Exact Dynamic Programming for Decentralized POMDPs with Lossless Policy Compression

DAMAS Décision, Adaptation, Multi-AgentS
www.damas.ift.ulaval.ca

Abdeslam Boularias and Brahim Chaib-draa

DAMAS Laboratory,
Department of Computer Science and Software Engineering,
Laval University, Canada
{boularias;chaib}@damas.ift.ulaval.ca

UNIVERSITÉ LAVAL

1. Introduction

1. Present a new method to reduce the policy space dimensionality in DEC-POMDPs
2. Inspired from the Predictive State Representations (PSRs) [Littman et al., 2001], an approach which was proposed to reduced the state space dimensionality in POMDPs
3. Show empirical comparison of the proposed algorithm with the original Dynamic Programming algorithm

2. DEC-POMDPs [Bernstein et al., 2002]

Definition. *A Decentralized Partially Observable Markov Decision Process (DEC-POMDP) is defined by:*

- $\mathcal{I}$: *a finite set of agents*
- $\mathcal{S}$: *a finite set of states*
- $\mathcal{A}_i$: *a finite set of individual actions for each agent* $i \in \mathcal{I}$
- $\mathcal{P}$: *a stochastic transition function*
- Ω_i: *a finite set of individual observations for each agent* $i \in \mathcal{I}$
- $\mathcal{O}$: *a stochastic observation function*
- $\mathcal{R}$: *a reward function*
- $\mathcal{T}$: *a planning horizon*
- γ: *a discount factor*

3. Example

- A multi-robot navigation task, with 9 states, 5 actions (stay, move up, move down, move left, move right) and 2 observations (presence or absence of a wall around the robot)
- The goal of the two robots is to meet somewhere on the grid

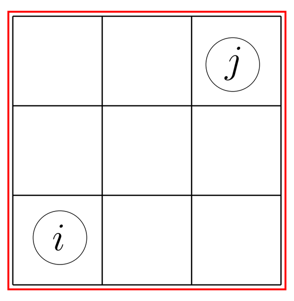


Figure 1: *Meeting On a Grid problem [Bernstein et al., 2007].*

4. Planning in DEC-POMDPs

- Planning algorithms for DEC-POMDPs aim to find the best joint policy of horizon T, which is a collection of local policies
- A policy of horizon t for agent i, denoted by q_i^t, is a mapping from local histories of observations $o_i^1 o_i^2 \dots o_i^t$ to actions in A_i
- Planning in DEC-POMDPs is centralized, while plan execution is decentralized
- During the execution of the plan, the agents cannot communicate their observations

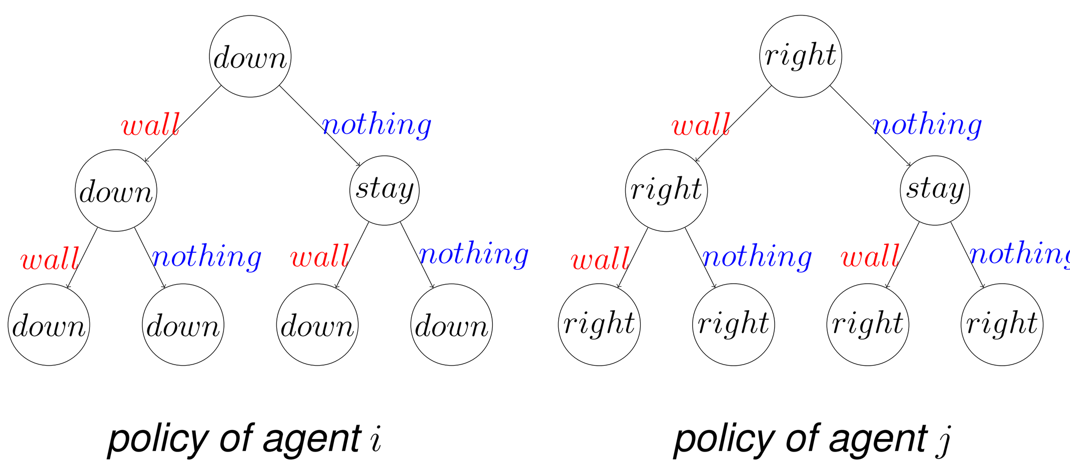


Figure 2: *An optimal joint policy of horizon 3.*

5. Multi-agent belief state

- The belief state b_i for agent i contains:
  - a probability distribution over the states S.
  - a probability distribution over the policies Q_j of agent j.
- $b_i(s, q_j)$ is the probability that the system is in state s and the current policy of agent j is q_j.

6. Dynamic Programming for DEC-POMDPs

- The expected discounted reward of a joint policy q^t, started from state s, is given recursively by *Bellman value function*:

$$V_{q^t}(s) = R(s, \vec{A}(q^t)) + \gamma \sum_{s' \in S} P(s'|s, \vec{A}(q^t)) \sum_{\vec{o} \in \vec{\Omega}} O(\vec{o}|s', \vec{A}(q^t)) V_{\vec{o}(q^t)}(s') \tag{1}$$

where $\vec{A}(q^t)$ is the first joint action of the policy q^t (the root node), $\vec{o}$ is a joint observation, and $\vec{o}(q^t)$ is the sub-policy of q^t below the root node and the observation $\vec{o}$.

- The value of an individual policy q_i^t, according to a belief state b_i, is given by the following function:

$$V_{q_i^t}(b_i) = \sum_{s \in S} \sum_{q_j^t \in Q_j^t} b_i(s, q_j^t) V_{\langle q_i^t, q_j^t \rangle}(s) \tag{2}$$

where $\langle q_i^t, q_j^t \rangle$ denotes the joint policy made up of q_i^t and q_j^t

- A policy q_i^t is said to be dominated if and only if:

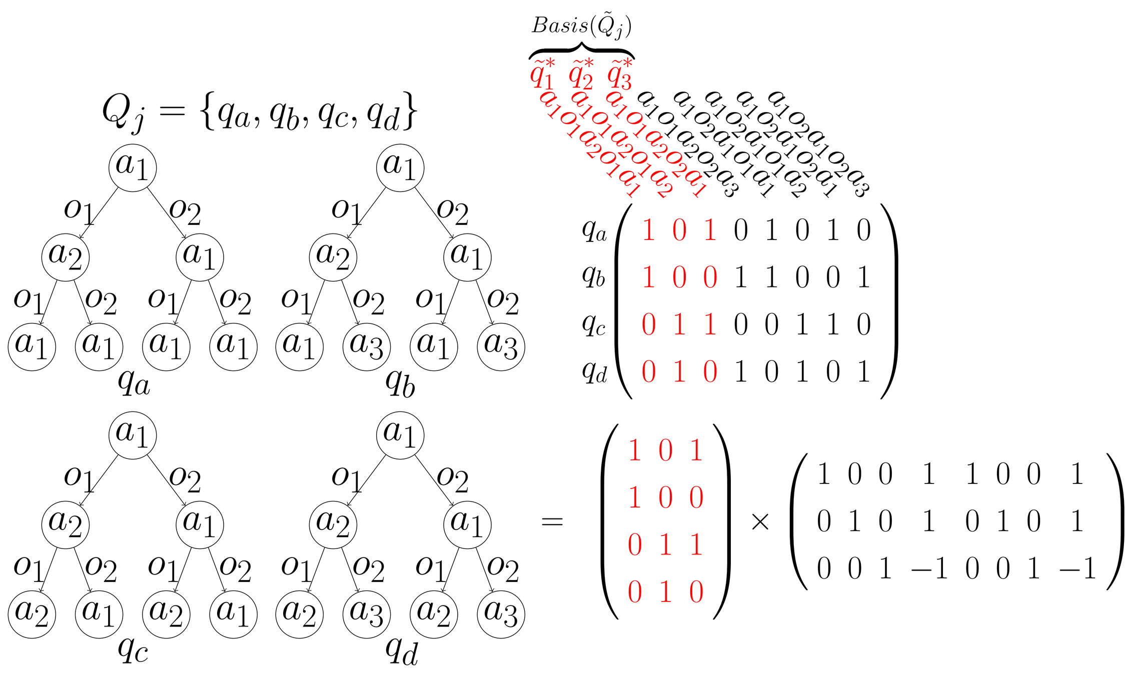
$$\forall b_i \in \Delta(S \times Q_j^t), \exists q_i^{t\prime} \in Q_i^t - \{q_i^t\}\colon V_{q_i^{t\prime}}(b_i) \geqslant V_{q_i^t}(b_i) \tag{3}$$

7. Dynamic Programming Algorithm [Hansen et al., 2004]

Require: Q_i^{t-1}, Q_j^{t-1} and V^{t-1}
$Q_i^t, Q_j^t \leftarrow$ fullBackup(Q_i^{t-1}), fullBackup(Q_j^{t-1})
Calculate the value vectors V^t by using V^{t-1}
repeat
  remove the policies of Q_i^t that are dominated
  remove the policies of Q_j^t that are dominated
until no more policies in Q_i^t or Q_j^t can be removed
Ensure: Q_i^t,Q_j^t and V^t

8. Linear reduction of the policy space dimensionality

- Given a set of policies Q_j, construct a binary matrix indicating for each policy which sequences of actions and observations are contained in this policy
- Use the linearly independent columns of this matrix as a basis for all the remaining sequences



9. Finding the basis sequences

Theorem. *Let Q_j^{t-1} be a set of horizon $t-1$ policies, $\tilde{Q}_j^{t-1}$ is the set of horizon $t-1$ sequences corresponding to Q_j^{t-1}, Q_j^t is the set of horizon t policies created from Q_j^{t-1} by an exhaustive backup, and $\tilde{Q}_j^t$ is the set of horizon t sequences corresponding to Q_j^t, then:*

$$Basis(\tilde{Q}_j^t) = \{ao\tilde{q}_j^{t-1} : a \in A_j, o \in \Omega_j, \tilde{q}_j^{t-1} \in Basis(\tilde{Q}_j^{t-1})\}$$

10. Reduced belief states and reduced value vectors

- A reduced belief state $\tilde{b}_i$ contains the probabilities of basis sequences instead of the distribution over policies
- The value function of an individual policy q_i^t in a reduced belief state $\tilde{b}_i$ is given by:

$$V_{q_i^t}(\tilde{b}_i) = \sum_{s \in S} \sum_{\substack{\tilde{q}_j^* \in Basis(\tilde{Q}_j^t) \\ \tilde{q}_i^* \in Basis(\tilde{Q}_i^t) \cap q_i^t}} \tilde{b}_i(s, \tilde{q}_j^*) \tilde{V}_{\langle \tilde{q}_i^*, \tilde{q}_j^* \rangle}(s)$$

where $\tilde{V}_{\langle \tilde{q}_i^*, \tilde{q}_j^* \rangle}$ is a reduced value vector

- The reduced value vectors are calculated by a Bellman-like equation

11. Dynamic Programming with Policy Compression

Require: Q_i^{t-1},Q_j^{t-1},$Basis(\tilde{Q}_i^{t-1})$,$Basis(\tilde{Q}_j^{t-1})$, $\tilde{V}_i^{t-1}$,$\tilde{V}_j^{t-1}$
$Q_i^t, Q_j^t \leftarrow$ fullBackup(Q_i^{t-1}), fullBackup(Q_j^{t-1})
$Basis(\tilde{Q}_i^t) \leftarrow A_i \times O_i \times Basis(\tilde{Q}_i^{t-1})$
$Basis(\tilde{Q}_j^t) \leftarrow A_j \times O_j \times Basis(\tilde{Q}_j^{t-1})$
Calculate the vectors $\tilde{V}^t$ by using $\tilde{V}^{t-1}$
repeat
  remove the policies of Q_i^t that are dominated
  remove the policies of Q_j^t that are dominated
until no more policies in Q_i^t or Q_j^t can be removed
REMOVE$(Q_i^t, Basis(\tilde{Q}_i^t), Basis(\tilde{Q}_j^t), \tilde{V}^t)$
REMOVE$(Q_j^t, Basis(\tilde{Q}_j^t), Basis(\tilde{Q}_j^t), \tilde{V}^t)$
Ensure: Q_i^t, Q_j^t, $Basis(\tilde{Q}_i^t)$, $Basis(\tilde{Q}_j^t)$, $\tilde{V}_i^t$, $\tilde{V}_j^t$

function REMOVE($Q_i^t, Basis(\tilde{Q}_i^t), Basis(\tilde{Q}_j^t), \tilde{V}^t$):
  Use a decomposition method to find the linearly dependent sequences in $Basis(\tilde{Q}_i^t)$, and remove them
  for removed sequence $\tilde{q}_i$ from $Basis(\tilde{Q}_i^t)$ **do**
    for basis sequence $\tilde{q}_i^*$ from $Basis(\tilde{Q}_i^t)$ **do**
      for basis sequence $\tilde{q}_j^*$ from $Basis(\tilde{Q}_j^t)$ **do**
        $\tilde{V}_{\langle \tilde{q}_i^*, \tilde{q}_j^* \rangle} \leftarrow \tilde{V}_{\langle \tilde{q}_i^*, \tilde{q}_j^* \rangle} + w_{\tilde{q}_i}(\tilde{q}_i^*) \tilde{V}_{\langle \tilde{q}_i, \tilde{q}_j^* \rangle}$
        // $w_{\tilde{q}_i}(\tilde{q}_i^*)$ is the weight of $\tilde{q}_i^*$ in $\tilde{q}_i$
      end for
    end for
  end for
Ensure: updated $Basis(\tilde{Q}_i^t), \tilde{V}^t$
end function

12. Experimental Results

		DP		DP with Policy Compression		
Problem	T	runtime	policies	runtime	sequences	ratio
MA-Tiger	2	0.20	(27,27)	0.17	(18,18)	1.5
	3	2.29	(675,675)	1.79	(90,90)	7.5
	4	-	-	534.90	(540,540)	361.25
MABC	2	0.12	(8,8)	0.14	(8,8)	1
	3	0.46	(72,72)	0.36	(24,24)	3
	4	17.59	(1800,1458)	4.59	(80,80)	44.1

Table 1: *The runtime (in seconds) and the number of policies and sequences of DP algorithms, with and without compression.*

13. Discussion & Further Work

✓ A new compression technique for DEC-POMDPs, based on projecting the belief points from the high dimensional space of trees to the low dimensional space of sequences, using matrix factorization methods to reduce the number of sequences

✓ A significant improvement of both memory space and runtime of Dynamic Programming algorithm

✗ An under-constrained linear program is used for pruning dominated policies, leading to more policies

Further work:

1. Use this approach in approximate DP algorithms, mainly for domains with a large observations space
2. Investigate quick and lossy factorization techniques, and more specifically, binary-matrices factorization algorithms

References

Littman, M., Sutton, R., and Singh, S. (2001). Predictive Representations of State. In *Advances in Neural Information Processing Systems 14 (NIPS'01)*, pages 1555–1561.

Bernstein, D., Immerman, N., and Zilberstein, S. (2002). The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research* 27(4):819–840.

Hansen, E., Bernstein, D., and Zilberstein, S. (2004). Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, pages 709–715.

Bernstein, D., Hansen, E., and Zilberstein, S. (2007). Bounded Policy Iteration for Decentralized POMDPs. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1287–1292.