On Markov Policies For Decentralized POMDPs

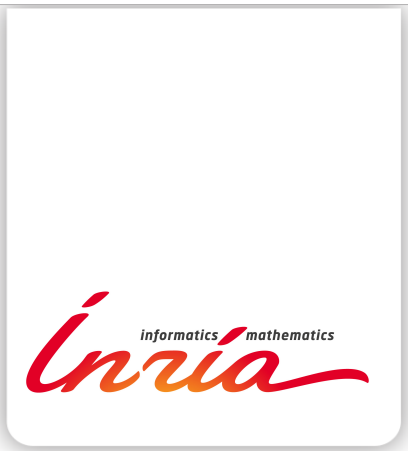
Jilles Dibangoye

On Markov Policies For Decentralized POMDPs

Jilles S. Dibangoye

RESEARCH REPORT
N° 9202
Août 2018
Project-Teams Chroma

ISSN 0249-6399 ISRN INRIA/RR--9202--FR+ENG



On Markov Policies For Decentralized POMDPs

Jilles S. Dibangoye

Project-Teams Chroma

Research Report n° 9202 — Août 2018 — 19 pages

Abstract: This paper formulates the optimal decentralized control problem for a class of mathematical models in which the system to be controlled is characterized by a finite-state discrete-time Markov process. The states of this internal process are not directly observable by the agents; rather, they have available a set of observable outputs that are only probabilistically related to the internal state of the system. The paper demonstrates that, if there are only a finite number of control intervals remaining, then the optimal payoff function of a *Markov policy* is a piecewise-linear, convex function of the current observation probabilities of the internal partially observable Markov process. In addition, algorithms for utilizing this property to calculate either the optimal or an error-bounded *Markov policy* and payoff function for any finite horizon is outlined.

Key-words: Decentralized control, centralized planning, Markov policies, Markov decision processes

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Sur les politiques Markoviennes pour les Dec-POMDPs

Résumé : Cet article formule le problème du contrôle optimal décentralisé pour une classe de modèles mathématiques dans laquelle le système à contrôler est caractérisé par un processus de Markov à temps discret et à états finis. Les états de ce processus ne sont pas directement observables par les agents; ces derniers ont à leur disposition un ensemble d'observations lié de manière probabiliste à l'état du système. L'article démontre que, s'il ne reste qu'un nombre fini de pas de décision, la mesure de performance optimale d'une politique Markovienne est une fonction convexe, linéaire par morceaux, des probabilités d'observation courantes. En outre, sont décrits les algorithmes approchés d'exploitation de cette propriété pour le calcul de politiques Markoviennes et la mesure de performance associée pour tout horizon fini.

Mots-clés : Processes décisionnels de de Markov partiallement observables et décentralisés, Planification Multi-Agents

Contents

1 Introduction

The two concepts of state and state transition are essential to the modeling of complex dynamic systems. The concept of state allows one to focus on the features system that are essential to the problem at hand, while the concept of state transition provides the mechanism for structuring the system's dynamic behavior. In most situations, there is an element of uncertainty in the transitions of the process from one state to another, and this leads naturally to the use of Markov processes as quantitative models of the system.

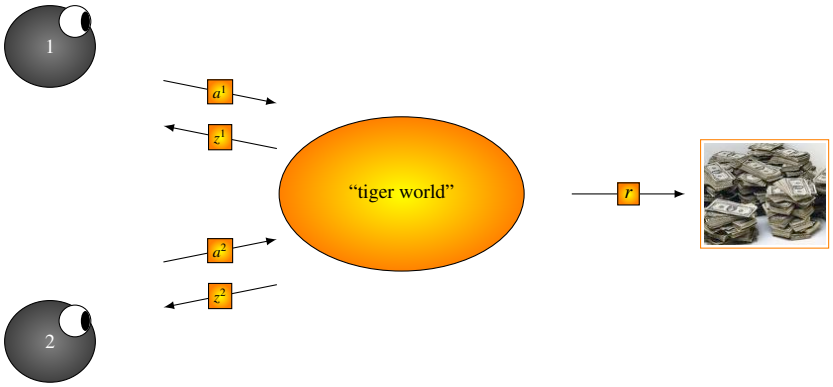


Figure 1: Decentralized Control of partially observable Markov process.

Unfortunately, in many practical applications agents are not permitted exact observation of the state of the process; rather, each agent possesses only certain local, unshared observation outputs, and acts without full knowledge of what others observe or plan to do. For example, there are many situations in game theory in which we would like to model the dynamics of the game's internal state as a Markov process, but this state is not directly observable. In such cases, we can often model what is observable as probabilistically related to the true state of the system. Figure 1 presents a pictorial representation of such a model, termed a decentralized partially observable Markov process (Witsenhausen, 1971; Yoshikawa and Kobayashi, 1978; Bernstein et al., 2002; Nair et al., 2003; Emery-Montemerlo et al., 2004). In this paper we shall consider decentralized partially observable Markov processes for which the underlying Markov process is a discrete-time finite-state Markov process; in addition, we shall limit the discussion to processes for which the number of possible outputs at each observation is finite.

As an example of this system, consider a simple two-agent "tiger world" – see Figure 1 – where the optimal control policies require the agents to coordinate their control alternatives (Nair et al., 2003). In this world there are two doors: behind one randomly chosen door is a hungry tiger, and behind the other is a pile of gold. Each agent has unique abilities. Agent 1 (the tiger listener) can hear the tiger roar – *e.g.,* z^1 – which is a noisy indication of its current location, but cannot open the doors. Agent 2 (the door opener) can open the doors – *e.g.,* a^2 – but cannot hear the roars. To facilitate communication, agent 1 has two control alternatives, signal left and signal right – *e.g.,* a^1 – which each produce a unique observation – *e.g.,* z^2 – for agent 2. When a door is opened, the world resets and the tiger is placed behind a randomly chosen door. To act optimally, agent 1 must listen to the tiger's roars until it is confident about the tiger's location and then send the appropriate signal to agent 2. Agent 2 must wait for this signal and then open the appropriate door.

This example illustrates the characteristics of the general optimal decentralized control problem for partially observable Markov processes. This paper formulates and solves this general optimal distributed problem for a process that is to operate for only a finite number of periods under the control of a Markov

policy. A later paper will examine this decentralized control problem for a process that is to operate into the indefinite future.

Over the past decade there has been extensive research into decentralized partially observable Markov processes. Earlier work on decentralized control problems relies on distributed planning (Peshkin et al., 2000). By distributed planning we meant both:

- the planning process is distributed among a variety of agents;
- a control policy is formulated that can be distributed among a variety of agents;

This is the most challenging version of distributed planning, that is when both the planning process and its results are intended to be distributed (Durfee, 2001). In this case, it might be unnecessary to ever have a multi-agent control policy represented in its entirely anywhere in the system, and yet the distributed pieces of the control policy should be compatible, which at a minimum means that agents should not conflict with each other when executing the control policies, and preferably should help each other achieve their control policies when it would be rational to do so (*e.g.,* when a helping agent is no worse off for its efforts).

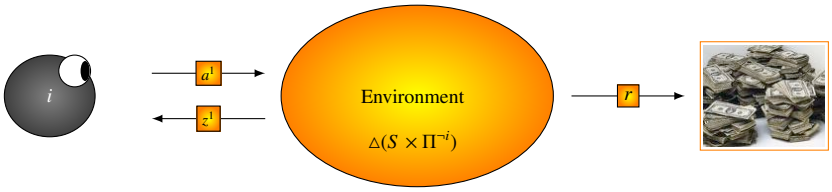


Figure 2: Agent 1's internal partially observable Markov process.

The literature of this kind of distributed planning is relatively rich and varied. It ranges from ad-hoc distributed coordination strategies, *e.g.,* logic formalization; theory of intentions (Cohen and Levesque, 1990; Jennings and Mamdani, 1992), to more formal approaches, *e.g.,* distributed constraint reasoning (Yokoo, 2001). In decentralized control of partially observable Markov processes, the control problem is broken into separate and independent partially observable Markov processes, one for each agent. Figure 2 illustrates a pictorial representation of such a model. In order to avoid sequences of control alternatives that lead to conflicts, each agent i's internal partially observable Markov process incorporates the total available information – denoted $\triangle(S \times \Pi^{\neg i})$ – of the other agents, that is, what they observe *and* plan to do. As such, the planning process for each agent is achieved independently and separately for almost all exact state-of-the-art methods. Such solution methods discussed in (Hansen et al., 2004) and (Szer and Charpillet, 2006) suggest to computing the optimal *history-dependent policy* for each agent over the continuum $\triangle(S \times \Pi^{\neg i})$. That is, the space of all possible probability distributions over what the other agents observe and plan to do. Unfortunately, the space of history-dependent policies $\Pi^{\neg i}$ of the other agents can become prohibitively huge as time goes on. More precisely, it grows doubly exponentially with the number of agents and control intervals. While such approaches preserves the ability to eventually find an optimal control policy for each agent – which is a key property – yet it makes planning impractical even for small toy problems. In attempts to scale up, specific policy classes have been addressed. (Amato et al., 2010; Szer and Charpillet, 2005; Oliehoek et al., 2008; Dibangoye et al., 2011; Oliehoek et al., 2013; Dibangoye et al., 2014, 2015, 2016) focus the effort of exact techniques only over memory-bounded history-dependent policies. (Seuken and Zilberstein, 2008; Dibangoye et al., 2009; Kumar and Zilberstein, 2010; Kumar et al., 2011, 2015) pushed a little bit further the envelop by investigating the approximate calculation of memory-bounded and history-dependent policies.

In this paper, we review the optimal Markov policy computation in the context of decentralized control of partially observable Markov decision processes (Dibangoye et al., 2012, 2013), providing new insights. From the perspective of applications, we find it comforting that by restricting attention to Markov policies, which are simple to implement and calculate, we may achieve as large expected total reward as if we used approximate memory-bounded history-dependent policies.

Of special significance for this paper is the work of (Dibangoye et al., 2016), who considered the general problem from the standpoint of centralized planning for decentralized control. (Dibangoye et al., 2016) argued that control policies that are to be executed in distributed fashion can nonetheless be formulated in centralized manner. A centralized coordinator agent with such a policy can break it into separate threads. These separate control policies can be passed to agents that can execute them. To do so, (Dibangoye et al., 2016) formulated the general problem as a partially observable Markov problem with a single constraint. The constraint induced that the resulting control policy can be broken into separate control policies, one for each agent. This process is illustrated in Figure 3. However, this work is significantly different. Indeed, it is the first research to consider control policies from the class of Markov policies. This class of policy plays a central role in our results.

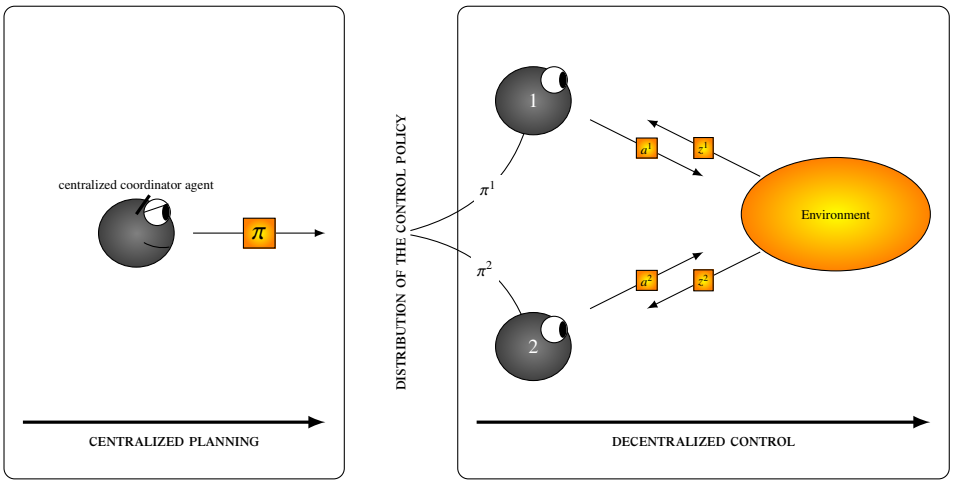


Figure 3: The centralized planning for decentralized control.

2 Properties of the Model

2.1 Basic Terminology

To begin the explicit formulation of the control problem for a decentralized partially observable Markov process, we assume that the internal dynamics of the system under control can be modeled as a $|S|$-state discrete-time Markov process. If there are τ control periods remaining, the problem is to select an alternative from the available set A that will optimize the performance of the system during its remaining lifetime. If alternative a_τ is selected, then the conditional probability that the internal process will make its next transition to state $s_{\tau-1}$ if it is presently in state s_τ will be written as

$$T(s'|s,a) \quad = \quad p(s' = s_{\tau-1}|s = s_\tau, a = a_\tau) \tag{1}$$

An observation will follow each transition, with

$$O(z'|s,a,s') \quad = \quad p(z' = z_{\tau-1}|s = s_\tau, a = a_\tau, s' = s_{\tau-1}) \tag{2}$$

denoting the probability of observing output $z_{\tau-1}$ if the internal state of the process is $s_{\tau-1}$ and alternative a_τ is controlling the system in state s_τ. It will prove convenient to define the probability of transiting from internal state s and observation output z to the internal state s' and observation output z' when agents together execute action alternative a:

$$p(s',z'|s,a) \quad = \quad T(s'|s,a) \cdot O(z'|s,a,s'). \tag{3}$$

With this representation of the process, it is easy to see that, if $O(z'|s,a,s')$ is independent of s', then the observation of the output will yield no additional information about the internal state of the process. This is the case of the nonobservable Markov process. The other extreme is the more usual case that has been studied extensively in the literature (Howard, 1960; Puterman, 1994). If there is one output for each internal state of the process and if for each alternative $O(z'|s,a,s') > 0$ if and only if $s' \equiv z'$, then the process is said to be completely observable. In the paper, we refer to the latter model as decentralized Markov decision process with full joint observability (Bernstein et al., 2002).

The calculation of an optimal control policy requires a reward structure for the process. Thus, we define

$$R(s,a) \quad = \quad r(s = s_\tau, a = a_\tau) \tag{4}$$

as the immediate award accrued if, while under the control of the alternative a_τ during one control interval, the process internal state is s_τ. The analysis to follow assumes that the centralized coordinator agent has no direct observation of the accrued rewards; that is, it only observes the outputs of the observation part of the process. If this assumption is violated, then it is easy to redefine the observation outputs of the process to include the internal states that are immediately available to the centralized coordinator agent.

These quantities (S, A, Z, R, T, O) define the probabilistic model that underlies each decentralized control problem of partially observable Markov decision process. When the agents operate over N control intervals and has a discount factor $\lambda \in [0, 1)$, the model is referred as finite horizon case with discounted rewards.

2.2 Decentralized Control Policy

The central objective of decentralized control of Markov decision process planning is to compute a decentralized control policy $\boldsymbol{\xi}$ for selecting action alternatives in order to maximize the expected sum of reward, that it gets on the next N control intervals; it should maximize

$$E_{\boldsymbol{\xi}} \left\{ \sum_{\tau=0}^{N-1} \lambda^\tau \cdot r_\tau(s_\tau, a_\tau) + r_N(s_N) \right\} \tag{5}$$

where r_τ is the reward received at control interval τ.

A decentralized control policy $\boldsymbol{\xi} \equiv \{\boldsymbol{\xi}^i\}_i$ describes the behavior $\boldsymbol{\xi}^i$ of the agents at the execution time. When there is only one control interval remaining, all the agents can do is to take a single action alternative. With two control intervals remaining, they can take an action, make an observation, then take another action. In this paper, we represent an agent's control policy $\boldsymbol{\xi}^i$ by a sequence of *decision rules* $\boldsymbol{\xi}^i := (\xi_0, \cdots, \xi_{N-1})$. A decision rule ξ_τ at time τ is a mapping from histories of observation outputs to actions. In general decentralized decision-making, there are two different classes of policies of potential interest. In the more general class, an action can be chosen on the basis of the entire history of past observations. That is, the class of *history-dependent* decision rules and policies, which are often represented

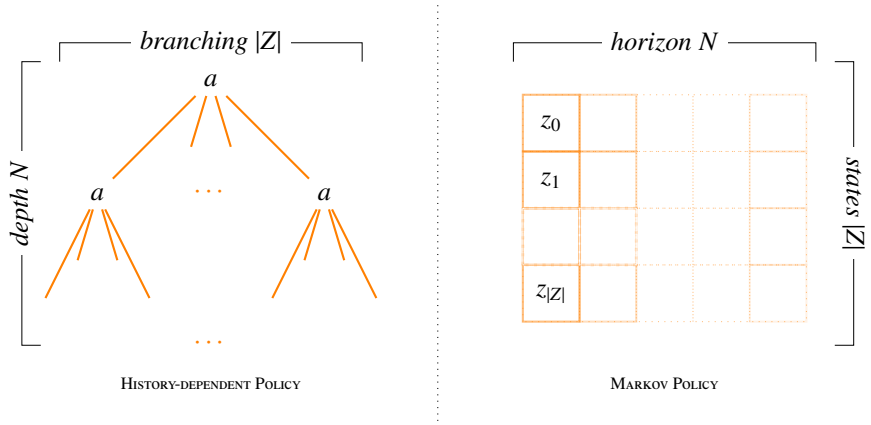


Figure 4: Exponential-size (left-hand side) vs. polynomial-size (right-hand size) deterministic control policies.

using decision tree, as illustrated in Figure 4. In a more restricted class, the action at time τ is based only on the current observation.

Agent i's policy $\boldsymbol{\xi}^i = (\xi_0^i, \xi_1^i, \cdots, \xi_{N-1}^i)$ is said to be *Markovian* if it depends on the past history only through the local observation. That is, (Markov) decision rule ξ_τ^i is a mapping from observations $z^i \in Z^i$ to local actions $a^i \in A^i$. We note $p^{\xi_\tau^i}(a^i|z^i)$ the probability of taking local action alternative a^i after perceiving local observation output z^i when executing decision rule ξ_τ^i. Since we are mainly concerned with decentralized decision making, we need to introduce a specific class of policies that is appropriate for this purpose. We define the class of decentralized control policies as follows. Team policy $\boldsymbol{\xi} = (\xi_0, \cdots, \xi_{N-1})$ is said to be a *decentralized control policy* if decision rule ξ_τ is defined as a n-tuple $(\xi_\tau^1, \cdots, \xi_\tau^n)$ of decision rules, one ξ_τ^i for each agent $i = 1, \cdots, n$. Notice that $p^{\xi_\tau}(a|z) = \prod_i p^{\xi_\tau^i}(a^i|z^i)$ is the probability of selecting joint action alternative a after perceiving joint observation z when following ξ_τ.

2.3 Sufficient Statistic

The uncertainty in the dynamic of the internal process produce uncertainty about the internal state of the system. For our formulation of the decentralized control problem, the current state of information about the internal state of the system can be encoded as the occupancy state $\boldsymbol{\theta}$, where $\theta(s, z)$ is the probability of observing output z and the internal state of the process is s. In other words, if the centralized coordinator agent has available to him his past observations of the process's outputs, then at any time the vector $\boldsymbol{\theta}$ is a sufficient statistic for these past sequences of observations. Appendix A presents a proof of this rather intuitive result.

From this result it follows that the dynamics behavior of the occupancy state $\boldsymbol{\theta}$ is itself a discrete-time continuous-state deterministic Markov process. This dynamic behavior of the state information is crucial to the calculation of the optimal as well as approximate control policy. If our prior state information about the observation output of the system is denoted by $\boldsymbol{\theta}$, and if the current decision rule is ξ_τ, then we must be able to calculate our updated state information. If $\theta'(s', z')$ is the updated probability that the observation output and internal state of the system are z' and s' respectively given the new information, then the application of simple probability operations yields the following equation (Appendix A contains

the complete derivation):

$$\theta'(s', z') = \sum_{s,z,a} p^{\xi_\tau}(a|z) \cdot \theta(s, z) \cdot p(s', z'|s, a) \tag{6}$$

Equation (6) defines a transformation from the vector $\boldsymbol{\theta}_\tau$ to the vector $\boldsymbol{\theta}_{\tau-1}$. Since this transformation plays an important role in the succeeding development, it is useful to introduce the notation

$$\boldsymbol{\theta}_{\tau-1} = \chi(\boldsymbol{\theta}_\tau|\xi_\tau) \tag{7}$$

The space of all probability distributions over S and Z is known as a standard $(|S||Z|-1)$-simplex, denoted for the sake of simplicity $\triangle$ in the remainder of the paper.

2.4 The Optimality Equations

With this as a background, the remainder of this section will introduce the optimality equations. These equations and their solutions play a central role in the theory of decentralized control of Markov decision processes.

To this end, we define $V_\tau(\boldsymbol{\theta}_\tau)$ as the maximum expected reward that the system can accrue during the lifetime of the process if the current occupancy state is $\boldsymbol{\theta}_\tau$ and there are τ control intervals remaining before the process terminates. Then, expanding over all possible next transitions yields the recursive equation (8), for any occupancy state $\boldsymbol{\theta}_\tau \in \triangle$.

$$V_\tau(\boldsymbol{\theta}_\tau) = \max_{\xi_\tau} \sum_{s,z,a} p^{\xi_\tau}(a|z) \cdot \boldsymbol{\theta}_\tau(s, z) \cdot R(s, a) + \lambda V_{\tau-1}(\chi(\boldsymbol{\theta}_\tau|\xi_\tau)) \tag{8}$$

We refer to this system of equations (8) as *the optimality equations* for decentralized control of partially observable Markov decision processes under the control of Markov policies. Equation (8) can be simplified somewhat by defining the expected immediate reward for occupancy state $\boldsymbol{\theta}_\tau$ if decision rule ξ_τ is used during the next control interval as

$$R(\boldsymbol{\theta}_\tau, \xi_\tau) := \sum_{s,z,a} p^{\xi_\tau}(a|z) \cdot \boldsymbol{\theta}_\tau(s, z) \cdot R(s, a) \tag{9}$$

Equation (8) then becomes

$$V_\tau(\boldsymbol{\theta}_\tau) = \max_{\xi_\tau} R(\boldsymbol{\theta}_\tau, \xi_\tau) + \lambda V_{\tau-1}(\chi(\boldsymbol{\theta}_\tau|\xi_\tau)) \tag{10}$$

Equation (10) is valid for any control interval $\tau \geq 1$. The value of terminating the process with a final occupancy state $\boldsymbol{\theta}_0$ is just:

$$V_0(\boldsymbol{\theta}_0) = \max_{\xi_0} R(\boldsymbol{\theta}_0, \xi_0) \tag{11}$$

Equation (10) represents a dynamic-programming problem over a continuous state space, the space of occupancy states. This is consistent with the previous assertion that the occupancy state $\boldsymbol{\theta}_\tau$ is itself the state of the discrete-time continuous state *deterministic* Markov process. Appendix A discusses this in more detail.

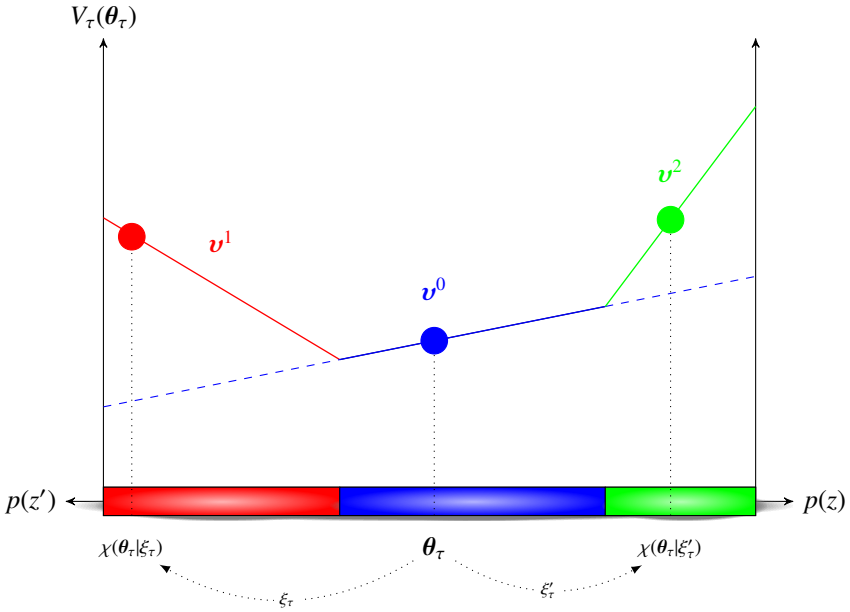


Figure 5: Information state transformation over $\{z, z'\}$ of the multi-agent tiger problem.

2.5 Convexity and Piece-Wise Linearity

We have shown that the solution of the decentralized control of Markov decision process is the solution of the optimality equations

$$\begin{aligned} V_\tau(\boldsymbol{\theta}_\tau) &= \max_{\xi_\tau} R(\boldsymbol{\theta}_\tau, \xi_\tau) + V_{\tau-1}(\chi(\boldsymbol{\theta}_\tau|\xi_\tau)) \\ V_0(\boldsymbol{\theta}_0) &= \max_{\xi_0} R(\boldsymbol{\theta}_0, \xi_0), \qquad \boldsymbol{\theta}_0 \in \triangle \end{aligned} \tag{12}$$

Although (12) appears rather formidable, its solution has a rather simple form. In particular, we shall show that $V_\tau(\boldsymbol{\theta}_\tau)$ is piecewise linear and convex, and can thus be written as

$$V_\tau(\boldsymbol{\theta}_\tau) = \max_k \ (\upsilon_\tau^\ell \cdot \boldsymbol{\theta}_\tau^\top) \tag{13}$$

for some set of vectors υ_τ^k, for $k = 1, 2, \cdots$. We shall use the term *hyperplane* to refer to one of the vectors υ_τ in Equation (13). Apart from being an amusing curiosity the convexity and piecewise linearity of the value function is particularly useful for establishing convergence properties as well as for simplification of numerical algorithms.

Before giving the main theorem we will establish some simple properties of convex functions. We have

Lemma 1. *Let $f_1(x)$ and $f_2(x)$ be convex functions. The function $f(x) = \max\{f_1(x), f_2(x)\}$ is then also convex.*

We have further

Lemma 2. *Let the function $R(\cdot, \xi_\tau)\colon \triangle \to \mathbb{R}$ be convex and let $\chi(\cdot|\xi_\tau)$ be a convex transformation which maps $\triangle$ into $\triangle$. The function $f\colon \triangle \to \mathbb{R}$ defined by $f(x) = R\left(\chi(x|\xi_\tau), \xi_\tau\right), \ \forall x \in \triangle$ is then also convex.*

Let $0 \leq \lambda \leq 1$ and $\mu = 1 - \lambda$, take $x \in \triangle$ and $y \in \triangle$, then using convexity of R_{ξ_τ} and $\chi(\cdot|\xi_\tau)$ we find

$$\begin{aligned} f(\lambda x + \mu y) &= R\left(\lambda \cdot \chi(x|\xi_\tau) + \mu \cdot \chi(y|\xi_\tau), \xi_\tau\right) && (14) \\ &\leq \lambda \cdot R\left(\chi(x|\xi_\tau), \xi_\tau\right) + \mu \cdot R_{\xi_\tau}\left(\chi(y|\xi_\tau), \xi_\tau\right) && (15) \\ &= \lambda f(x) + \mu f(y) && (16) \end{aligned}$$

and the result is established. □

We can now state the main result.

Theorem 1. *Let $\chi(\cdot|\xi_\tau)$ be mappings from $\triangle$ into $\triangle$, the functions $V_\tau \colon \triangle \to \mathbb{R}$ defined recursively by (12) are then convex.*

The linear function $R(\boldsymbol{\theta}_\tau, \xi_\tau)$ is convex. By repeated application of Lemma 1 we now find that $V_0(\boldsymbol{\theta}_0)$ is convex. Now consider $V_1(\boldsymbol{\theta}_1)$. It follows from Lemma 2 that $V_0(\chi(\boldsymbol{\theta}_1|\xi_1))$ is convex. As a sum of convex functions is convex we find that both terms within the brackets of the right member of (12) are convex. Application of Lemma 1 now shows that $V_1(\boldsymbol{\theta}_1)$ is convex. Now proceeding by induction we can show that all functions $V_\tau(\boldsymbol{\theta}_\tau)$ are convex, and the theorem is proved. □

There are two important practical points to keep in mind. First, if the set of hyperplanes for $V_{\tau-1}$ has been calculated, then it is possible to calculate the optimum decentralized control policy and the corresponding hyperplane for any specified occupancy state $\boldsymbol{\theta}_\tau$ for the τ-horizon case. This property will be most useful when we derive an algorithm for calculating the value function in Section 3. Secondly, the calculation of a new hyperplane using (13) yields an optimal decentralized control policy associated with each new hyperplane. Thus, in storing the optimal distributed decision rule, it is not necessary to store the complete description of the policy regions as illustrated in Figure 5; we need only store the set of hyperplanes along with the appropriate distributed decision rule for each hyperplane. Then, to find the optimal distributed decision rule for some occupancy state $\boldsymbol{\theta}_\tau$, we merely carry out the maximization in (13) and then use the control alternative associated with the maximizing hyperplane. This represents a considerable practical saving over previous solutions to this problem.

3 Exact Algorithm for Computing $V_\tau(\boldsymbol{\theta}_\tau)$

Having discovered the relatively simple form of the solution to the optimal decentralized control problem, it only remains to construct an orderly practical procedure for computing the hyperplanes and the corresponding mapping of these vectors onto the set of distributed alternative controls. In the succeeding discussion, we shall assume that the hyperplanes $\upsilon_{\tau-1}^k$ for the case of $(\tau - 1)$ control intervals have been calculated. The problem then is to find an algorithm for calculating the hyperplanes υ_τ^k from this information. Let $\mathbf{P}_{\xi_\tau}$ be the matrix whose the rows are $p(\cdot, \cdot|s, \xi_\tau(z))$ and R_{ξ_τ} be the vector whose components are $R(s, \xi_\tau(z))$, for any internal state and observation output (s, z). To implement the exact calculation of exact value function V_τ, we first generate hyperplanes υ_τ^{ξ} defined as follows:

$$\upsilon_\tau^{\xi} = R_{\xi_\tau} + \gamma \mathbf{P}_{\xi_\tau} \cdot \upsilon_{\tau-1}, \quad \forall \xi_\tau, \ \forall \upsilon_{\tau-1} \tag{17}$$

Finally value function V_τ is represented by the set of hyperplanes υ_τ^{ξ}.

It is often the case that an hyperplane υ_τ will be completely dominated by another hyperplane $\hat{\upsilon}_\tau$. Similarly, an hyperplane may be fully dominated by a set of other hyperplanes. Those hyperplanes can be pruned away without affecting the solution. Checking whether a single hyperplane is dominated requires solving a linear program with $|S||Z|$ variables and $|V_\tau|$ constraints. But, it can be time-effective to apply pruning after each control interval to prevent an explosion of the solution size. Formally, an hyperplane

υ is dominated, if for each occupancy state $\boldsymbol{\theta}$ there exists another hyperplane $\hat{\upsilon}$ such that $(\hat{\upsilon} - \upsilon) \cdot \boldsymbol{\theta} \geq 0$. This leads directly to the linear program below:

$$\begin{aligned} \text{maximize} \quad & \zeta \\ \text{subject to:} \quad & (\hat{\upsilon} - \upsilon) \cdot \boldsymbol{\theta} \geq \zeta, \quad (\forall \hat{\upsilon} \neq \upsilon) \\ \text{with} \quad & \textstyle\sum_{s,z} \theta(s,z) = 1 \quad (\forall \boldsymbol{\theta}) \end{aligned}$$

If the result of that program is negative or equal to zero ($\zeta \leq 0$), then hyperplane υ can be pruned away without affecting the solution.

To better understand the complexity of the exact calculation of V_τ, let $|V_{\tau-1}|$ be the number of hyperplanes required to represent value function $V_{\tau-1}$, and $\mathcal{D}$ the space of deterministic decision rules. So, in the worst case, the new solution V_τ has size $|\mathcal{D}||V_{\tau-1}|$. Given that this polynomial growth occurs for over times, the importance of pruning away unnecessary hyperplanes is clear. The most critical point of this exact algorithm for computing $V_\tau^\star$ is the linear program described above. While the theoretical complexity of solving such a problem is *only polynomial*, in practice this problem is computationally intensive since in the worst case the number of constraints increases with decreasing control intervals remaining. Notice that we consider only deterministic control policies, while the main body of the paper consider randomized control policies. We rely on the assumption that deterministic policies may achieve as large an expected discounted total reward as if we used randomized control policies.

4 An Algorithm for Approximating $V_\tau(\boldsymbol{\theta}_\tau)$

To begin the algorithm, we pick a set of representative occupancy states at control interval τ, say Θ_τ, and then applies value function calculation on those occupancy states only. As shown in the previous section, given hyperplanes at control interval $(\tau - 1)$, it is relatively straightforward to generate hyperplanes at control interval τ. In this algorithm, we apply this procedure to the entire set of occupancy states Θ_τ such that we generate a full value function at control interval τ. Given value function $V_{\tau-1}$, we compute the hyperplane $\upsilon^{\boldsymbol{\theta}_\tau}$ that is optimum for each occupancy state $\boldsymbol{\theta} \in \Theta_\tau$ as follows:

$$\upsilon^{\boldsymbol{\theta}_\tau} \quad := \quad \max_{\xi_\tau} \; R_{\xi_\tau} \cdot \boldsymbol{\theta}_\tau^\top + \gamma V_{\tau-1}(\chi(\boldsymbol{\theta}_\tau, \xi_\tau)) \tag{18}$$

To better understand the complexity of information-based approximations, let $|\Theta_\tau|$ be the number of occupancy states at control interval τ, and $|V_{\tau-1}|$ the number of hyperplanes in value function one step earlier. A full information-based approximation takes only polynomial time, *i.e.,* $|S||Z||\mathcal{D}||V_{\tau-1}||\Theta_\tau|$, and even more crucially, the size of value function V_τ remains constant, *i.e.,* $|\Theta_\tau|$, over control intervals.

Despite these outstanding results, it is likely that the information-state approximation does not scale up to realistic applications. That is mainly because, in such applications size $|\mathcal{D}|$ of the set of all possible decision rules may be too prohibitive. Moreover, a large number of hyperplanes generated when using the exact technique is pruned away since each information-state $\boldsymbol{\theta}_\tau \in \Theta_\tau$ keeps only a single hyperplane.

4.1 The Error-Bound

For any occupancy state set $\boldsymbol{\Theta}_\tau$ and control interval τ, our approximation produces V_τ. We now show that the error between V_τ and the optimal value function $V_\tau^\star$ is bounded. The bound depends on how densely $\boldsymbol{\theta}_\tau$ samples the occupancy state simplex $\triangle$; with denser sampling, V_τ converges to $V_\tau^\star$. Cutting off the approximate algorithm iterations at any control interval, we know that the difference between V_τ and the optimal $V_\tau^\star$ is not too large.

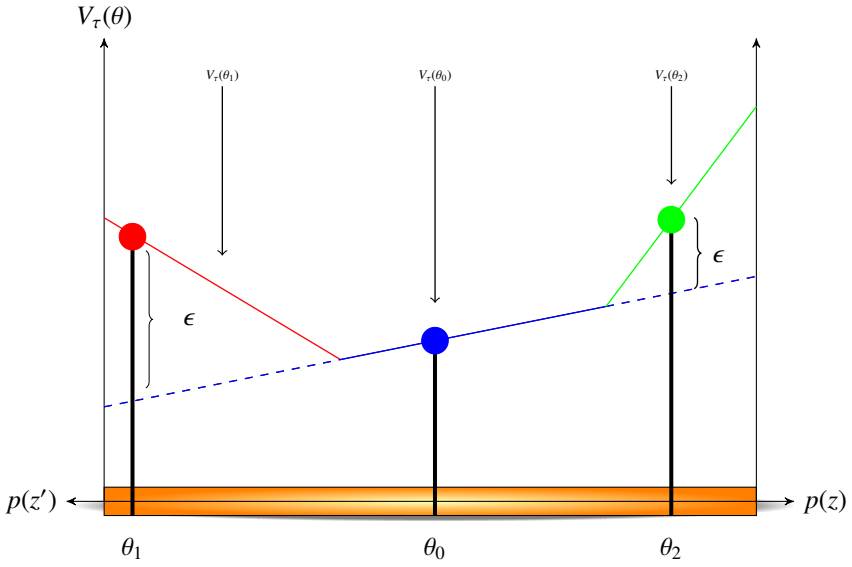


Figure 6: Theoretical properties of the approximation of V_τ.

We begin by defining the density β of a set of occupancy states $\boldsymbol{\theta}$ to be the maximum distance from any legal occupancy state to $\boldsymbol{\theta}$ (Pineau et al., 2006). More precisely:

$$\beta \quad = \quad \max_{\boldsymbol{\theta}' \in \triangle} \min_{\boldsymbol{\theta} \in \Theta_\tau} \|\theta' - \theta\|_1 \tag{19}$$

Then, we can prove that the error introduced by a single application of operator $\tilde{\mathbf{H}}$, instead of $\triangle_\tau$, is bounded by ϵ,

$$\epsilon \quad \leq \quad \frac{1-\gamma^{N-\tau}}{1-\gamma} \|r\|_\infty \beta \tag{20}$$

To better understand this error bound, let $\boldsymbol{\theta}' \in \triangle$ be the occupancy state where our approximate algorithm makes its worst error in the value update, and $\boldsymbol{\theta} \in \Theta_\tau$ be the closest (1-norm) sampled occupancy state to $\boldsymbol{\theta}'$. Let $\boldsymbol{\upsilon}$ be the hyperplane that is maximal at $\boldsymbol{\theta}$, and υ' be the hyperplane that would be maximal at $\boldsymbol{\theta}'$. By failing to include $\boldsymbol{\upsilon}'$ in its solution set, our approximation makes an error σ of at most $(\boldsymbol{\upsilon}' \cdot \boldsymbol{\theta}' - \boldsymbol{\upsilon} \cdot \theta')$. On the other hand, since $\boldsymbol{\upsilon}$ is maximal at $\boldsymbol{\theta}$, then $(\boldsymbol{\upsilon}' \cdot \boldsymbol{\theta} \leq \boldsymbol{\upsilon} \cdot \boldsymbol{\theta})$. So,

$$\begin{aligned}
\epsilon &\leq \boldsymbol{\upsilon}' \cdot \boldsymbol{\theta}' - \boldsymbol{\upsilon} \cdot \boldsymbol{\theta}' \\
&\leq \boldsymbol{\upsilon}' \cdot \boldsymbol{\theta}' - \boldsymbol{\upsilon} \cdot \boldsymbol{\theta}' + (\boldsymbol{\upsilon}' \cdot \boldsymbol{\theta} - \boldsymbol{\upsilon}' \cdot \boldsymbol{\theta}) \\
&\leq \boldsymbol{\upsilon}' \cdot \boldsymbol{\theta}' - \boldsymbol{\upsilon} \cdot \boldsymbol{\theta}' + \boldsymbol{\upsilon} \cdot \boldsymbol{\theta} - \boldsymbol{\upsilon}' \cdot \boldsymbol{\theta} \\
&\leq (\boldsymbol{\upsilon}' - \boldsymbol{\upsilon}) \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) \\
&\leq \|\boldsymbol{\upsilon}' - \boldsymbol{\upsilon}\|_\infty \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_1 \\
&\leq \frac{1-\gamma^{N-\tau}}{1-\gamma} \|r\|_\infty \beta
\end{aligned}$$

The last inequality holds for $\|r\|_\infty = \max_{s,a} |R(s,a)|$. Moreover, $\frac{1-\gamma^{N-\tau}}{1-\gamma}\|r\|_\infty$ represents the maximum reward achievable starting at any state and following some sequence of action alternatives and observation

outputs during $(N-\tau)$ control intervals. When we apply operator $\tilde{\mathbf{H}}$ over the τ control intervals remaining the error $\sigma_\tau = \frac{(1-\gamma^{N-\tau})(1-\gamma^{\tau+1})}{(1-\gamma)^2}\|r\|_\infty\beta$ produced by our approximation is given by,

$$\begin{aligned}
\epsilon_\tau &= \|V_\tau - V_\tau^*\|_\infty && (21)\\
&= \|\tilde{\mathbf{H}}V_{\tau-1} - \mathbf{H}V_{\tau-1}^*\|_\infty && (22)\\
&\leq \|\tilde{\mathbf{H}}V_{\tau-1} - \mathbf{H}V_{\tau-1}\|_\infty + \|\mathbf{H}V_{\tau-1} - \mathbf{H}V_{\tau-1}^*\|_\infty && (23)\\
&\leq \frac{1-\gamma^{N-\tau}}{1-\gamma}\|r\|_\infty\beta + \|\mathbf{H}V_{\tau-1} - \mathbf{H}V_{\tau-1}^*\|_\infty && (24)\\
&\leq \frac{1-\gamma^{N-\tau}}{1-\gamma}\|r\|_\infty\beta + \gamma\|V_{\tau-1} - V_{\tau-1}^*\|_\infty && (25)\\
&= \frac{1-\gamma^{N-\tau}}{1-\gamma}\|r\|_\infty\beta + \gamma\epsilon_{\tau-1} && (26)\\
&= \frac{(1-\gamma^{N-\tau})(1-\gamma^{\tau+1})}{(1-\gamma)^2}\|r\|_\infty\beta && (27)
\end{aligned}$$

The bound described in this section depends on how densely $\boldsymbol{\theta}$ samples the occupancy state simplex $\triangle$. In the case where not all occupancy states are reachable, we don't need to sample all of $\triangle$ densely, but can replace $\triangle$ by the set of reachable occupancy states $\bar{\triangle}$ (Figure 7).

4.2 Occupancy State Set Θ_τ Expansion

There is a clear trade-off between including fewer occupancy states (which would favor fast planning over good performance), versus including many occupancy states (which would slow down planning, but ensure better performance). This brings up the question of how many occupancy states should be included. However, the number of occupancy states is not the only consideration. It is likely that some collections of occupancy states (for example those frequently encountered) are more likely to produce a good value function than others. This brings up the question of which occupancy states should be included.

The error bound in (20) suggests that the occupancy state-based approximation performs best when its occupancy state set is uniformly dense in the set of reachable occupancy states. As shown in Figure 7, we can build a tree of reachable occupancy states. In this representation, each path through the tree corresponds to a sequence in the occupancy state space, and increasing depth corresponds to an increasing plan horizon.

As shown in this figure, the set of reachable occupancy states at control interval τ – denoted $\bar{\triangle}_\tau$ – increases exponentially with increasing time, *i.e.,* $|\bar{\triangle}_\tau| = O(|\mathcal{D}|^{\tau-1})$. Including all reachable occupancy states would guarantee optimal performance, but at the expense of computational tractability. Therefore, we must select a subset $\Theta_\tau \subset \bar{\triangle}_\tau$, which is sufficiently small for computational tractability, but sufficiently large for good value function approximation.

The approach we propose consists of initializing set Θ_0 to contain the initial occupancy state $\boldsymbol{\theta}_0$, and then gradually expanding $\boldsymbol{\theta}_\tau$ by greedily choosing new reachable occupancy states that improve the worst-case density as rapidly as possible.

To choose new reachable occupancy states, we stochastically simulate single-step forward trajectories from those occupancy states already in $\boldsymbol{\theta}_\tau$. Simulating a single-step forward trajectory for a given $\boldsymbol{\theta}_\tau \in \Theta_\tau$ requires selecting a distributed decision rule ξ_τ, and then computing the new occupancy state $\boldsymbol{\theta}_\tau^{\xi_\tau} = \chi(\boldsymbol{\theta}_\tau|\xi_\tau)$. Rather than selecting a single distributed decision rule to simulate the forward trajectory for a given $\boldsymbol{\theta}_\tau \in \Theta_\tau$, we do so with each distributed decision rule, thus producing new occupancy state set $\Theta_{\tau+1} = \{\theta^{\xi_\tau}|\forall\xi_\tau \in \mathcal{D}, \forall\theta \in \Theta_\tau\}$. Rather than accepting all new occupancy states, we calculate the L_1

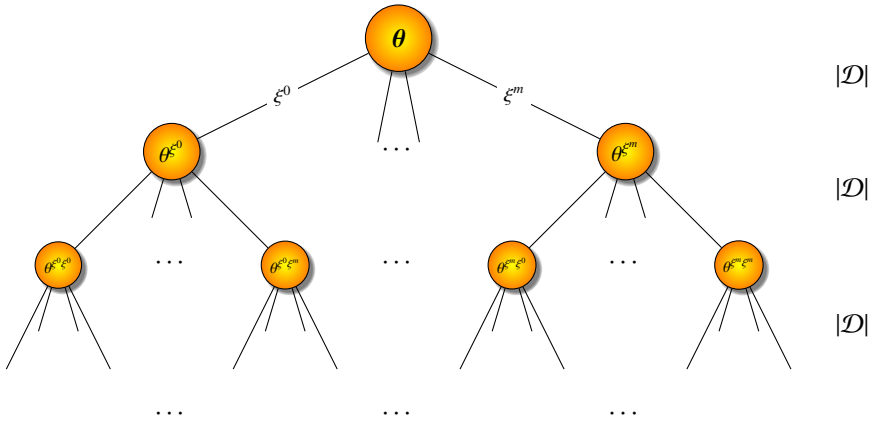


Figure 7: The reachable occupancy states.

distance between each $\boldsymbol{\theta}_\tau^{\xi_\tau}$ and its closest neighbor in $\Theta_{\tau+1}$. We then keep only the occupancy state $\boldsymbol{\theta}_\tau^{\xi_\tau}$ that is farthest away from any occupancy state already in $\Theta_{\tau+1}$.

5 Conclusion and discussion

As stated earlier, previous authors have attacked the calculation of the optimal decentralized control policy for partially observable Markov processes. Most of these exact methods are based on a single agent occupancy state – that is a mixture over the internal state of the process and all possible control policies of its teammates. With this as a background, the authors derive different smart strategies to compute the optimal history-dependent control policy for each agent. While useful, this occupancy state increases exponentially with increasing control intervals – thus restricting the ability to scale up to realistic applications. As an attempt to scale up, multiple approximate solution methods have emerged. Among all the class of information-state based approximation has shown impressive performances on the two-agent small toy problems. These methods – known as point-based algorithms – perform the update of the value function based solely on the mixture over the internal state of the system. By planning over a bounded number of such occupancy states, they favor fast planning over good performance. Indeed, a recent analysis on the solution produced by such techniques show that the error increases with decreasing number of occupancy states and increasing planning horizon. This is mainly because the occupancy state based on the mixture over the internal states of the system does not provide the sufficient statistic for decision-making in decentralized control of partially observable Markov decision processes. The other major limitation of information-states based approximations is the selection of a small parameter that bounds the number of possible control policies. Unfortunately, choosing the right parameter is not obvious, even more dynamically adjusting it may raise non negligible computation costs. Our work provides a alternative to this approximation scheme.

In this paper, we provide both theoretical and algorithmic insights for optimal as well as error-bounded approximate decentralized control of partially observable Markov decision processes using Markov policies. On the one hand, we demonstrate that the mixture over internal states of the process and observation outputs is the sufficient statistic for decision-making in such a setting. Having discovered the relatively simple form of the sufficient statistic, we demonstrate the piecewise linear, and convex nature of the value function. On the other hand, we derive a method for automatic generation of these occupancy states. This opens a large range of possible applications. From the perspective of applications, we find it comforting

that by restricting attention to Markov policies, which are simple to implement and calculate, we may achieve as large expected total reward as if we used approximate memory-bounded history-dependent policies. We hope this work lays the foundation for further work in applying Markov policies in decentralized decision-making problem.

6 Appendix A

In this appendix we show that a sufficient statistic for the past histories of a decentralized control problem of partially observable Markov processes under the control of a Markov policy is just the current observation-state vector $\boldsymbol{\theta}_\tau$. In demonstrating this property, we derive the rule for updating the observation-state vector from one control interval to the next. To make this explicit, we define $\boldsymbol{\xi}_{0:t}$ as the total available information about the process at the end of the control interval t. Notice that in this appendix the time variable t increases with increasing time, whereas in the main body of the paper the time variable τ, which is equal to the number of remaining control intervals, decreased with increasing time. For the process as defined in this paper, the only information that we obtain during a control interval is the fact that a particular *distributed* decision rule ξ_t has been executed. If $\boldsymbol{\xi}_{0:t-1}$ and ξ_t denote the *distributed* Markov policy at the end of control interval $(t-1)$, and the current decision rule, respectively, then we can write

$$\boldsymbol{\xi}_{0:t} \quad = \quad (\boldsymbol{\xi}_{0:t-1}, \xi_t) \tag{28}$$

That is, $\boldsymbol{\xi}_{0:t}$ represents our state information prior to control interval t plus the additional information that a particular distributed decision rule was recorded.

By the definition of the observation-state vector,

$$\theta_t(s', z') \quad = \quad p(s' = s_t, z' = z_t | \boldsymbol{\xi}_{0:t}), \tag{29}$$

where z_t is the discrete-valued randomized variable equal to the observation output of the process at the conclusion of control interval t. Application of the definition of conditional probability yields

$$\theta_t(s', z') \quad = \quad p(s' = s_t, z' = z_t, \boldsymbol{\xi}_{0:t}) / p(\boldsymbol{\xi}_{0:t}) \tag{30}$$

The expansion of (30) over all possible internal states and observation outputs of the process at the end of $(t-1)$ plus the expansion of the joint probability as a product of conditional probabilities produces

$$\theta_t(s', z') \quad = \quad \sum_{s,z} p(s', z' | s, z, \boldsymbol{\xi}_{0:t}) \cdot p(s, z, | \boldsymbol{\xi}_{0:t}) \tag{31}$$

The substitution of (28) into (31) yields

$$\theta_t(s', z') \quad = \quad \sum_{s,z} p(s', z' | s, z, \boldsymbol{\xi}_{0:t-1}, \xi_t) \cdot p(s, z | \boldsymbol{\xi}_{0:t-1}, \xi_t) \tag{32}$$

The first probability in (32) will be independent of $\boldsymbol{\xi}_{0:t-1}$, since the next internal state s' and observation output z' of the system depend on the past history only through the previous internal state s, observation output z and control alternative prescribed by decision rule ξ_t in $\boldsymbol{\xi}_{0:t}$. Moreover, the second probability in (32) will be independent of ξ_t, since control alternatives do not impact past internal states or observation outputs of the system. Then, expression (32) becomes:

$$\theta_t(s', z') \quad = \quad \sum_{s,z} p(s', z' | s, z, \xi_t) \cdot p(s, z | \boldsymbol{\xi}_{0:t-1}) \tag{33}$$

Expanding the first probability in (34) over all possible control alternatives $a \in A$ yields

$$\theta_t(s', z') \quad = \quad \sum_{s \in S, z \in Z} p(s, z|\xi_t) \sum_{a \in A} p^{\xi_t}(a|z) \cdot p(z'|s, a, s') \cdot p(s'|s, a) \tag{34}$$

where $p^{\xi_t}(a|z)$ denotes the probability of taking control alternative a after perceiving observation output z when following decision rule ξ_t. The remaining two probabilities in (34) are just transition probabilities and observation probabilities for the process, while the first probability is just the information-state vector defined as follows:

$$p(s = s_{t-1}, z = z_{t-1}|\boldsymbol{\xi}_{0:t-1}) \quad = \quad \theta_{t-1}(s, z)$$

Thus, we have

$$\theta_t(s', z') \quad = \quad \sum_{s \in S, z \in Z} \theta_{t-1}(s, z) \sum_{a \in A} p^{\xi_t}(a|z) \cdot O(z'|s, a, s') \cdot T(s'|s, a) \tag{35}$$

The important feature of (35) is that the calculation of the observation-state vector after control interval t requires only θ_{t-1}, the observation-state vector after control interval $t-1$; thus, θ_{t-1} summarizes all the information gained prior to control interval t and represents a sufficient statistic for the complete past history of the process $\boldsymbol{\xi}_{0:t-1}$.

In fact, (35) describes the possible transformations for a continuous-state deterministic Markov process in which the state of the process is the occupancy state vector θ_t. For this process, the transition probability from one state to another is deterministic $p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \xi_t) = 1$ if and only if $\boldsymbol{\theta}' = \chi(\boldsymbol{\theta}|\xi_t)$ otherwise $p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \xi_t) = 0$. This is a rather special case of continuous-state Markov process, since the state is continuous but the transition function is both discrete and deterministic.

Acknowledgments

References

Christopher Amato, Daniel S Bernstein, and Shlomo Zilberstein. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Journal of Autonomous Agents and Multi-Agent Systems*, 21(3):293–320, 2010.

Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*, 27(4), 2002.

Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3): 213–261, 1990.

Jilles S. Dibangoye, Abdel-Illah Mouaddib, and Brahim Chaib-draa. Point-based incremental pruning heuristic for solving finite-horizon DEC-POMDPs. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 569–576, 2009.

Jilles S. Dibangoye, Abdel-Illah Mouaddib, and Brahim Chaib-draa. Toward error-bounded algorithms for infinite-horizon DEC-POMDPs. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems*, pages 947–954, 2011.

Jilles S. Dibangoye, Christopher Amato, and Arnaud Doniec. Scaling Up Decentralized MDPs Through Heuristic Search. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 217–226, 2012.

Jilles S. Dibangoye, Christopher Amato, Arnaud Doniec, and François Charpillet. Producing Efficient Error-bounded Solutions for Transition Independent Decentralized MDPs. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems*, pages 539–546, 2013.

Jilles S. Dibangoye, Olivier Buffet, and François Charpillet. Error-Bounded Approximations for Infinite-Horizon Discounted Decentralized POMDPs. In *Proceedings of the Twenty-Fourth European Conference on Machine Learning*, pages 338–353, 2014.

Jilles S. Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. Exploiting Separability in Multiagent Planning with Continuous-State MDPs (Extended Abstract). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4254–4260, 2015.

Jilles S Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. Optimally Solving Dec-POMDPs as Continuous-State MDPs. *Journal of Artificial Intelligence Research*, 55, 2016.

Edmund H Durfee. Distributed Problem Solving and Planning. In *EASSS*, pages 118–149, 2001.

Rosemary Emery-Montemerlo, Geoffrey J Gordon, Jeff G Schneider, and Sebastian Thrun. Approximate Solutions for Partially Observable Stochastic Games with Common Payoffs. In *Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems*, pages 136–143, 2004.

Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 709–715, 2004.

Ronald A Howard. *Dynamic Programming and Markov Processes*. The M.I.T. Press, 1960.

Nicholas R. Jennings and E. H. Mamdani. Using joint responsibility to coordinate collaborative problem solving in dynamic environments. In *AAAI*, pages 269–275, 1992.

Akshat Kumar and Shlomo Zilberstein. Point-based backup for decentralized POMDPs: complexity and new algorithms. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, pages 1315–1322, 2010.

Akshat Kumar, Shlomo Zilberstein, and Marc Toussaint. Scalable Multiagent Planning Using Probabilistic Inference. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2140–2146, 2011.

Akshat Kumar, Shlomo Zilberstein, and Marc Toussaint. Probabilistic Inference Techniques for Scalable Multiagent Decision Making. *Journal of Artificial Intelligence Research*, 53:223–270, 2015.

Ranjit Nair, Milind Tambe, Makoto Yokoo, David V Pynadath, and Stacy Marsella. Taming Decentralized POMDPs: Towards Efficient Policy Computation for Multiagent Settings. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 705–711, 2003.

Frans A Oliehoek, Matthijs T J Spaan, and Nikos A Vlassis. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Frans A Oliehoek, Matthijs T J Spaan, Christopher Amato, and Shimon Whiteson. Incremental Clustering and Expansion for Faster Optimal Planning in Dec-POMDPs. *Journal of Artificial Intelligence Research*, 46:449–509, 2013.

Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning to Cooperate via Policy Search. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.

Joelle Pineau, Geoffrey J Gordon, and Sebastian Thrun. Anytime Point-Based Approximations for Large POMDPs. *Journal of Artificial Intelligence Research*, 27:335–380, 2006.

Martin L Puterman. *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. Wiley-Interscience, Hoboken, New Jersey, 1994.

Sven Seuken and Shlomo Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Journal of Autonomous Agents and Multi-Agent Systems*, 17(2):190–250, 2008.

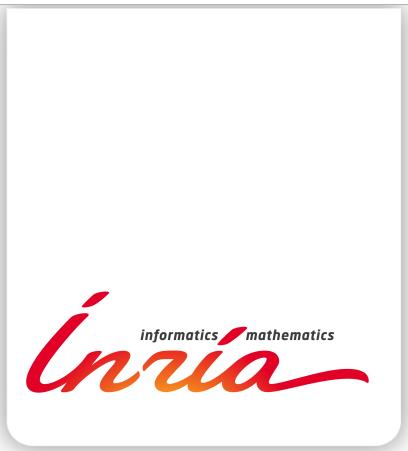
Daniel Szer and François Charpillet. An Optimal Best-First Search Algorithm for Solving Infinite Horizon DEC-POMDPs. In *Proceedings of the Fifteenth European Conference on Machine Learning*, pages 389–399, 2005.

Daniel Szer and François Charpillet. Point-based Dynamic Programming for DEC-POMDPs. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*, pages 16–20, 2006.

Hans S Witsenhausen. Separation of estimation and control for discrete time systems. In *Proceedings of the IEEE*, volume 59, pages 1557–1566, 1971.

Makoto Yokoo. *Distributed constraint satisfaction: foundations of cooperation in multi-agent systems*. Springer-Verlag, London, UK, 2001. ISBN 3-540-67596-5.

Tsuneo Yoshikawa and Hiroaki Kobayashi. Separation of estimation and control for decentralized stochastic control systems. *Automatica*, 14(6):623–628, 1978.



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399