
  - GOA [Nair et al. ’05]
  - LID-JESP [Nair et al. ’05]
  - SPIDER [Pradeep et al. ’07]
  - FANS [Marecki et al. ’08]

- Scalability with planning horizon and number of agents remained very limited

- A recent point-based approach called CBDP is more promising [Kumar & Zilberstein, ’09]

- New approach provides strong theoretical guarantees with respect to runtime and scalability
A Point-Based Algorithm for ND-POMDPs

Main features of the Constraint-Based Dynamic Programming (CBDP) algorithm for ND-POMDPs [Kumar & Zilberstein, ’09]

- Focuses planning on the reachable part of belief space
- Algorithm has linear time and space complexity w.r.t. problem horizon
- Complexity is also linear w.r.t the number of agents
- Complexity per horizon step is exponential only in the induced tree-width of the agent interaction hypergraph
  - Tree-width $\ll$ number of agents
  - Highly scalable to large loosely-coupled multiagent systems
**Point-Based DP Framework**

**Main idea:**
- Precompute a belief selection heuristic
- Select a small set of belief points
- Plan for the selected beliefs only

**Belief selection heuristic**
- Use underlying factored MDP of the ND-POMDP model: exponential action and state space in the number of agents

**Planning for selected beliefs**
- Search through exponential number of joint policies

Each of these steps is non-trivial for ND-POMDPs!
Point-Based DP Framework

Main idea:

- Precompute a belief selection heuristic
- Select a small set of belief points
- Plan for the selected beliefs only

Belief selection heuristic

- Use underlying factored MDP of the ND-POMDP model: exponential action and state space in the number of agents

Planning for selected beliefs

- Search through exponential number of joint policies

Each of these steps is non-trivial for ND-POMDPs!
Point-Based DP Framework

Main idea:
- Precompute a belief selection heuristic
- Select a small set of belief points
- Plan for the selected beliefs only

Belief selection heuristic
- Use underlying factored MDP of the ND-POMDP model: exponential action and state space in the number of agents

Planning for selected beliefs
- Search through exponential number of joint policies
Each of these steps is non-trivial for ND-POMDPs!
Dynamic Programming for ND-POMDPs

Key observations

- Each of these steps can be transformed into a constraint optimization problem (COP)
- Can be solved using existing efficient algorithms for COP

A COP can be described by

- Variables $X = \{X_1, \ldots, X_n\}$
- Domain for each of them
- Constraints $\{f_1, \ldots, f_k\}$ define utilities over subsets of variables
- Objective: find assignment to variables maximizing $\sum_i f_i$
MDP Heuristic for ND-POMDPs

Approximate the factored-MDP value function

\[
\hat{V}(s^{t-1}) = \max_a \sum_l (r_i^{t-1} + \sum_{s_u,s_i} p_u^{t-1} p_i^{t-1} \tilde{V}_i^t)
\]

- Can be reformulated as a COP
  - Variables \(A_i\) – one for each agent
  - Domain of \(A_i\) – action space of agent \(i\)
  - Find joint action maximizing \(\hat{V}(s^{t-1}) = \max_a \sum_l f_i\)
  - Can be solved using the bucket elimination approach in \(O(n|A_i|^d)\) time and space
  - Constraint network for the sensor network example:
Belief Sampling Using MDP Heuristic

Observation: belief over joint state is also factored

- \( b(s) = b_u(s_u)b_1(s_1) \ldots b_n(s_n) \)
- Given a state \( s^t \), select action \( a^* \) using MDP policy
- \( s^{t+1}_i = \text{Sample}(P(\cdot | s^t_i, s^t_u, a^*_i)) \)
- Transition independence helps to avoid reasoning about the exponential joint state-space

Use a portfolio of heuristics (random, observation based, ...)

Spaan, Amato, Zilberstein  AAMAS Tutorial  May 10  2010
Planning for Selected Belief Points

Compute the best joint-policy for a selected belief
Number of joint-policies is exponential $\Rightarrow$ exhaustive search infeasible

Can be reformulated as a COP

\[ V_{\pi^*}(b) = \max_{\pi} \sum_i \sum_{s_l, s_u} b_u(s_u) b_l(s_l) V_{\pi_i}(s_l, s_u, \langle \rangle) \]

- Variables $\Pi_i$ – one for each agent
- Domain of $\Pi_i$ – backed up policies of agent $i$
- Find joint policy $\pi$ maximizing $\sum_i f_i$
- Can be solved using the bucket elimination algorithm
- Runtime is linear in number of agents and exponential in induced tree-width
Planning for Selected Belief Points

Compute the best joint-policy for a selected belief
Number of joint-policies is exponential $\Rightarrow$ exhaustive search infeasible
Can be reformulated as a COP

$$V_{\pi^*}(b) = \max_{\pi} \sum_l \sum_{s_U, s_L} b_u(s_u)b_l(s_l)V_{\pi_l}(s_l, s_u, \langle \rangle)$$

- Variables $\Pi_i$ – one for each agent
- Domain of $\Pi_i$ – backed up policies of agent $i$
- Find joint policy $\pi$ maximizing $\sum_l f_l$
- Can be solved using the bucket elimination algorithm
- Runtime is linear in number of agents and exponential in induced tree-width
Experiments: 7 Agent Domain

- CBDP compared with the best existing algorithm, FANS [Marecki et al. ’08]
- Used existing benchmark problems with 5, 7, 11 and 15 agents
- “Link” and “Node” are versions of FANS
Experiments: 11 and 15 Agent Domains

FANS does not scale well with horizon for larger domains

*Horizon = 3*
Experiments: Scalability of CBDP with Horizon

CBDP scales well with horizon even for largest 15-agent domain
CBDP Summary

Exploiting domain structure can increase the scalability of DEC-POMDP algorithms
- transition and observation independence
- locality of interactions
- many practical domains exhibit such structure

CBDP is an efficient point-based approach for ND-POMDPs
- exploits the structure of ND-POMDPs using constraints
- linear time and space complexity with the horizon
- linear complexity w.r.t. the number of agents
- exponential only in the induced-width
Where do we go from here?

- Simulation and sampling based methods to replace explicit model representation
- Graphical models and message-passing algorithms for arbitrary DEC-POMDPs
- Generalizing policies from small scale networks to larger networks with same local structure
- Hierarchical methods for DEC-POMDPs (extending hierarchical factored MDPs) present many challenges
- Solving large-scale collaborative multi-agent problems using market-based methods developed for self-interested agents
Journal Publications


Main Conference Publications


