

Q-value Heuristics for Approximate Solutions of Dec-POMDPs

Frans A. Oliehoek and Nikos Vlassis

ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{faolieho,vlassis}@science.uva.nl

Abstract

The Dec-POMDP is a model for multi-agent planning under uncertainty that has received increasingly more attention over the recent years. In this work we propose a new heuristic Q_{BG} that can be used in various algorithms for Dec-POMDPs and describe differences and similarities with Q_{MDP} and Q_{POMDP} . An experimental evaluation shows that, at the price of some computation, Q_{BG} gives a consistently tighter upper bound to the maximum value obtainable.

Introduction

In recent years the artificial intelligence (AI) community has shown an increasing interest in multi-agent systems (MAS), thereby narrowing the gap between game theoretic and decision theoretic reasoning. Especially popular are frameworks based on Markov decision processes (MDPs) (Puterman 1994). In this paper we focus on the decentralized partially observable Markov decision process (Dec-POMDP), a variant for multi-agent (decentralized) planning in stochastic environments that can only be partially observed.

Examples of application fields for Dec-POMDPs are cooperative robotics, distributed sensor networks and communication networks. Two specific examples are by Emery-Montemerlo *et al.* (2004), who considered multi-robot navigation in which a team of agents with noisy sensors has to act to find/capture a goal, and Becker *et al.* (2004), who introduced a multi-robot space exploration example in which the agents (mars rovers) have to decide on how to proceed their mission.

Unfortunately, optimally solving Dec-POMDPs is provably intractable (Bernstein *et al.* 2002), leading to the need for smart approximate methods and good heuristics. In this work we focus on the latter, presenting a taxonomy of heuristics for Dec-POMDPs. Also we introduce a new heuristic, dubbed Q_{BG} , which is based on *Bayesian games* (BGs) and therefore, contrary to the other described heuristics, takes into account some level of decentralization.

The mentioned heuristics could be used by different methods, but we particularly focus on the approach by Emery-Montemerlo *et al.* (2004), because this approach gives a very natural introduction to the application of BGs in Dec-POMDPs.

This paper is organized as follows: First, the Dec-POMDP is formally introduced. Next, we describe how

heuristics can be used to find approximate policies using BGs. Different heuristics including the new Q_{BG} heuristic are placed in a taxonomy after that. Before we conclude with a discussion, we present a preliminary experimental evaluation of the different heuristics.

The Dec-POMDP framework

Definition 1 A *decentralized partially observable Markov decision process (Dec-POMDP)* with m agents is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, O \rangle$ where:

- \mathcal{S} is a finite set of states.
- The set $\mathcal{A} = \times_i \mathcal{A}_i$ is the set of *joint actions*, where \mathcal{A}_i is the set of actions available to agent i . Every time-step, one joint action $\mathbf{a} = \langle a_1, \dots, a_m \rangle$ is taken. Agents do not observe each other's actions.
- T is the transition function, a mapping from states and joint actions to probability distributions over states: $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$.¹
- R is the immediate reward function, a mapping from states and joint actions to real numbers: $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$.
- $\mathcal{O} = \times_i \mathcal{O}_i$ is the set of joint observations, where \mathcal{O}_i is a finite set of observations available to agent i . Every time-step one joint observation $\mathbf{o} = \langle o_1, \dots, o_m \rangle$ is received, from which each agent i observes its own component o_i .
- O is the observation function, a mapping joint actions and successor states to probability distributions over joint observations: $O : \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{P}(\mathcal{O})$.

The planning problem is to find the best behavior, or an optimal *policy*, for each agent for particular number of time-steps h , also referred to as the *horizon* of the problem. Additionally, the problem is usually specified along with an initial 'belief' $b^0 \in \mathcal{P}(\mathcal{S})$; this is the initial state distribution at time $t = 0$.²

The policies we are looking for are mappings from the histories the agents can observe to actions. Therefore we will first formalize two types of histories:

¹We use $\mathcal{P}(X)$ to denote the infinite set of probability distributions over the finite set X .

²Unless stated otherwise, all superscripts are time step indices.

Definition 2 We define the *action-observation history* for agent i , $\vec{\theta}_i^t$, as the sequence of actions taken and observations received by agent i until time-step t :

$$\vec{\theta}_i^t = (a_i^0, o_i^1, a_i^1, \dots, a_i^{t-1}, o_i^t). \quad (1)$$

The *joint action-observation history* is a tuple with the action-observation history for all agents $\vec{\theta}^t = \langle \vec{\theta}_1^t, \dots, \vec{\theta}_m^t \rangle$. The set of all action-observation histories for agent i at time t is denoted $\vec{\Theta}_i$.

Definition 3 The *observation history* for agent i is the sequence of observations an agent has received:

$$\vec{o}_i^t = (o_i^1, \dots, o_i^t). \quad (2)$$

Similar to action-observation histories, \vec{o}^t denotes a joint observation history and $\vec{\mathcal{O}}_i$ denotes the set of all observation histories for agent i .

Now we can give a definition for deterministic policies:

Definition 4 A *pure- or deterministic policy*, π_i , for agent i in a Dec-POMDP is a mapping from observation histories to actions, $\pi_i : \vec{\mathcal{O}}_i \rightarrow \mathcal{A}_i$. The set of pure policies of agent i is denoted Π_i . A pure joint policy π is a tuple containing a pure policy for each agent.

Bernstein *et al.* (2002) have shown that optimally solving a Dec-POMDP is NEXP-complete, implying that any optimal algorithm will be doubly exponential in the horizon. We will illustrate this by describing the simplest algorithm: naive enumeration of all joint policies.

Brute force policy evaluation

Because there exists an optimal pure joint policy for a finite-horizon Dec-POMDP, it is in theory possible to enumerate all different pure joint policies and choose the best one. To evaluate the expected future reward of a particular joint policy, for each state and joint observation history pair, the value can be calculated using:

$$V^{t,\pi}(s^t, \vec{o}^t) = R(s^t, \pi(\vec{o}^t)) + \sum_{s^{t+1}, \vec{o}^{t+1}} \Pr(s^{t+1}, \vec{o}^{t+1} | s^t, \pi(\vec{o}^t)) V^{t+1,\pi}(s^{t+1}, \vec{o}^{t+1}), \quad (3)$$

which can be used to calculate the value for an initial belief b^0 :

$$V(\pi) = V^{0,\pi}(b^0) = \sum_{s^0} b^0(s^0) V^{0,\pi}(s^0, \vec{o}^0). \quad (4)$$

We explained that the individual policies are mappings from observation histories to actions. The number of pure joint policies to be evaluated therefore is:

$$O\left(|\mathcal{A}_*|^{\frac{m \cdot (|\mathcal{O}_*|^h - 1)}{|\mathcal{O}_*| - 1}}\right), \quad (5)$$

where $|\mathcal{A}_*|$ and $|\mathcal{O}_*|$ denote the largest individual action and observation sets. The cost of evaluating each policy is

	$\vec{\theta}_1^{t=0}$	$\vec{\theta}_2^{t=0}$	()	
	a_1	a_2	a_2	\bar{a}_2
	\bar{a}_1	\bar{a}_1	+2.75	-4.1
	()		-0.9	+0.3

	$\vec{\theta}_1^{t=1}$	$\vec{\theta}_2^{t=1}$	(a_2, o_2)		(a_2, \bar{o}_2)		...
	a_1	a_1	a_2	\bar{a}_2	a_2	\bar{a}_2	...
	\bar{a}_1	\bar{a}_1	-0.3	+0.6	-0.6	+4.0	...
	(a_1, o_1)	(a_1, \bar{o}_1)	-0.6	+2.0	-1.3	+3.6	...
	a_1	a_1	+3.1	+4.4	-1.9	+1.0	...
	\bar{a}_1	\bar{a}_1	+1.1	-2.9	+2.0	-0.4	...
	(\bar{a}_1, o_1)	(\bar{a}_1, \bar{o}_1)	-0.4	-0.9	-0.5	-1.0	...
	\bar{a}_1	\bar{a}_1	-0.9	-4.5	-1.0	+3.5	...
	(\bar{a}_1, \bar{o}_1)

Figure 1: The Bayesian game for the first and second time-step (top: $t = 0$, bottom: $t = 1$). The entries are given by $Q(\vec{\theta}^t, \mathbf{a}^t)$ and represent the expected payoff of performing \mathbf{a}^t when the joint action-observation history is $\vec{\theta}^t$. Light shaded entries indicate the solutions. Dark entries will not be realized given $\langle a_1, a_2 \rangle$ the solution of the BG for $t = 0$. The probability for the four joint action-observation histories that can be reached given $\langle a_1, a_2 \rangle$ at $t = 0$ is uniform. Therefore, when $\langle a_1, a_2 \rangle$ gives immediate reward 0, we have that $2.75 = 0.25 \cdot 2.0 + 0.25 \cdot 3.6 + 0.25 \cdot 4.4 + 0.25 \cdot 1.0$.

$O(|\mathcal{S}| \cdot |\mathcal{O}|^h)$. The total cost of brute-force policy evaluation becomes:

$$O\left(\left(|\mathcal{A}_*|^{\frac{|\mathcal{O}_*|^h - 1}{|\mathcal{O}_*| - 1}}\right)^m \cdot |\mathcal{S}| \cdot (|\mathcal{O}_*|^h)^m\right), \quad (6)$$

which is doubly exponential in the horizon h .

Approximate policy search via Bayesian games

Besides brute force search, more sophisticated methods of searching for policies have been proposed in recent years (Hansen, Bernstein, & Zilberstein 2004; Becker *et al.* 2004; Emery-Montemerlo *et al.* 2004; Gmytrasiewicz & Doshi 2005). In this paper we focus on the algorithm of Emery-Montemerlo *et al.* (2004). This algorithm makes a forward sweep through time, solving a Bayesian game (BG) for each time step: First, the BG for $t = 0$ is constructed and solved, yielding a policy $\pi^{t=0}$, then the BG for time step $t = 1$ given $\pi^{t=0}$ is constructed and subsequently solved resulting in $\pi^{t=1}$, etc.

In this setting, a Bayesian game for time step t is defined by the joint action-observation histories $\vec{\Theta}^t$, the joint actions \mathcal{A} , a probability function $\Pr(\vec{\Theta}^t)$ and a payoff function $Q(\vec{\theta}^t, \mathbf{a})$. Figure 1 shows the Bayesian games for $t = 0$ and $t = 1$ for a fictitious Dec-POMDP with 2 agents.

Solving a Bayesian game means finding the best joint policy $\pi^{t,*}$ which is a tuple of individual policies $\langle \pi_1, \dots, \pi_m \rangle$. Each individual policy is a mapping from individual action-observation histories to individual actions $\pi_i : \vec{\Theta}_i^t \rightarrow \mathcal{A}_i$. The optimal joint policy $\pi^{t,*}$ for the Bayesian game is given

by:

$$\pi^{t,*} = \arg \max_{\pi^t} \sum_{\vec{\theta}^t \in \vec{\Theta}^t} \Pr(\vec{\theta}^t) Q(\vec{\theta}^t, \pi^t(\vec{\theta}^t)), \quad (7)$$

where $\pi^t(\vec{\theta}^t) = \langle \pi_1(\vec{\theta}_1^t), \dots, \pi_m(\vec{\theta}_m^t) \rangle$ denotes the joint action formed by execution of the individual policies.

The probability $\Pr(\vec{\theta}^t)$ of a specific action-observation history can be calculated as a product of two factors: 1) an action component and 2) an observation component. The former is the probability of performing the actions specified by $\vec{\theta}^t$, the latter is the probability that those actions lead to the observations specified by $\vec{\theta}^t$. When only considering pure policies, the aforementioned action probability component is 1 for joint action-observation histories $\vec{\theta}^t$ that are consistent with the past policy ψ^t , the policy executed for time steps $0, \dots, t-1$, and 0 otherwise. This is also illustrated in figure 1: when $\pi^{t=0}(\vec{\theta}^{t=0}) = \langle a_1, a_2 \rangle$, only the non-shaded part of the BG for $t=1$ can be reached.

If we make a forward sweep through time, i.e. consecutively solving games for time step $0, \dots, h-1$, a past joint policy ψ^t is always available. When also assuming pure policies, (7) can be rewritten as follows:

$$\pi^{t,*} = \arg \max_{\pi^t} \sum_{\substack{\vec{\theta}^t \in \vec{\Theta}^t \text{ s.t.} \\ \vec{\theta}^t \text{ consist with } \psi^t}} \Pr(\vec{\theta}^t) Q(\vec{\theta}^t, \pi^t(\vec{\theta}^t)), \quad (8)$$

Note that a joint observation history \vec{o}^t and past joint policy ψ^t uniquely identify an action-observation history $\vec{\theta}^t$. Hence the above summation can be interpreted as a summation over observation histories $\vec{o}^t \in \vec{\mathcal{O}}^t$. Therefore, the BGs effectively reduce to ones over joint observation histories rather than over joint action-observation histories. The worst-case complexity of optimally solving such BGs is:

$$O\left(\left(|\mathcal{A}_*|^{|\mathcal{O}_*|^t}\right)^m \cdot |\vec{\mathcal{O}}^t|\right), \quad (9)$$

where the first term is the number of pure joint policies for the BG and where the second term $|\vec{\mathcal{O}}^t| \equiv O(|\mathcal{O}_*|^t)^m$ is the cost of evaluating one of these pure joint policies. As the size of these BGs grows exponentially with t , they quickly become too large to build and solve exactly. To resolve this, low-probability observation histories can be pruned or clustered (Emery-Montemerlo *et al.* 2004). In this paper we will not further address this topic.

Instead, we will focus on an outstanding question regarding BGs: where does the payoff function Q come from? If we were able to calculate the optimal Q -values, the procedure as outlined would produce an optimal joint policy. Unfortunately, it is not possible to calculate the optimal Q -values in a single backward sweep through time, as is the case for MDPs and POMDPs: When trying to construct the BG for the last time step $t = h-1$, it is unclear over which $\vec{\theta}^t \in \vec{\Theta}^t$ we should sum, as there is no past policy ψ^t for time steps $0, \dots, h-2$.³ The only solution would be

³Alternatively, when interpreting the summation as one over observation histories, the probabilities $\Pr(\vec{o}^{h-1})$ are dependent on the past policy ψ .

to construct BGs for all possible past policies for time steps $0, \dots, h-2$. Effectively, this means that finding the optimal Q -values is as hard as brute-force search. Therefore, rather than trying to find optimal Q -values, employing heuristic Q -values is more plausible.

However, even when resorting to heuristic values, the summation in (8) presents us with a problem: it remains unclear over which $\vec{\theta}^t \in \vec{\Theta}^t$ the summation will be performed and therefore for which $\vec{\theta}^t$ the heuristic Q -values should be calculated. However, as the heuristics are easier to compute, in this work we will assume they are calculated for all $\vec{\theta}^t \in \vec{\Theta}^t$.

Heuristics

In this section we describe three different domain-independent heuristics that can be used as the payoff function Q for the BGs. By describing what assumptions they imply, how their values relate to each other and the optimal values, and how this trades off computational costs, we describe a taxonomy of heuristics for Dec-POMDPs.

Q_{MDP}

We start with Q_{MDP} , the heuristic used by Emery-Montemerlo *et al.* (2004). This heuristic was originally proposed in the context of POMDPs (Littman, Cassandra, & Kaelbling 1995). The idea is that the optimal action values Q^* for a single agent acting in a partially observable environment can be approximated by the expected future reward obtainable when the environment would be fully observable. These latter values can be calculated using standard dynamic programming techniques for finite horizon MDPs (Puterman 1994). Let $Q_M^t(s, a)$ denote the calculated values representing the expected future reward of state s at time step t . Then the approximated value for a belief b^t is given by:

$$Q_M(b^t, a) = \sum_{s \in \mathcal{S}} Q_M^t(s, a) b^t(s). \quad (10)$$

Q_{MDP} can also be applied in the context of Dec-POMDP by solving the ‘underlying joint MDP’, i.e. the MDP in which there is centralized operator that takes joint actions and observes the state, yielding $Q_M^t(s, \mathbf{a})$ -values. Consequently, in the Dec-POMDP setting, the Q_{MDP} -heuristic entails the assumption of centralized control as well as that of full observability of nominal states.

In order to obtain the required $Q_M(\vec{\theta}^t, \mathbf{a})$ -values, the same procedure can be used. Observe that a joint action-observation history in a Dec-POMDP also specifies a probability over states, which we will refer to as *the joint belief* $b^{\vec{\theta}^t}$ (it corresponds to the belief as would be held by a centralized operator that selects joint actions and receives joint observations). Therefore we have:

$$Q_M(\vec{\theta}^t, \mathbf{a}) \equiv Q_M(b^{\vec{\theta}^t}, \mathbf{a}) = \sum_{s \in \mathcal{S}} Q_M^t(s, \mathbf{a}) b^{\vec{\theta}^t}(s). \quad (11)$$

This computation has to be done for all $\sum_{t=0}^{h-1} (|\mathcal{A}| |\mathcal{O}|)^t = \frac{(|\mathcal{A}| |\mathcal{O}|)^h}{(|\mathcal{A}| |\mathcal{O}|) - 1}$ joint action-observation histories. Dynamic programming to calculate the $Q_M^t(s, \mathbf{a})$ -values is done in a separate phase and has a cost of $O(|\mathcal{S}| \cdot h)$. This, however, can

be ignored because the the total complexity of evaluating (11) is higher, namely:

$$O\left(\frac{(|\mathcal{A}||\mathcal{O}|)^h}{(|\mathcal{A}||\mathcal{O}|-1)} \cdot |\mathcal{S}|\right) \quad (12)$$

When used for POMDPs, it is well known that Q_{MDP} is an *admissible heuristic*, i.e., it is guaranteed to be an over-estimation of the value obtainable by the optimal POMDP policy. This is intuitively clear as Q_{MDP} assumes more information than is actually available: full observability of nominal states. Also clear is that, when applied in the setting of Dec-POMDPs, Q_{MDP} is still an admissible heuristic. For a more formal proof, we refer to (Szer, Charpillet, & Zilberstein 2005).

A side effect of the assuming full observability of nominal states, however, is that actions that gain information regarding the true state, but do not yield any reward, will have low Q_{MDP} -values. As a consequence, policies found for POMDPs using Q_{MDP} are known not to take information gaining action, where the optimal POMDP policy would (Littman, Cassandra, & Kaelbling 1995). Clearly, the same drawback is to be expected when applying Q_{MDP} in Dec-POMDP contexts.

Q_{POMDP}

Intuitively, the POMDP framework is closer to the Dec-POMDP than the MDP is. Not surprisingly, researchers have also used value functions obtained by solving the underlying POMDP as a heuristic for Dec-POMDPs (Szer, Charpillet, & Zilberstein 2005; Emery-Montemerlo 2005).

As before, Q_{POMDP} -values for use in a Dec-POMDP should be calculated from a time-indexed value function, i.e., heuristic values for time step t of a Dec-POMDP should be calculated from the POMDP values for the same time step. To do this, we propose the simplest algorithm: generating all possible joint beliefs and solving the ‘belief MDP’. Generating all possible beliefs is easy: starting with b^0 corresponding to the empty joint action-observation history $\vec{\theta}^{t=0}$, for each \mathbf{a} , for each \mathbf{o} we calculate the resulting $\vec{\theta}^{t=1}$ and corresponding belief $b^{\vec{\theta}^{t=1}}$ and continue recursively. Solving the belief MDP amounts to evaluating the following formula:

$$Q_{\text{P}}^t(\vec{\theta}^t, \mathbf{a}) = R(\vec{\theta}^t, \mathbf{a}) + \underbrace{\sum_{\mathbf{o}^{t+1} \in \mathcal{O}} \Pr(\mathbf{o}^{t+1} | \vec{\theta}^t, \mathbf{a}) \max_{\mathbf{a}^{t+1} \in \mathcal{A}} Q_{\text{P}}^{t+1}(\vec{\theta}^{t+1}, \mathbf{a}^{t+1})}_{\text{exp. future reward}}, \quad (13)$$

where $R(\vec{\theta}^t, \mathbf{a}) = \sum_{s \in \mathcal{S}} R(s, \mathbf{a}) b^{\vec{\theta}^t}(s)$ is the expected immediate reward. The probability of receiving \mathbf{o}^{t+1} , the joint observation leading to $\vec{\theta}^{t+1}$, is given by:

$$\Pr(\mathbf{o}^{t+1} | \vec{\theta}^t, \mathbf{a}) = \sum_{s^t, s^{t+1}} \Pr(s^{t+1}, \mathbf{o}^{t+1} | s^t, \mathbf{a}) b^{\vec{\theta}^t}(s) \quad (14)$$

Evaluating (13) for all joint action-observation histories $\vec{\theta}^t \in \vec{\Theta}^t$ can be done in a single backwards sweep through

	$\vec{\theta}_2^{t=0}$	()	
$\vec{\theta}_1^{t=0}$		a_2	\bar{a}_2
	a_1	+3.1	-4.1
	\bar{a}_1	-0.9	+0.3

	$\vec{\theta}_2^{t=1}$	(a_2, o_2)		(a_2, \bar{o}_2)		...
$\vec{\theta}_1^{t=1}$		a_2	\bar{a}_2	a_2	\bar{a}_2	...
(a_1, o_1)	a_1	-0.3	+0.6	-0.6	+4.0	...
	\bar{a}_1	-0.6	+2.0	-1.3	+3.6	...
(a_1, \bar{o}_1)	a_1	+3.1	+4.4	-1.9	+1.0	...
	\bar{a}_1	+1.1	-2.9	+2.0	-0.4	...
(\bar{a}_1, o_1)	a_1	-0.4	-0.9	-0.5	-1.0	...
	\bar{a}_1	-0.9	-4.5	-1.0	+3.5	...
(\bar{a}_1, \bar{o}_1)

Figure 2: Backward calculation of Q_{POMDP} -values. Note that the solutions (the highlighted entries) are different from those in figure 1. The highlighted ‘+3.1’ entry for the Bayesian game for $t = 0$ is calculated as the expected immediate reward (= 0) plus a weighted sum of the maximizing entry (joint action) per next joint observation history: $+3.1 = 0 + 0.25 \cdot 2.0 + 0.25 \cdot 4.0 + 0.25 \cdot 4.4 + 0.25 \cdot 2.0$.

time, as we mentioned earlier. This can also be visualized in Bayesian games as illustrated in figure (2); the expected future reward is calculated as a maximizing weighted sum of the entries of the next time step BG. The worst-case complexity of such a evaluation is

$$O\left(\frac{(|\mathcal{A}||\mathcal{O}|)^h}{(|\mathcal{A}||\mathcal{O}|-1)} (|\mathcal{S}| + 1)\right), \quad (15)$$

which is only slightly worse than the complexity of calculating Q_{MDP} .

It is intuitively clear that Q_{POMDP} is also an admissible heuristic for Dec-POMDPs, as it still that assumes more information is available than actually is the case. Also it should be clear that, as less assumptions are made, Q_{POMDP} should yield less of an over-estimation than Q_{MDP} . I.e., the Q_{POMDP} -values should lie between the Q_{MDP} - and optimal Q^* -values.

In contrast to Q_{MDP} , Q_{POMDP} does not assume full observability of nominal states. As a result the latter does not share the drawback of undervaluing actions that will gain information regarding the nominal state. When applied in a Dec-POMDP setting, however, Q_{POMDP} does share the assumption of centralized control. This assumption also causes a relative undervaluation which now becomes apparent: if actions will gain information regarding the joint (i.e. each other’s) observation history, this is considered redundant, while in decentralized execution this might be very beneficial, as it allows for better coordination.

Q_{BG}

We explained that the Q_{POMDP} heuristic is a tighter bound to the value function of the optimal Dec-POMDP policy than Q_{MDP} , because it makes fewer assumptions. Here we present Q_{BG} , a new heuristic which loosens the assumptions

even further. In contrast to Q_{POMDP} , Q_{BG} does not assume that at every time step t the agents know the joint (and thus each other's) action-observation history $\vec{\theta}^t$. Instead, Q_{BG} assumes that the agents know $\vec{\theta}^{t-1}$, the joint action-observation history up to time step $t-1$, and the joint action \mathbf{a}^{t-1} that was taken at the previous time step.

The intuition behind this assumption becomes clear when considering its implications. Instead of selecting the maximizing action for each joint action-observation history as illustrated in figure 2, the agents will now have to reason about the last joint observation. I.e., they will have to solve the BG for this time step restricted to the action-observation histories consistent with $\vec{\theta}^{t-1}$, \mathbf{a}^{t-1} . In the BGs for $t=1$ in figures 1 and 2 these restricted BGs are exactly the non-shaded parts. In fact, figure 1 shows the solution of the BG and how the expected value of this BG can be used at time step $t=0$. Formally Q_{BG} -values are given by:

$$Q_{\text{B}}(\vec{\theta}^t, \mathbf{a}) = R(\vec{\theta}^t, \mathbf{a}) + \max_{\pi_{\vec{\theta}^t, \mathbf{a}}} \sum_{\mathbf{o}^{t+1} \in \mathcal{O}} \Pr(\mathbf{o}^{t+1} | \vec{\theta}^t, \mathbf{a}) Q_{\text{B}}^{t+1}(\vec{\theta}^t, \pi_{\vec{\theta}^t, \mathbf{a}}(\mathbf{o}^{t+1})). \quad (16)$$

Here $\pi_{\vec{\theta}^t, \mathbf{a}} = \langle \pi_{1, \vec{\theta}^t, \mathbf{a}}, \dots, \pi_{m, \vec{\theta}^t, \mathbf{a}} \rangle$ is a joint policy for a Bayesian game as explained in the section introducing Bayesian games, but now the Bayesian game is restricted to joint action-observation histories $\vec{\theta}^{t+1}$ that are consistent with $\vec{\theta}^t$ and \mathbf{a} . This means that the individual policies are mappings from single individual observations to actions, $\pi_{i, \vec{\theta}^t, \mathbf{a}} : \mathcal{O}_i \rightarrow \mathcal{A}_i$.

Notice that difference between (16) and (13) solely lies in the position and argument of the maximization operator (a conditional policy vs. an unconditional joint action). The difference with (8) lies in the summation and the sort of policies: the latter sums over all consistent joint action-observation histories (or equivalently all joint observation-histories) and considers policies that maps from those to actions. Here the summation is only over the last joint observation and policies are mappings from this last observation to actions.

The complexity of computing Q_{BG} for all $\vec{\theta}^t, \mathbf{a}$ is given by:

$$O \left(\frac{(|\mathcal{A}| |\mathcal{O}|)^{h-1}}{(|\mathcal{A}| |\mathcal{O}|) - 1} \cdot \left[(|\mathcal{S}| |\mathcal{A}| |\mathcal{O}|) + (|\mathcal{A}_*| |\mathcal{O}_*|)^m \right] \right), \quad (17)$$

which when compared to Q_{MDP} and Q_{POMDP} (respectively (12) and (15)) contains an additional exponential term. In contrast to (9), however, this term does not depend on the horizon of the problem, but only on the number of agents, actions and observations.

Experimental evaluation

We provide some empirical evidence that Q_{BG} indeed gives a tighter bound to the optimal Dec-POMDP value function than Q_{POMDP} and Q_{MDP} . To this end we use the decentralized tiger (Dec-Tiger) test problem, which was introduced

$\mathbf{a}^{t=0}$	Q_{MDP}	Q_{POMDP}	Q_{BG}
$\langle \text{Li}, \text{Li} \rangle$	38	13.0155	8.815
$\langle \text{Li}, \text{OL} \rangle$	-6	-35.185	-50
$\langle \text{Li}, \text{OR} \rangle$	-6	-35.185	-50
$\langle \text{OL}, \text{Li} \rangle$	-6	-35.185	-50
$\langle \text{OL}, \text{OL} \rangle$	25	-4.185	-19
$\langle \text{OL}, \text{OR} \rangle$	-60	-89.185	-104
$\langle \text{OR}, \text{Li} \rangle$	-6	-35.185	-50
$\langle \text{OR}, \text{OL} \rangle$	-60	-89.185	-104
$\langle \text{OR}, \text{OR} \rangle$	25	-4.185	-19

Table 1: Q-values for the $t=0$ in the horizon 3 Dec-Tiger problem. In this case, the optimal policy specifies $\mathbf{a}^{t=0} = \langle \text{Li}, \text{Li} \rangle$ and yields an expected reward of 5.19.

by Nair *et al.* (2003) and originates from the (single agent) tiger problem (Kaelbling, Littman, & Cassandra 1998). It concerns two agents that are standing in a hallway with two doors. Behind one of the doors is a tiger, behind the other a treasure. Therefore there are two states, the tiger is behind the left door (s_l) or behind the right door (s_r). Both agents have three actions at their disposal: open the left door (OL), open the right door (OR) and listen (Li) and can only receive 2 observations: they hear the tiger in the left or in the right room.

At $t=0$ the state is s_l or s_r with 50% probability. For the exact transition-, observation- and reward function we refer to (Nair *et al.* 2003). Roughly we can say that when both agents perform Li, the state remains unchanged. In all other cases the state is reset to s_l or s_r with 50% probability. Only when both agents perform Li, they get an informative observation which is correct 85% of the time, for other joint actions the agents receive any observation with 50% probability. I.e., when the state is s_l , and both agents perform action Li, the chance that both agents hear the tiger in the left room is $.85 * .85 = .72$. Regarding the reward, we can say that that when the agents open the door for the treasure they receive a positive reward (+20), when they open the wrong door they receive a penalty (-100). When opening the wrong door jointly, the penalty is less severe (-50).

Figure 3 shows the Q-values for the horizon 5 Dec-Tiger problem. Shown are all possible joint beliefs and their maximal value. I.e., for all $b^{\vec{\theta}^t}$ the figure shows points $(b^{\vec{\theta}^t}, v)$ with

$$v = \max_{\mathbf{a}} Q(b^{\vec{\theta}^t}, \mathbf{a}). \quad (18)$$

Clearly visible is that Q_{BG} gives the tightest bound to the optimal value, followed by Q_{POMDP} and Q_{MDP} . To also get an impression of the heuristics for the other (non-maximizing) joint actions, table 1 shows the Q-values for the $t=0$ in the horizon 3 Dec-Tiger problem. Again, the relative ordering of the values of the different heuristics is the same.

Discussion

In this paper we introduced Q_{BG} , a new Q-value heuristic that can be used in solving Dec-POMDPs. We described the

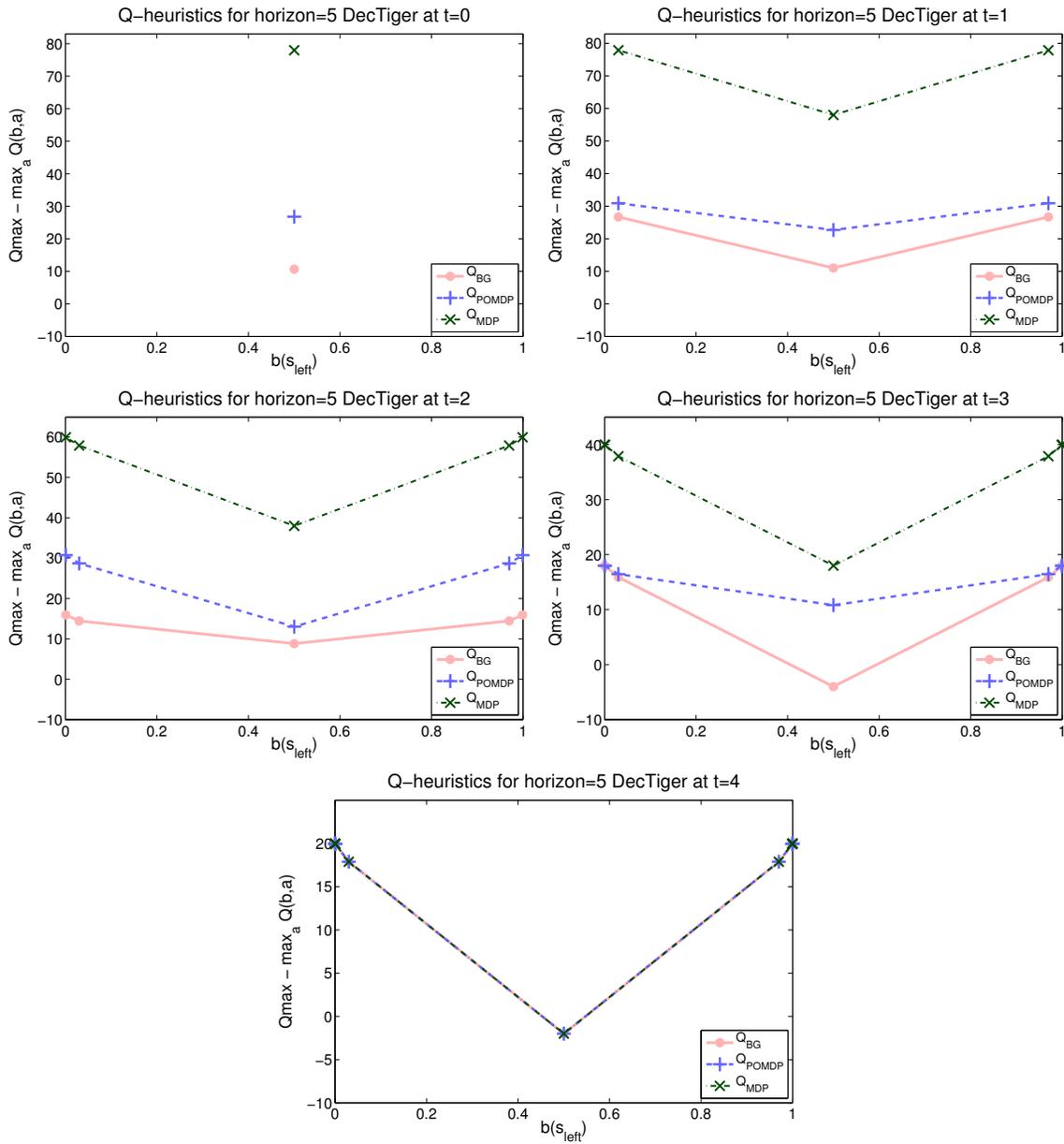


Figure 3: Q-values for horizon 5 DecTiger. For each unique joint belief $b^{\vec{\theta}^t}$, corresponding to some $\vec{\theta}^t$, the maximal $Q(b^{\vec{\theta}^t}, a)$ -value is plotted.

relation with known heuristics Q_{MDP} and Q_{POMDP} in terms of assumptions, relation to the optimal Q-values and computational cost. Experimental evaluation indicates that Q_{BG} indeed gives a tighter bound to the optimal values.

Although described in the setting of approximate planning using Bayesian games, as proposed by Emery-Montemerlo *et al.* (2004), these heuristics possibly can be used by other methods. For example, MAA* (Szer, Charpillet, & Zilberstein 2005) is an algorithm that searches the space of pure joint policies with a growing horizon and uses a heuristic to predict the value obtainable over time steps not yet considered, yielding an A*-like algorithm. When using admissible heuristics, the method is guaranteed to find an optimal policy. We would hope to find that using a tighter heuristic like Q_{BG} allows cutting larger parts of the search-tree thus speeding up the method. The application of Q_{BG} to existing and new planning methods is an important branch of future work.

The type of algorithm can also influence the computation of the heuristics. For example, although the complexities of Q_{MDP} and Q_{POMDP} are not far apart, the former can be more easily computed on-line, because the Q_{MDP} -value of a particular joint action-observation history $\bar{\theta}^t$ is not dependent on the values of successor histories $\bar{\theta}^{t+1}$. On the other hand, Q_{POMDP} and Q_{BG} , when used with approximate planning via BGs, will need to be calculated either recursively or by making a backward sweep through time before planning takes place. However, there might be other (on-line) methods, which do not require calculating all $Q(\bar{\theta}^t, \mathbf{a})$ -values up front. In this case it might be possible to calculate the Q-values in a depth-first manner, yielding exponential space savings.

A different branch of future work is to generalize Q_{BG} : instead of assuming that only the last joint observation is unknown to the agents, we can assume that the last k observations are unknown. This would mean that the BGs used to calculate Q_{BG} grow and thus computational cost will increase. On the other hand, this might yield an even tighter bound to the optimal value. When $k = h$ the agents will never know each others observations and the joint belief, thus this setting seems to reduce to the regular Dec-POMDP setting.

Acknowledgment

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

References

- Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. V. 2004. Solving transition independent decentralized Markov decision processes. *Journal of Artificial Intelligence Research (JAIR)* 22:423–455.
- Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Math. Oper. Res.* 27(4):819–840.

Emery-Montemerlo, R.; Gordon, G.; Schneider, J.; and Thrun, S. 2004. Approximate solutions for partially observable stochastic games with common payoffs. In *AAAI '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 136–143.

Emery-Montemerlo, R. 2005. *Game-Theoretic Control for Robot Teams*. Ph.D. Dissertation, Carnegie Mellon University.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24:49–79.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. In *AAAI '04: Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 709–715.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artif. Intell.* 101(1-2):99–134.

Littman, M.; Cassandra, A.; and Kaelbling, L. 1995. Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning*, 362–370.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D. V.; and Marsella, S. 2003. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 705–711.

Puterman, M. L. 1994. *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, Inc.

Szer, D.; Charpillet, F.; and Zilberstein, S. 2005. MAA*: A heuristic search algorithm for solving decentralized POMDPs. In *Proceedings of the Twenty First Conference on Uncertainty in Artificial Intelligence*.