# Decentralized Control of Partially Observable Markov Decision Processes

Christopher Amato, Girish Chowdhary, Alborz Geramifard, N. Kemal Üre, and Mykel J. Kochenderfer

*Abstract*— **Markov decision processes (MDPs) are often used to model sequential decision problems involving uncertainty under the assumption of centralized control. However, many large, distributed systems do not permit centralized control due to communication limitations (such as cost, latency or corruption). This paper surveys recent work on decentralized control of MDPs in which control of each agent depends on a partial view of the world. We focus on a general framework where there may be uncertainty about the state of the environment, represented as a decentralized partially observable MDP (Dec-POMDP), but consider a number of subclasses with different assumptions about uncertainty and agent independence. In these models, a shared objective function is used, but plans of action must be based on a partial view of the environment. We describe the frameworks, along with the complexity of optimal control and important properties. We also provide an overview of exact and approximate solution methods as well as relevant applications. This survey provides an introduction to what has become an active area of research on these models and their solutions.**

## I. INTRODUCTION

Optimal sequential decision making and control problems under uncertainty have been extensively studied both in the artificial intelligence and control systems literature (see e.g. [1]–[4]). The stochastic processes that describe the evolution of the states of many real world dynamical systems and decision domains can be assumed to satisfy the Markov property, which posits that the conditional distribution of future states of the process depends only upon the present state and the action taken at that state. Hence, the Markov Decision Process (MDP) framework has been widely used to formulate both discrete and continuous optimal decision making and control problems. Solution strategies have been developed for MDP formulations when the full state information is available, including dynamic programming [5], [6].

However, full state information is not always available in many real world problems. Åström introduced the partially observable MDP (POMDP) formulation for control with imperfect state information and showed how to transform a POMDP into a continuous-state MDP (the belief-state MDP) [7]. Since then, several solution strategies that focus on the efficiency and feasibility of obtaining a solution have been explored for POMDPs in the AI community [8]–[10]. Control problems with incomplete state information have also been tackled in the control systems literature. One of the most successful examples of this work is the Linear Quadratic Gaussian Regulator framework, which guarantees a closed form optimal control solution for output feedback control problems with linear state transition dynamics and Gaussian state transition uncertainties, representing a subclass of POMDPs [11].

Many real world problems, however, can be tackled more effectively by a collaborative approach in which various (potentially heterogeneous) agents collaborate to achieve common goals. A collaborative approach provides robustness to individual agent failures and is generally more scalable to complex, long duration missions. Examples of missions that would benefit from a collaborative approach include wide-area persistent surveillance, forward base resupply, extraterrestrial operation, and disaster mitigation (see e.g. [12]–[14]). These problems are often characterized by incomplete or partial information about the environment and the state of other agents due to limited, costly or unavailable communication. For example, not all agents may be aware of the states of other agents or may only have limited information about the states of the environment. Furthermore, it is often unrealistic to assume the existence of an all-knowing central agent for computing optimal policies. That is, it is often unreasonable or undesirable to communicate all available information to other agents or a central decision-maker. Hence, there is a significant research effort underway focused on creating decentralized decision making and control algorithms for collaborative agent networks where decision making depends on partial views of the world.

The decentralized POMDP (Dec-POMDP) model, which is an extension of the POMDP model, is one way of formulating multiagent decision making and control problems under uncertainty with incomplete or partial state information [15]. In a Dec-POMDP, each agent receives a separate observation and action choices are based solely on this local information, but there is a single global reward for the system. The dynamics of the system and the global reward depend on the actions taken by all of the agents. A desired solution maximizes a shared objective function while agents make choices based on local information. The Dec-POMDP model is more general and can potentially outperform many ofter multiagent frameworks such as consensus-based multiagent control [12], [16], [17], which assumes a given behavior rather than optimizing the action choices given the limited information. The result of this generality (which also includes general dynamics and rewards/costs) is a high complexity for generating an optimal solution in a

C. Amato is with CSAIL at MIT, Cambridge, MA. G. Chowdhary is with LIDS at MIT, Cambridge, MA and Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, OK. A. Geramifard and N. K. Üre are with LIDS at MIT, Cambridge, MA. M. J. Kochenderfer is with the Department of Aeronautics and Astronautics at Stanford University, Stanford, CA. Email: camato@csail.mit.edu, girish.chowdhary@okstate.edu, agf@csail.mit.edu, ure@mit.edu, mykel@stanford.edu.

Research is supported in part by AFOSR MURI project #FA9550-091-0538.

Dec-POMDP. In fact, it has been shown that even for just two agents, the Dec-POMDP problem is nondeterministic exponential (NEXP) complete [15]. Hence, solving decentralized multiagent optimal control problems represented as Dec-POMDPs generally involves approximation techniques and identifying additional domain structure.

In this paper, we present a brief survey of several recent advances in tackling the Dec-POMDP problem. We begin in Section II by formally discussing the Dec-POMDP model and an associated optimal solution. We then describe in Section III notable subclasses such as the Dec-MDP, network distributed POMDPs (ND-POMDPs), and Dec-POMDPs with explicit communication. In Section IV we present the computational complexity of the Dec-POMDP and a number of subclasses. We provide an overview of optimal and approximate algorithms for general Dec-POMDPs as well as some algorithms for subclasses in Section V. In Section VI, we discuss some of the application domains and some work on learning with these models (relaxing the assumption that the Dec-POMDP model is known). Finally, we conclude in Section VII.

## II. BACKGROUND

We focus on solving sequential decision making problems with discrete time steps and stochastic models with finite states, actions, and observations, though the model can be extended to continuous problems. A key assumption is that state transitions are *Markovian*, meaning that the state at time $t$ depends only on the state and events at time $t-1$. This section presents the general Dec-POMDP formulation and discusses solutions.

### A. Dec-POMDP Model

A Dec-POMDP is a tuple $\langle I, S, \{A_i\}, T, R, \{\Omega_i\}, O, h\rangle$,

- $I$, a finite set of agents.
- $S$, a finite set of states with designated initial state distribution $b_0$.
- $A_i$, a finite set of actions for each agent, $i$ with $A = \times_i A_i$ the set of joint actions, where $\times$ is the Cartesian product operator.
- $T$, a state transition probability function, $T : S \times A \times S \to [0,1]$, that specifies the probability of transitioning from state $s \in S$ to $s' \in S$ when the set of actions $\vec{a} \in A$ are taken by the agents. Hence, $T(s, \vec{a}, s') = \Pr(s'|\vec{a}, s)$.
- $R$, a reward function: $R : S \times A \to \mathbb{R}$, the immediate reward for being in state $s \in S$ and taking the set of actions $\vec{a} \in A$.
- $\Omega_i$, a finite set of observations for each agent, $i$, with $\Omega = \times_i \Omega_i$ the set of joint observations.
- $O$, an observation probability function: $O : \Omega \times A \times S \to [0,1]$, the probability of seeing the set of observations $\vec{o} \in \Omega$ given the set of actions $\vec{a} \in A$ was taken which results in state $s' \in S$, Hence $O(\vec{o}, \vec{a}, s') = \Pr(\vec{o}|\vec{a}, s')$.
- $h$, the number of steps until the problem terminates, called the horizon.

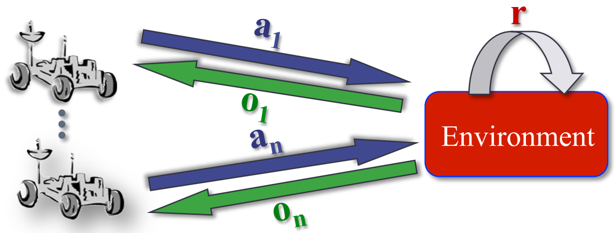

Fig. 1. Representation of $n$ agents in a Dec-POMDP setting with actions $a_i$ and observations $o_i$ for each agent $i$ along with a single reward $r$.

As depicted in Fig. 1, a Dec-POMDP [15] involves multiple agents that operate under uncertainty based on different streams of observations[1]. Like an MDP or a POMDP, a Dec-POMDP unfolds over a finite or infinite sequence of steps. At each step, every agent chooses an action (in parallel) based purely on its local observations, resulting in an immediate reward and an observation for each individual agent.

The reward is typically only used as a way to specify the objective of the task. It is generally not observed during execution. The assumption of a common shared reward allows very general formulations without having to specify sub-rewards for sub-goals.

Because the full state is not directly measured, it may be beneficial for each agent to remember a history of its measurements (i.e., observations). This problem is akin to output feedback control in which a history of output measurements is required to reconstruct the original signal [19], [20]. Unlike POMDPs, it is not typically possible to calculate a centralized estimate of the system state from the observation history of a single agent.

### B. Dec-POMDP Solutions

A solution to a Dec-POMDP is a *joint policy* or a set of policies, one for each agent in the problem. A *local policy* for an agent is a mapping from local observation histories to actions. Like the POMDP case, the goal is to maximize the total cumulative reward, beginning at some initial distribution over states $b_0$. In general, the agents do not observe the actions or observations of the other agents, but the rewards, transitions, and observations depend on the decisions of all agents. The work discussed in this paper (and the vast majority of work in the Dec-POMDP community) considers the case where the model is assumed to be known to all agents.

The value of a joint policy, $\pi$, from state $s$ is

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

which represents the expected value of the immediate reward for the set of agents summed for each step of the problem given the action prescribed by the policy until the horizon is reached. In the finite-horizon case, the discount factor, $\gamma$, is

[1] Dec-POMDPs are also related to multiagent team decision problems [18]

typically set to 1. In the infinite horizon case, as the number of steps is infinite, the discount factor $\gamma \in [0, 1)$ is included to maintain a finite sum and $h = \infty$. An *optimal policy* beginning at state $s$ is $\pi^*(s) = \text{argmax}_\pi V^\pi(s)$.

## III. NOTABLE SUBCLASSES

We now discuss a number of subclasses of Dec-POMDPs. The motivation for these subclasses is to reduce the complexity of the problem while making assumptions that match real-world problem domains.

### A. Dec-MDPs

A Dec-MDP is a Dec-POMDP that is *jointly fully observable*. Joint full observability is said to hold if the aggregated observations made by all the agents uniquely determines the global state, or if $O(\vec{o}, \vec{a}, s') > 0$ then $\Pr(s'|\vec{o}) = 1$.

A *factored $n$-agent Dec-MDP* (Dec-MDP$_n$) is a Dec-MDP where the world state can be factored into $n$ components, $S = S_1 \times \ldots \times S_n$ where each agent, $i$, possess a local state set $S_i$. Another state component, $S_0$, is sometimes added to represent an "unaffected state" that is a property of the environment which is not affected by any agent actions. For clarity reasons, we omit $S_0$ from the discussion below, but it can be incorporated in a straightforward manner. A factored, Dec-MDP$_n$ is said to be *locally fully observable* if each agent observes its own state component, $\forall o_i \, \exists s_i : \Pr(s_i|o_i) = 1$. In factored Dec-MDPs, $s_i \in S_i$ is referred to as the *local state*, $a_i \in A_i$ as the *local action* and $o_i \in \Omega_i$ as the *local observation* for agent $i$.

### B. Dec-MDPs with Independence

A factored, Dec-MDP$_n$ is said to be *transition independent* if there exists $T_1$ through $T_n$ such that

$$T(s, \vec{a}, s') = \prod_{i=1}^{n} T_i(s_i, a_i, s'_i).$$

That is, the transition probability for an agent depends only on that agent's action and previous local state. This type of independence occurs if the dynamics of agents do not interfere with each other's dynamics.

Similarly, a factored, Dec-MDP$_n$ is said to be *observation independent* if there exists $O_1$ through $O_n$ such that

$$O(\vec{o}, \vec{a}, s') = \prod_{i=1}^{n} O_i(o_i, a_i, s'_i).$$

That is, an agent's observation probability depends only on that agent's resulting local state and action. This type of independence may occur due to the lack of sensors to detect the effects of other agents on the environment, such as when agents may be operating in different locations or when they do not affect the environment at all. Many tracking problems can be assumed to be observation independent [21]–[23].

If a Dec-MDP has independent observations and transitions, then the Dec-MDP is also locally fully observable. This occurs because the observations collectively must fully determine the state of the system, but they cannot be affected by the other agents. As a result, there cannot be noise concerning local state components. A Dec-MDP with independent transitions and observations is often referred to as a *TI Dec-MDP*, dropping the observation independence label since it is implied that the observations are represented by local states in this problem.

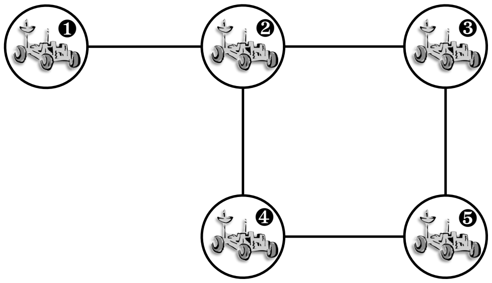

Fig. 2. An example of a networked distributed POMDP (ND-POMDP), in which transition and observation models for each agent is independent of the others, while the reward function is only dependent on the neighboring agents: $R(s, \vec{a}) = R(s_1, s_2, a_1, a_2) + R(s_2, s_3, a_2, a_3) + R(s_2, s_4, a_2, a_4) + R(s_3, s_5, a_3, a_5) + R(s_4, s_5, a_4, a_5)$.

A factored Dec-MDP$_n$ is said to be *reward independent* if there exist $R_1$ through $R_n$ such that $R((s_1, \ldots, s_n), \vec{a}) = f\left(R_1(s_1, a_1), \ldots, R_n(s_n, a_n)\right)$ and $f$ is a monotonically, non-decreasing function. These assumptions allow the reward to be decomposed in a way that ensures that the global reward is maximized by maximizing the local rewards. It is often assumed that the rewards are additive, $R(s, \vec{a}) = \sum_i R_i(s_i, a_i)$. Problems with additive rewards (but not independent transitions and observations) are very general and natural domains include various types of multi-robot foraging problems [24].

### C. Networked Distributed POMDPs

Networked distributed POMDPs (ND-POMDPs) [21] represent factored Dec-POMDPs with independent transitions and observations with an additional assumption: block reward independence. As a result, rewards in ND-POMDPs can be decomposed based on neighboring agents and summed as $R(s, \vec{a}) = \sum_l R(s_{l_1}, \ldots, s_{l_k}, s_0, \langle a_{l_1}, \ldots, a_{l_k} \rangle)$ where $l$ represents a group of $k = |l|$ neighboring agents and $s_0$ represents the "unaffected state." Also note that transition and observation independence in the factored Dec-POMDP case are the same as defined for Dec-MDPs above. Figure 2 depicts an example ND-POMDP with 5 agents and their connectivity network and a resulting set of overlapping binary reward functions. As discussed in Section VI, ND-POMDPs have been used to represent various target tracking and networking problems. While, in general, ND-POMDPs have the same worst-case complexity as general Dec-POMDPs, algorithms are able to make use of locality of interaction to solve them more efficiently in practice (as discussed in Section V).

### D. MMDPs

Another subclass is the multiagent Markov decision process (MMDP) [25]. In an MMDP, each agent is able to observe the true state of the system, making the problem fully observable. More formally, a Dec-POMDP is *fully observable* if there exists a mapping for each agent $i, f_i : \Omega_i \rightarrow S$ such that whenever $O(\vec{o}, \vec{a}, s')$ is non-zero then $f_i(o_i) = s'$. Because each agent is able to observe the true state, an MMDP can be solved as an MDP by using coordination mechanisms to ensure agent policies are consistent with each other. The MMDP model is appropriate when agents observe the true state, but still must coordinate on their selection of actions. Efficient solution methods have also been studied in similar models using factored MDPs [26], [27].

### *E. Dec-POMDPs with Explicit Communication*

While communication can be included into the actions and observations of the general Dec-POMDP model, communication can also be considered explicitly. Free, instantaneous, and lossless communication is equivalent to centralization as all agents have access to all observations at each step (allowing the problem to be solved as a POMDP [28]). When communication has a cost or can be delayed or lost, agents must reason about what and when to communicate. In particular, a Dec-POMDP with Communication (Dec-POMDP-Com) [29] augments the Dec-POMDP formulation with a set of communication messages $\Sigma$. The reward function $R(s, \vec{a}, \vec{\sigma})$ is a function of the current state, joint action, and the joint message $\vec{\sigma}$. The complexity of a Dec-POMDP-Com remains the same as a Dec-POMDP, but in some cases it may be beneficial to consider communication explicitly. For instance, it may be useful to reason about and optimize communication separately or under a different criterion. Several other communication models have also been studied [18], [30].

## IV. NUMERICAL COMPLEXITY

We first discuss the worst-case complexity of general Dec-POMDPs and Dec-MDPs, and then elaborate on the complexity of the subclasses.

Given a Dec-POMDP$_n$ and a Dec-MDP$_n$ with a value threshold and a bound on the horizon $h < |S|$, then

*Theorem 1:* $\forall n \geq 2,$ Dec-POMDP$_n \in$ NEXP.

*Theorem 2:* Dec-MDP$_2$ is NEXP-hard.

*Corollary 3:* $\forall n \geq 2,$ both Dec-POMDP$_n$ and Dec-MDP$_n$ are NEXP-complete.

The proof [15] is not included due to space considerations, but for intuition note that Dec-POMDPs (and Dec-MDPs) are solvable in NEXP time by guessing a solution in exponential time and then, given this fixed solution, evaluating it by generating the appropriate Markov process (which can be seen as an exponentially bigger belief-state MDP). The NEXP-hardness result follows from a reduction from the Tiling problem [31] (each agent must place a tile in a grid based solely on local information and the result must be consistent).

*Theorem 4:* In Dec-MDPs with independent transitions and observations (and no unobserved state $S_0$), optimal policies for each agent depend only on the local state and not on agent histories, resulting in NP-completeness.

TABLE I

WORST-CASE COMPLEXITY OF (FINITE-HORIZON) PROBLEMS

| Model | Complexity |
|---|---|
| MDP | P-complete |
| MMDP | P-complete |
| TI Dec-MDP with independent rewards | P-complete |
| TI Dec-MDP | NP-complete |
| POMDP | PSPACE-complete |
| MPOMDP | PSPACE-complete |
| ND-POMDP | NEXP-complete |
| Dec-MDP | NEXP-complete |
| Dec-POMDP-Com | NEXP-complete |
| Dec-POMDP | NEXP-complete |

The full proof [30], [32] is again deferred, but note that action and observation histories do not provide additional information about an agent's own state information (since this is locally fully observable) and because of transition and observation independence, these histories do not provide additional information about the other agents. The optimal policy for a TI Dec-MDP is a non-stationary mapping from local states (observations) to actions for each agent. While it may be somewhat surprising that Dec-MDPs have the same complexity as Dec-POMDPs, the joint full observability property only implies that the true state is known when the observations are shared, which is not the case in general.

*Theorem 5:* Dec-MDPs with independent transitions, observations and rewards can be solved independently for each agent and have resulting complexity that is P-complete.

This theorem follows from the fact that solving an MDP is P-complete [33], [34].

Table I summarizes the complexity results. Because infinite-horizon POMDPs are undecidable [35], all infinite-horizon Dec-POMDP-based models are also undecidable. Additional complexity results for these and other models have also been studied [30], [34], [36].

## V. ALGORITHMS

In this section, we consider algorithms for the case where the Dec-POMDP model is assumed to be known to all agents. Many algorithms also assume offline centralization for planning, but decentralized execution of the policy. In this way, agents can coordinate in choosing the set of policies that will be used, but the specific actions chosen and observations seen will not be known to the other agents during execution. The Dec-POMDP model does not make any assumptions about how the solution is generated (in a centralized or decentralized fashion), only that the resulting policy can be executed in a decentralized manner. Also note that POMDP algorithms cannot be easily extended to apply to Dec-POMDPs. One reason for this is that the decentralized nature of the Dec-POMDP framework results in a lack of a shared belief state, typically making it impossible to properly estimate the state of the system based on local information.

Because a shared belief state cannot typically be calculated, the policy is not typically recoverable from the value function as in POMDP methods [8]. As a result,

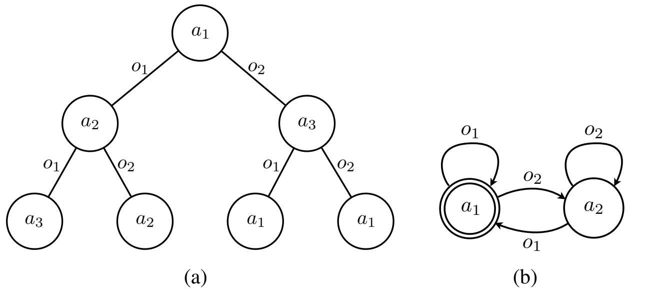

Fig. 3. A single agent's policy represented as (a) policy tree and (b) finite-state controller with initial state shown with a double circle.

explicit policies are usually maintained in the form of policy trees in the finite-horizon case or finite-state controllers in the infinite-horizon case as shown in Fig. 3. One tree or controller is maintained per agent and the policy can be extracted by starting at the root or initial node of the controller and continuing to the subtree or next node based on the observation seen. A policy can be evaluated by summing the rewards at each step weighted by the likelihood of transitioning to a given state and observing a given set of observations. For a set of agents, the value of trees or controller nodes $\vec{q}$ while starting at state $s$ is given by

$$V(\vec{q}, s) = R(\vec{a}^{\vec{q}}, s) + \sum_{s', \vec{o}} T(s', \vec{a}^{\vec{q}}, s) O(\vec{o}, \vec{a}^{\vec{q}}, s') V(\vec{q}^{\vec{o}}, s'),$$

where $\vec{a}^{\vec{q}}$ are the actions defined at $\vec{q}$, while $\vec{q}^{\vec{o}}$ are the subtrees or resulting nodes of $\vec{q}$ that are visited after $\vec{o}$ have been seen. An optimal policy can be shown to be deterministic, but stochastic controllers can be used to represent the same value with fewer nodes [37].

### A. *Optimal Approaches*

Like MDPs [5] and POMDPs [8], [10], dynamic programming methods have been used in the context of Dec-POMDPs [38]. Here, a set of $T$-step policy trees, one for each agent, is generated from the bottom up. On each step, all $t$-step policies are generated that build off the policies from step $t-1$. Thus, all 1-step trees (single actions) would be generated on the first step. Any policy that has lower value than some other policy for all states and possible policies of the other agents is then removed, or pruned (using linear programming). This generation and pruning continues until the given horizon is reached and the set of trees with the highest value at the initial state distribution is chosen. More efficient dynamic programming methods have also been developed, by reducing the number of policy trees generated at each step through reachability analysis [39] or by compressing policy representations [40]. A dynamic programming method has also been developed for generating $\epsilon$-optimal (stochastic) finite-state controllers for infinite-horizon problems [37].

Instead of computing policy trees for Dec-POMDPs using the bottom-up approach of dynamic programming, trees can also be built using a top-down approach via heuristic search [41]. In this case, a search node is a set of partial policies for the agents up to a given horizon. These partial policies can be evaluated up to that horizon and then a heuristic (such as an MDP or POMDP solution value) can be added. The resulting heuristic values are over-estimates of the true value, allowing an A*-based search [42] through the space of possible policies for the agents, expanding promising search nodes to horizon $t+1$ from horizon $t$. A more general search representation using the framework of Bayesian games was also developed [43]. Recent work has greatly improved the scalability of the original algorithms by clustering probabilistically equivalent histories and incrementally expanding nodes in the search tree [44].

Other alternatives have also been developed. One recent approach takes advantage of the centralized planning phase for decentralized control by transforming Dec-POMDPs into continuous-state MDPs with piecewise-linear convex value functions [45]. This allows powerful POMDP methods to be utilized and extended to take advantage of the structure in Dec-POMDPs, greatly increasing scalability over previous methods. Other methods include a mixed integer linear programming formulation [46] and an average reward formulation for transition independent Dec-MDPs [47].

### B. *Approximate Approaches*

While optimal solution methods for Dec-POMDPs have been an active area of research, scalability is the main concern. Hence, a number of approximate methods have been developed. The major limitation of dynamic programming approaches is the explosion of memory and time requirements as the horizon grows. This lack of scalability occurs because each step requires generating and evaluating all joint policy trees (sets of policy trees for each agent) before performing the pruning step. Memory bounded dynamic programming (MBDP) techniques mitigate this problem by keeping a fixed number of policy trees for each agent at each step [48]. A number of approaches have improved upon MBDP by limiting [49] or compressing [50] observations, replacing the exhaustive backup with branch-and-bound search in the space of joint policy trees [51] as well as constraint optimization [52] and linear programming [53] to increase the efficiency of selecting the best trees at each step.

As an alternative to MBDP-based approaches, a method called joint equilibrium search for policies (JESP) [54] utilizes alternating best response. Initial policies are generated for all agents and then all but one is held fixed. The remaining agent can then calculate a best response (local optimum) to the fixed policies. This agent's policy then becomes fixed and the next agent calculates a best response. These best response calculations to fixed other agent policies continue until no agent changes its policy. The result is a joint policy that is only locally optimal, but it may be high-valued. JESP can be made more efficient by incorporating dynamic programming in the policy generation.

Like finite-horizon approaches, methods for producing $\epsilon$-optimal infinite-horizon solutions are can also become intractable. As a result, $\epsilon$-optimal solutions cannot typically be found for any reasonable bound of the optimal solution in practice. To combat this intractability, approximate infinite-horizon algorithms have sought to produce a high quality solution while keeping the controller sizes for the agents fixed. The concept behind these approaches is to choose a controller size $|Q_i|$ for each agent and then determine what the actions and transitions should be for the set of controllers. Approximate infinite-horizon algorithms set these action selection and node transition parameters using methods such as heuristic search in the space of deterministic controllers [55], continuous optimization techniques in the space of stochastic controllers [56]–[58] or expectation maximization [59]–[61].

The above algorithms improve scalability to larger problems over optimal methods, but do not possess any bounds on solution quality. A few approximate algorithms do possess such a bound, including a method for bounding value in pruning additional policies in dynamic programming [62] and an approach that estimates the value function using repeated sampling [63].

### C. Algorithms for Subclasses

Additional methods have also been developed to solve transition and observation independent Dec-MDPs more efficiently. These methods include a bilinear programming algorithm [64] and recasting the problem as a continuous MDP with a decentralizeable policy [65].

There are ND-POMDP methods that produce quality bounded solutions [66], use finite-state controllers for agent policies [67], employ constraint-based dynamic programming [22], and combine inference techniques [68]. Other formulations for locality of interaction have also been developed. These include more general models such as factored Dec-POMDPs [69] and weakly coupled Dec-POMDPs [70] as well as models that assume agents only coordinate in certain locations [71]–[73].

A number of researchers have explored solution methods using communication. This includes using a centralized policy as a basis for communication [74], and forced synchronizing communication [75] as well as myopic communication, where an agent decides whether or not to communicate based on the assumption that the communication can take place on this step or never again [76]. Other work includes stochastically delayed communication [77] and communication for online planning in Dec-POMDPs [78].

## VI. Applications and Learning

A number of motivating applications for Dec-POMDPs have been discussed. Many of the earlier applications were motivating, but not deployed, while some of the newer work has been deployed on various platforms. Applications include multi-robot coordination in the form of space exploration rovers [79], helicopter flights [18], foraging [24] and navigation [71], [80], [81], load balancing for decentralized queues [82], network congestion control [83], [84], network routing [85], wireless networking [61] as well as sensor networks for target tracking [21], [22] and weather phenomena [23]. There is also an application of Dec-POMDPs to a real-time strategy video game.[2]

This paper discussed the planning problem in which the model is assumed to be known. Other work that is out of the scope of this paper has developed a few learning techniques that relax the model availability assumption. These approaches include model-free reinforcement learning methods using gradient-based methods to improve the policies [86], [87], learning using local signals and modeling the remaining agents as noise [88] and using communication to learn solutions in ND-POMDPs [89] and Dec-POMDPs [90].

## VII. Conclusions

The decentralized partially observable Markov decision process (Dec-POMDP) is a rich framework to formulate sequential decision making and control problems for a distributed group of agents collaborating to achieve a common goal under uncertainty. As it is often the case that communication has some cost, latency or unreliability, centralization may not be possible or may result in a poor solution. In contrast, solutions to Dec-POMDPs yield decentralized control policies that the agents execute to collaboratively optimize the common objective. However, while many more specialized multiagent models have been widely studied, the more general problem of scaling up Dec-POMDP solution methods with an increasing number of agents is still an open research question. Fortunately, there has been a large amount of work in recent years on utilizing problem structure to increase scalability in optimal and approximate solution methods as well as more scalable subclasses that relax problem assumptions which show a large amount of progress. In this paper, we surveyed the Dec-POMDP model, a number of these subclasses, provided an overview of their complexity, and discussed the main classes of solution methods. We also presented a brief overview of the significant ongoing research activity in scaling up Dec-POMDP solution methods and applying Dec-POMDP formulations to real-world problems. Due to the increasing trend of tackling real-world problems with distributed teams of heterogeneous agents, we expect that significant research activity in these areas will continue and result in even greater scalability in the near future.

## VIII. Acknowledgments

We would like to thank Shlomo Zilberstein and Matthijs Spaan for developing material in conjunction with Christopher Amato on a related tutorial which served as an inspiration for this paper.

## References

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control.* Belmont, MA: Athena Scientific, 2007, vol. I–II.

[2] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, 2010.

[3] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control.* Waltham: Blaisdell Publishing Company, 1969.

[4] R. F. Stengel, *Stochastic Optimal Control: Theory and Application.* New York: J. Wiley and Sons, 1986.

[2] See the video at http://www.screencast.com/t/M2Y2ZDA0M from Christopher Jackson, Kenneth Bogert, and Prashant Doshi.

[5] R. A. Howard, *Dynamic Programming and Markov Processes*. MIT Press, 1960.

[6] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[7] K. J. Åström, "Optimal control of Markov decision processes with incomplete state estimation," *Journal of Mathematical Analysis and Applications*, vol. 10, pp. 174–205, 1965.

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 1–45, 1998.

[9] P. Poupart, "Partially observable Markov decision processes," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 754–760.

[10] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, pp. 1–51, 2012.

[11] A. E. Bryson, *Applied Linear Optimal Control: Examples and Algorithms*. Cambridge University Press, 2002.

[12] R. Murray, "Recent research in cooperative control of multi-vehicle systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, 2007.

[13] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205–2213, 2009.

[14] Office of the Secretary of Defense, "Unmanned aerial vehicles roadmap 2002–2027," Tech. Rep., December 2002.

[15] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

[16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[17] M. Egerstedt and M. Mesbahi, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[18] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *Journal of Artificial Intelligence Research*, vol. 16, pp. 389–423, 2002.

[19] A. J. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201 – 1211, 2001.

[20] H. K. Khalil, *Nonlinear Systems*. New York: Macmillan, 2002.

[21] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo, "Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs," in *Proceedings of the Twentieth National Conference on Artificial Intelligence*, 2005.

[22] A. Kumar and S. Zilberstein, "Constraint-based dynamic programming for decentralized POMDPs with structured interactions," in *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 561–568.

[23] ——, "Event-detecting multi-agent MDPs: Complexity and constant-factor approximation," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, 2009, pp. 201–207.

[24] D. Shi, M. Z. Sauter, X. Sun, L. E. Ray, and J. D. Kralik, "An extension of Bayesian game approximation to partially observable stochastic games with competition and cooperation," in *International Conference on Artificial Intelligence*, 2010.

[25] C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999, pp. 478–485.

[26] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored MDPs," in *Advances in Neural Information Processing Systems*, ser. 15, 2001, pp. 1523–1530.

[27] C. Guestrin, S. Venkataraman, and D. Koller, "Context specific multiagent coordination and planning with factored MDPs," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 253–259.

[28] F. A. Oliehoek and M. T. J. Spaan, "Tree-based solution methods for multiagent POMDPs with delayed communication," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, July 2012, pp. 1415–1421.

[29] C. V. Goldman and S. Zilberstein, "Optimizing information exchange in cooperative multi-agent systems," in *Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems*, 2003.

[30] ——, "Decentralized control of cooperative systems: Categorization and complexity analysis," *Journal of Artificial Intelligence Research*, vol. 22, pp. 143–174, 2004.

[31] C. H. Papadimitriou, *Computational Complexity*. Addison-Wesley, 1994.

[32] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman, "Solving transition-independent decentralized Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 22, pp. 423–455, 2004.

[33] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[34] M. Allen and S. Zilberstein, "Complexity of decentralized control: Special cases," in *Advances in Neural Information Processing Systems*, ser. 22, 2009, pp. 19–27.

[35] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and related stochastic optimization problems," *Artificial Intelligence*, vol. 147, pp. 5–34, 2003.

[36] S. Seuken and S. Zilberstein, "Formal models and algorithms for decentralized control of multiple agents," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 190–250, 2008.

[37] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein, "Policy iteration for decentralized control of Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 34, pp. 89–132, 2009.

[38] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 2004, pp. 709–715.

[39] C. Amato, J. S. Dibangoye, and S. Zilberstein, "Incremental policy generation for finite-horizon DEC-POMDPs," in *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling*, 2009, pp. 2–9.

[40] A. Boularias and B. Chaib-draa, "Exact dynamic programming for decentralized POMDPs with lossless policy compression," in *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling*, 2008.

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

[42] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, July.

[43] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and approximate Q-value functions for decentralized POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, pp. 289–353, 2008.

[44] F. A. Oliehoek, M. T. J. Spaan, C. Amato, and S. Whiteson, "Incremental clustering and expansion for faster optimal planning in Dec-POMDPs," *Journal of Artificial Intelligence Research*, vol. 46, pp. 449–509, 2013.

[45] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet, "Optimally solving Dec-POMDPs as continuous-state MDPs," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[46] R. Aras, A. Dutech, and F. Charpillet, "Mixed integer linear programming for exact finite-horizon planning in decentralized POMDPs," in *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling*, 2007, pp. 18–25.

[47] M. Petrik and S. Zilberstein, "Average-reward decentralized Markov decision processes," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2007, pp. 1997–2002.

[48] S. Seuken and S. Zilberstein, "Memory-bounded dynamic programming for DEC-POMDPs," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2007, pp. 2009–2015.

[49] ——, "Improved memory-bounded dynamic programming for decentralized POMDPs," in *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, 2007, pp. 344–351.

[50] A. Carlin and S. Zilberstein, "Value-based observation compression for DEC-POMDPs," in *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[51] J. S. Dibangoye, A.-I. Mouaddib, and B. 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*, 2009.

[52] A. Kumar and S. Zilberstein, "Point-based backup for decentralized POMDPs: complexity and new algorithms," in *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 1315–1322.

[53] F. Wu, S. Zilberstein, and X. Chen, "Point-based policy generation for decentralized POMDPs," in *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 1307–1314.

[54] R. Nair, D. Pynadath, M. Yokoo, M. Tambe, and S. Marsella, "Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003, pp. 705–711.

[55] D. Szer and F. Charpillet, "An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs," in *Proceedings of the Sixteenth European Conference on Machine Learning*, 2005, pp. 389–399.

[56] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, "Bounded policy iteration for decentralized POMDPs," in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 2005, pp. 1287–1292.

[57] C. Amato, D. S. Bernstein, and S. Zilberstein, "Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 3, pp. 293–320, 2010.

[58] C. Amato, B. Bonet, and S. Zilberstein, "Finite-state controllers based on Mealy machines for centralized and decentralized POMDPs," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010, pp. 1052–1058.

[59] A. Kumar and S. Zilberstein, "Anytime planning for decentralized POMDPs using expectation maximization," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 294–301.

[60] J. K. Pajarinen and J. Peltonen, "Periodic finite state controllers for efficient POMDP and DEC-POMDP planning," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 2636–2644.

[61] J. Pajarinen and J. Peltonen, "Efficient planning for factored infinite-horizon DEC-POMDPs," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, July 2011, pp. 325–331.

[62] C. Amato, A. Carlin, and S. Zilberstein, "Bounded dynamic programming for decentralized POMDPs," in *Proceedings of the Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains, the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[63] C. Amato and S. Zilberstein, "Achieving goals in decentralized POMDPs," in *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 593–600.

[64] M. Petrik and S. Zilberstein, "A bilinear programming approach for multiagent planning," *Journal of Artificial Intelligence Research*, vol. 35, pp. 235–274, 2009.

[65] J. S. Dibangoye, C. Amato, A. Doniec, and F. Charpillet, "Producing efficient error-bounded solutions for transition independent decentralized MDPs," in *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems*, 2013.

[66] P. Varakantham, J. Marecki, Y. Yabu, M. Tambe, and M. Yokoo, "Letting loose a SPIDER on a network of POMDPs: generating quality guaranteed policies," in *Proceedings of the Sixth International Conference on Autonomous Agents and Multiagent Systems*, 2007, pp. 218:1–218:8.

[67] J. Marecki, T. Gupta, P. Varakantham, M. Tambe, and M. Yokoo, "Not all agents are equal: Scaling up distributed POMDPs for agent networks," in *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[68] A. Kumar, M. Toussaint, and S. Zilberstein, "Scalable multiagent planning using probabilistic inference," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 201, pp. 2140–2146.

[69] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis, "Exploiting locality of interaction in factored Dec-POMDPs," in *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[70] S. J. Witwicki and E. H. Durfee, "Towards a unifying characterization for quantifying weak coupling in Dec-POMDPs," in *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems*, May 2011, pp. 29–36.

[71] M. T. J. Spaan and F. S. Melo, "Interaction-driven Markov games for decentralized multiagent planning under uncertainty," in *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 525–532.

[72] P. Varakantham, J.-y. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe, "Exploiting coordination locales in distributed POMDPs via social model shaping," in *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling*, 2009, pp. 313–320.

[73] F. Melo and M. Veloso, "Decentralized MDPs with sparse interactions," *Artificial Intelligence*, 2011.

[74] M. Roth, R. Simmons, and M. Veloso, "Reasoning about joint beliefs for execution-time communication decisions," in *Proceedings of the Fourth International Conference on Autonomous Agents and Multiagent Systems*, 2005.

[75] R. Nair and M. Tambe, "Communication for improving policy computation in distributed POMDPs," in *Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems*, 2004, pp. 1098–1105.

[76] R. Becker, A. Carlin, V. Lesser, and S. Zilberstein, "Analyzing myopic approaches for multi-agent communication," *Computational Intelligence*, vol. 25, no. 1, pp. 31–50, 2009.

[77] M. T. J. Spaan, F. A. Oliehoek, and N. Vlassis, "Multiagent planning under uncertainty with stochastic communication delays," in *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling*, 2008, pp. 338–345.

[78] F. Wu, S. Zilberstein, and X. Chen, "Multi-agent online planning with communication," in *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling*, 2009.

[79] D. S. Bernstein, S. Zilberstein, R. Washington, and J. L. Bresina, "Planetary rover control as a Markov decision process," in *Proceedings of the The Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2001.

[80] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Game theoretic control for robot teams," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 1163–1169.

[81] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib, "Coordinated multi-robot exploration under communication constraints using decentralized Markov decision processes," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[82] R. Cogill, M. Rotkowitz, B. Van Roy, and S. Lall, "An approximate dynamic programming approach to decentralized control of stochastic systems," in *Proceedings of the Forty-Second Allerton Conference on Communication, Control, and Computing*, 2004.

[83] J. M. Ooi and G. W. Wornell, "Decentralized control of a multiple access broadcast channel: Performance bounds," in *Proceedings of the 35th Conference on Decision and Control*, 1996, pp. 293–298.

[84] K. Winstein and H. Balakrishnan, "TCP ex Machina: Computer-generated congestion control," in *SIGCOMM*, August 2013.

[85] L. Peshkin and V. Savova, "Reinforcement learning for adaptive routing," in *Proceedings of the International Joint Conference on Neural Networks*, 2002, pp. 1825–1830.

[86] A. Dutech, O. Buffet, and F. Charpillet, "Multi-agent systems by incremental gradient reinforcement learning," in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001, pp. 833–838.

[87] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling, "Learning to cooperate via policy search," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 489–496.

[88] Y.-H. Chang, T. Ho, and L. P. Kaelbling, "All learning is local: Multi-agent learning in global reward games," in *Advances in Neural Information Processing Systems*, ser. 16, 2004.

[89] C. Zhang and V. Lesser, "Coordinated multi-agent reinforcement learning in networked distributed POMDPs," in *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems*, 2011.

[90] ——, "Coordinating multi-agent reinforcement learning with limited communication," in *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems*, 2013.