Interactive Learning and Decision Making: Foundations, Insights & Challenges

Frans A. Oliehoek
Delft University of Technology, The Netherlands
University of Liverpool, United Kingdom
f.a.oliehoek@tudelft.nl

Abstract

Designing "teams of intelligent agents that successfully coordinate and learn about their complex environments inhabited by other agents (such as humans)" is one of the major goals of AI, and it is the challenge that I aim to address in my research. In this paper I give an overview of some of the foundations, insights and challenges in this field of Interactive Learning and Decision Making.

1 Interactive Learning and Decision Making

Interactive Learning and Decision Making (ILDM) is the term that I have started to use to describe my research. What is it? Let us start with the first term. The Oxford dictionary defines 'interactive' as: 1) *(of two people or things) influencing each other*, 2) *Allowing a two-way flow of information between a computer and a computer-user.* As such, the key characteristic of interaction is a two-way flow of influence.

In my research, I focus on sequential decision making, where we seek to control an intelligent agent or team of such agents over a number of time steps in order to optimize the performance on a particular task. The nature of such tasks can vary greatly, ranging from controlling traffic lights in a large city to decision making for teams of robots in an industrial context. They have in common, however, the need to deal with various forms of uncertainty: many applications are complex due to uncertainty of the effect of actions (outcome uncertainty), limited sensors (state uncertainty), and uncertainty about actions of other agents (agent uncertainty).

If somebody would provide a complete and accurate model of the environment of the intelligent agent(s) that we are trying to construct, the decision making task boils down to *planning*: deductive inference as to what actions lead to the best performance. However, this is rare: it is much more likely that the agent will be handed an incomplete or inaccurate model, in which case we end up in a *reinforcement learning (RL)* setting [Kaelbling *et al.*, 1996] in which the agent needs to adapt or even needs to learn from scratch.

Already in the single-agent case, these problems are inherently *interactive* due to their sequential nature: clearly, the agent affects its environment with its actuators, but the agent is also influenced by the environment (incl. possibly other agents) due to changes in state (not caused by the agent) and knowledge (about the state and dynamics). Full appreciation of this interactive nature of learning and sequential decision making will enable us to better understand such problems, and thus to come up with more usable and effective solutions.

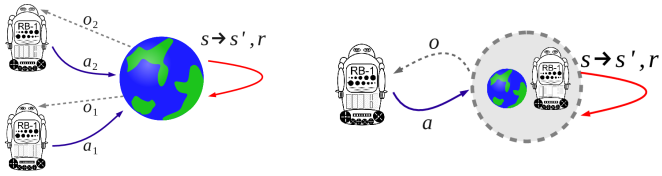

Figure 1: 'Objective' (left) and 'subjective' (right) perspective on multiagent systems.

2 Foundations

Ignoring uncertainties in decision making can lead to arbitrary poor behavior. As such, effective approaches to decision making should deal with these uncertainties, which starts with frameworks that can represent them. Before diving into technical details, however, I wish to stress that these formal models that follow below, really are just a straightforward consequence of the belief that we need to model uncertainties in order to deal with them in a principled manner. They are not a commitment to a particular solution method. So while it is understandable that many researchers shy away from, say, 'POMDPs' because "they are intractable", I feel that this misses the point: it is not the model ('POMDP'), but the problem ('decision making under state uncertainty') that causes this complexity. This certainly should affect our expectations: given the complexity results [Papadimitriou and Tsitsiklis, 1987; Littman, 1997; Bernstein *et al.*, 2000], we cannot expect to find optimal solutions in the most general cases. Instead we should investigate special cases, approximate solutions and heuristics. However, I think we should not ignore the problem and pretend these uncertainties do not exist: even if this seems to be the direction of steepest improvement, this may limit the progress we can make in the long term.

2.1 Frameworks: Dec-POMDPs and the Like

The *decentralized partially observable Markov decision process (Dec-POMDP)* framework [Bernstein *et al.*, 2000] is general enough to capture many of the aforementioned uncertainties. It takes an 'objective' approach (cf. Fig. 1) in that

it formalizes the decision problem for a team of agents. This stands in contrast to the 'subjective' approach, in which we formalize the decision making of a single agent that reasons about the other agents as part of its environment. A Dec-POMDP is a tuple $\mathcal{M} = \langle \mathcal{D}, \mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, O, h, b_0 \rangle$, where:

- $\mathcal{D} = \{1, \ldots, n\}$ is the set of n agents,
- $\mathcal{S}$ is the set of states s,
- $\mathcal{A}$ is the set of joint actions $a = \langle a_1, \ldots a_n \rangle$,
- T is the transition function that specifies $\Pr(s_{t+1}|s_t, a_t)$,
- $R(s, a)$ is the immediate reward function for the team,
- $\mathcal{O}$ is the set of joint observations $o = \langle o_1, \ldots, o_n \rangle$,
- O the observation function: $\Pr(o_{t+1}|a_t, s_{t+1})$,
- h is the horizon of the problem (finite or infinite),
- $b_0 \in \triangle(\mathcal{S})$, is the initial state distribution at time $t = 0$.

The stochastic transition function T models outcome uncertainty, while state uncertainty is modeled by O. At every time step or *stage*, each agent selects an individual action based on its *individual* observations, which means that agents are not certain what actions their teammates will take. There is no 'explicit' communication, but note that via the actions and observations the agents can communicate in pretty much the same way as human brains communicate to one another. In fact, it is possible to create a subset of 'message actions' (and corresponding observations) particularly for communication [Pynadath and Tambe, 2002; Goldman and Zilberstein, 2003]. An optimal plan for such a model, while difficult to compute, will 'embed' the optimal meaning to these messages void of a priori semantics.

Alternatively, it is possible to consider explicit communication: e.g., at each stage all agents might broadcast their individual observations. Under noise-free and cost-free communication this is the optimal thing to do [Pynadath and Tambe, 2002]. The resulting model is typically referred to as a *multiagent POMDP (MPOMDP)* and can be interpreted as a centralized model [Messias *et al.*, 2011].

Many other frameworks can be seen as special cases of the Dec-POMDP: a regular *POMDP* [Kaelbling *et al.*, 1998] is the special case with a single agent. And if this agent can perfectly observe the (Markov) state of the system we deal with an *MDP* [Bellman, 1957; Puterman, 1994]. An MPOMDP where the state is observable is referred to as a *multiagent MDP (MMDP)* [Boutilier, 1996]. A generalization of the Dec-POMDP in which each agent has its individual reward function is a *partially observable stochastic game (POSG)* [Hansen *et al.*, 2004]. For a detailed description of these *multiagent decision processes* see [Oliehoek and Amato, 2016].

2.2 Planning

If we have access to the entire model $\mathcal{M}$ (or an accurate *simulator*), including transition and possibly observation probabilities, of a (multiagent) decision process, we are dealing with a *planning* (or a *simulation-based planning*) problem.[1]

[1]Simulation-based planning is often treated as learning, but it offers more opportunities than the full RL setting (e.g., resetting the simulator to a desired state). I find it useful to discriminate from settings where there is inherent uncertainty about the model.

The goal is to compute a (joint) policy $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ that maximizes a certain optimality criterion, such as the expected ($\gamma \in [0, 1]$-discounted) cumulative reward, also referred to as *value*: $V(\pi) = \mathbf{E}\left[\sum_{t=0}^{h-1} \gamma^t R(s_t, a_t) \mid b_0, \pi\right]$. In an MDP, the Bellman optimality equations can represent this value of an optimal policy recursively:

$$Q^*(s_t, a_t) = R(s_t, a_t) + \gamma^t \sum_{s_{t+1}} \Pr(s_{t+1}|s_t, a_t) V^*(s_{t+1}),$$

with $V^*(s_t) = \max_{a_t} Q^*(s_t, a_t)$. Clearly, this is directly applicable to MMDPs too (and similar MPOMDPs can directly rely on value function formulations for POMDPs).

A key difficulty that sets Dec-POMDPs apart from frameworks as MPOMDPs and MMDPs, is that the joint policy is *decentralized:* the individual policy π_i of every agent i is a mapping from *individual observations histories* $\bar{o}_{i,t} = (o_{i,1}, \ldots, o_{i,t})$ to actions $\pi_i(\bar{o}_{i,t}) = a_{i,t}$. This decentralization has a profound effect on the complexity: Dec-POMDPs are *provably* intractable (NEXP-complete) [Bernstein *et al.*, 2000]. This is not to say that MMDPs are easy: while they can be solved in polynomial time, the size of their representation itself is already exponential in the number of agents.

2.3 Learning

In the event that $\mathcal{M}$ is not completely, or not accurately specified we are in a (multiagent) reinforcement learning setting where the agent(s) need to learn about the environment *while interacting with it*. For instance, we can think of $\mathcal{M} = \langle \mathcal{D} = \{1\}, \cdot, \mathcal{A} = \{\mathcal{A}_1\}, \cdot, \cdot, \mathcal{O} = \{\mathcal{O}_1\}, \cdot, h, \cdot \rangle$, where the missing entries ($\cdot$) are unknown, as a canonical single-agent *partially observable reinforcement learning (PORL) problem*. Phrasing this in the context of a larger Dec-POMDP tuple is a minimal assumption: it merely means that we believe that there could exist some latent state space $\mathcal{S}$, which would render the system Markovian. It does *not* commit us to picking a learning method that also aims to reconstruct that Markovian state. Moreover, depending on what we know in advance about the environment, we can use different frameworks as a starting point for our algorithm design: if we know that the agent's percepts $\mathcal{O}_1$ are Markovian, the MDP model would be a good starting point.

Clearly, the variety in the number of agents, the type and amount of knowledge that might be available on missing parts from the tuple $\mathcal{M}$, and (explicit) communication constraints that we expect to see in different applications is huge, and a full taxonomy is beyond the scope of this paper. I believe, however, that grounding such taxonomies in rich formal frameworks will yield better understanding of how different methods relate, and what properties of the problem they exploit.

3 Some Insights

Given these foundations, I will try and give a high-level overview of some of the insights that I have contributed to in the last decade.

3.1 Decentralization and Value

A single agent, indexed 1, in a POMDP can avoid remembering the entire action-observation history $\bar{\theta}_{1,t} = (a_{1,0}, o_{1,1}, \ldots, a_{1,t-1}, o_{1,t})$. Instead it can maintain a *belief* b_1, a posterior probability distribution over states $b_{1,t}(s) = \Pr(s|\bar{\theta}_{1,t}, b_0)$, since such a belief is a sufficient statistic to accurately predict the optimal value the agent can expect in the future [Kaelbling *et al.*, 1998; Bertsekas, 2007, p.251]. In a Dec-POMDP, however, such sufficient statistics that individual agents i could use to summarize $\bar{\theta}_{i,t}$ are not identified, and may not exist. Instead agents need to act based on observation histories $\bar{o}_{i,t}$ (actions can be discarded, as they can be inferred given deterministic policies).

An extension of the belief, called *multiagent belief,* $b_{i,t}(s, q_{\neq i})$, is defined over states s and the policies $q_{\neq i}$ (represented as trees) that the other agents will follow in the *future* [Hansen *et al.*, 2004]. This enables a form of dynamic programming: one can compute sets of policy trees q_i for increasing horizons $1, \ldots, h-1$ for each agent, pruning those that are dominated over the entire space of multiagent beliefs.

Other approaches are more like the POMDP belief and instead look at a statistic that summarize the *past*. As stated above, no statistics of the history are known that the agents can maintain during execution. Instead, these *plan-time sufficient statistics* capture information about the *policies* executed up to some stage t [Nayyar *et al.*, 2011; Dibangoye *et al.*, 2013; Oliehoek, 2013]. In particular, a (joint) policy can be seen as a sequence of (joint) decision rules $\pi = (\delta_0, \ldots, \delta_{h-1})$, and we can define partially specified joint policies $\varphi_t = (\delta_0, \ldots, \delta_{t-1})$. Now, the optimal value function for a Dec-POMDP can be defined as a function $V^*(b_0, \varphi_t)$ [Oliehoek *et al.*, 2008a; Oliehoek, 2010]. While this is a reasonably intuitive description with links to the notion of sequential rationality in game theory, it does not offer computational leverage.

However, it turns out that (for deterministic φ_t) we can replace the dependence of V^* on (b_0, φ_t) by a distribution over joint observation histories and states: $\sigma_t(s_t, \bar{o}_t) \triangleq \Pr(s_t, \bar{o}_t|b_0, \varphi_t)$ [Oliehoek, 2013]. This not only highlights the importance of how information is distributed (who observed what), it also provides computational leverage. It forms a basis for the lossless clustering of observation histories $\bar{o}_{i,t}$ [Oliehoek *et al.*, 2013a], and, similar to POMDPs, V^* is a piecewise-linear and convex function of σ_t [Nayyar *et al.*, 2011]. In fact, one can show that a Dec-POMDP can be converted to a *non-observable* MDP [Oliehoek and Amato, 2014b] to which POMDP methods apply and this approach has led to a significant increase in scalability of approximation methods for Dec-POMDPs [Dibangoye *et al.*, 2013]. This extends to settings with restricted classes of policies (e.g., finite state controllers) [Oliehoek and Amato, 2014b; MacDermed and Isbell, 2013], and these results can be extended to 2-player zero-sum POSGs [Wiggers *et al.*, 2016].

3.2 Factorization, Abstraction & Transfer

Scalability of exact methods has inherent limitations. In this section, I highlight some advances that provide scalability at the expense of guarantees.

Factored Value Functions To overcome the problem of exponentially large (in the number of agents) representations, structured representations such as *factored* MDPs [Boutilier *et al.*, 1999] or factored Dec-POMDPs [Nair *et al.*, 2005; Oliehoek *et al.*, 2008b] were introduced. These represent a state $s = \langle x^1, \ldots, x^m \rangle$ using m state variables, or *factors,* which enables compact representations of transitions, observations and rewards. Unfortunately, in general, value functions cannot be represented compactly [Koller and Parr, 1999] (even though exact algorithms that exploit structure are possible [Scharpff *et al.*, 2016]). In response, approximate solutions have been proposed that factorize the (Q-)value function as the sum of *individual* terms $Q(s,a) \approx \sum_{i \in \mathcal{D}} Q_i(x_i, a_i)$, or sums of *local* terms $Q(s,a) \approx \sum_{e \in \mathcal{E}} Q^e(x_e, a_e)$, where the components $e \in \mathcal{E}$ are defined over subsets of state factors x_e and agent actions a_e. The set $\mathcal{E}$ is a set of subsets of agents; corresponding to the hyper-edges in an interaction hyper-graph [Guestrin *et al.*, 2002a; Nair *et al.*, 2005].

Such *factored (Q-)value functions* were used in the context of MMDPs [Guestrin *et al.*, 2002a], motivated by the fact that in many cases this factorization enables an efficient maximization over joint actions. This has been highly influential, and similar approaches have been adopted in the context of Dec-POMDPs [Nair *et al.*, 2005; Varakantham *et al.*, 2007; Oliehoek *et al.*, 2008b], MPOMDPs [Amato and Oliehoek, 2015], multiagent RL [Guestrin *et al.*, 2002b; Kok and Vlassis, 2006; Kuyer *et al.*, 2008], and recent deep variants [Sunehag *et al.*, 2018; Rashid *et al.*, 2018].

Transfer Planning Factored value functions are a particular instantiation of linear function approximation [Guestrin *et al.*, 2003]. As such, most aforementioned approaches employ regression to find components Q^e that minimize prediction error. A different approach is taken in what I call *transfer planning (TP)* [Oliehoek *et al.*, 2013b]. The basic insight is that in order to derive good policies from Q-functions, it is more important that the relative values of the different joint actions are preserved, than having an (absolute) minimal prediction error. In problems with sufficient 'spatial' structure it may make sense to base the components Q^e on local abstractions of the problem, rather than using regression.

In particular, TP specifies $|\mathcal{E}|$ *source problems,* each of which is used to compute one component Q^e. These are local abstractions and only contain a small number of agents and hence are (relatively) easy to solve. The resulting Q^e are then *transferred* to the larger target task: since the subsets of agents may overlap, a 'plan repair' phase is needed to extract non-contradictory policies for each agent (for Dec-POMDPs) [Oliehoek *et al.*, 2013b], or message passing is needed for coordination (in the MMDP case) [Van der Pol and Oliehoek, 2016]. TP is agnostic to the method used to solve the source problems; for instance *deep Q-learning (DQN)* [Mnih *et al.*, 2013] was used to compute components Q^e in an application of TP to the problem of coordinating traffic lights [Van der Pol and Oliehoek, 2016].

Mixtures of Experts In a sense the source problems in TP can be seen as experts that make a prediction about the local value Q^e based on the local joint action a_e. However, not in all cases local reward components will be available.

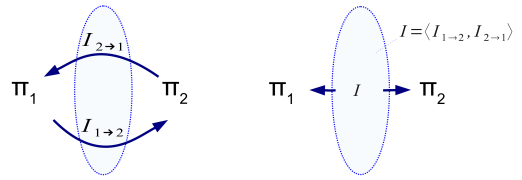

Figure 2: Influence-based abstraction (left) and search (right).

In such cases, one can still consider 'local experts' (which correspond to subsets of agents) that make predictions of the *total* value Q based on a_e. This allows the components e to learn in isolation, making it feasible to exploit factorization within *Monte Carlo tree search (MCTS)* [Amato and Oliehoek, 2015]. When the agents *do* have access to localized rewards this can bring further benefits [Pfrommer, 2016].

A further generalization of this idea is given by [Irissappane *et al.*, 2016], who scale up solution methods for complex POMDPs (that reason about many sellers and advisors) by considering multiple abstractions of those problems (which contain random subsets of sellers and advisors) and considering each of these random POMDPs as an expert. In general, the idea of using multiple experts has made a profound impact on machine learning, but the exploration of how these techniques can affect ILDM has only just begun.

Subjective Approximation The above techniques give a looser interpretation to factorization and become more like abstraction. Going even further, we can completely move to a subjective perspective: reasoning from the perspective of a single agent and the part of the state space it cares about. Such a subjective perspective can lead to good behavior as long as the protagonist agent can predict the actions of the other agents accurately enough. The *interactive POMDP* [Gmytrasiewicz and Doshi, 2005] gives an elegant solution by performing "k-level reasoning" about the other agents, but is computationally expensive.

Alternatives try to find leverage by exploiting *anonymity* [Varakantham *et al.*, 2014; Robbel *et al.*, 2016]. For instance, for a robot tasked with cleaning a warehouse, only *which* dirty spots will be cleaned by other agents is relevant, the identities of those other agents is not [Claes *et al.*, 2015]. Also, domain-specific heuristics can be used to predict teammates. For instance, in such robotic warehousing tasks one can use *decentralized MCTS* to facilitate fine-grained reasoning about movement and tasks appearance at various locations, while using heuristics developed in the robotics community to predict the teammates [Claes *et al.*, 2017].

3.3 Influence-based Abstraction & Search

Since much progress has been made by building on ideas of factorization and abstraction, I have worked on providing a deeper understanding of such abstractions, by characterizing *lossless* abstractions, and how they can facilitate coordination between agents.

In particular, I have contributed to the notion of *influence-based abstraction* [Oliehoek *et al.*, 2012], which tries to boil down interaction to its essentials: interaction is a two-way flow of *influence* and the influence, say of agent 1's policy on agent 2 ($I_{1\to2}$ in Fig. 2), is an abstract compressed form of all the information that agent 2 needs in order to compute its best response. In other words: many π_1 may have the same influence $I_{1\to2}$ and thus lead to the same best-response π_2.

Such ideas have been used to more efficiently search for joint policies in special cases of Dec-POMDPs, e.g, [Becker *et al.*, 2003; Witwicki and Durfee, 2010]. The main idea is that the space of *joint influences* can be much smaller than the space of joint policies, therefore significant speed-ups are possible by searching the former [Witwicki *et al.*, 2012]. In more complex settings, computing the influence points themselves is intractable. However, it is computationally affordable to make *optimistic* assumptions on what the influence could be [Oliehoek *et al.*, 2015]. Such an approach can be used to compute factored upper bounds on the optimal value functions of problems, which in turn can help interpreting the quality of heuristic solutions with hundreds of agents. For instance, such an analysis showed that the solution provided by Dec-POMDP transfer planning for the 50-agent Aloha benchmark is essentially optimal.

3.4 Learning as Planning

Most RL methods are based on the theory of MDPs, most RL problems, however, are partially observable. While in some cases one can get away with making the Markov assumption by using the last k observations, this in general is not the case. How to deal with exploration in a principled fashion in such PORL problems is still an open question. One appealing approach is given by the Bayes Adaptive POMDP framework [Ross *et al.*, 2011], which casts the learning problem, given a prior, as a planning problem. This means that advances in POMDP planning can be built upon [Katt *et al.*, 2017], and that it is possible to extend to POMDP-based multiagent settings [Oliehoek and Amato, 2014a; Amato and Oliehoek, 2015]. There are many challenges, however, in making such approaches more scalable.

4 Challenges

Providing scalability is a central question to all ILDM. In the last few years encouraging results have been obtained by deep multiagent reinforcement learning (MARL), and more progress is to be expected. Here I list just a small selection of important other challenges.

Truly Decentralized RL MARL is progressing at an amazing pace, but many approaches consider an offline training phase [Foerster *et al.*, 2018]. Other methods, such as independent DQN agents in theory could be applied on-line, but would be completely impractical due to their sample complexity. Moreover, it is often not clear what exactly is the right way to phrase these learning problems. For instance, policy gradient methods can readily be applied to Dec-POMDPs [Peshkin *et al.*, 2000], but require all of the agents to observe the rewards or return of the entire system, which might be difficult to realize in practice.

Learning Models in Interactive Decision Making Related to this, one of the grand challenges is to come up with methods to learn models of the behavior of other

agents, e.g., [Oliehoek and Amato, 2014a; Panella and Gmytrasiewicz, 2014], and indeed humans. To make substantial progress on this front, I suspect that we will need benchmark tasks and shareable models for them that can play a role similar to pre-trained imagenet networks.

Understanding Interactive Learning Finally, one of the current hot topics in machine learning revolves all around interactive learning in the form of competing networks (i.e., 'GANs') [Goodfellow *et al.*, 2014]. While the empirical progress and results have been astounding, theoretical understanding has been lagging [Arora *et al.*, 2017]. In particular, I have worked on the question of how such competing networks can avoid getting stuck in 'local Nash equilibria' [Oliehoek *et al.*, 2017]. More in general, I think that insights from ILDM, including both multiagent planning and learning, will have a big role to play in the development of machine learning in the future.

Acknowledgments

Thanks to Mathijs de Weerdt, Matthijs Spaan, Rahul Savani, Jacopo Castellini, James Butterworth for comments on a draft of this paper. I am grateful to all my collaborators and mentors—including (but certainly not limited to): Nikos Vlassis, Matthijs Spaan, Shimon Whiteson, Leslie Kaelbling, Karl Tuyls, and Rahul Savani—as well as funding agencies that have made my work possible: Dutch Ministry of Economic Affairs (BSIK), AFOSR (MURI), NWO (CATCH, VENI). I am currently funded by an EPSRC First Grant EP/R001227/1, and ERC Starting Grant #758824—INFLUENCE.

References

[Amato and Oliehoek, 2015] Christopher Amato and Frans A. Oliehoek. Scalable planning and learning for multiagent POMDPs. In *AAAI*, 2015.

[Arora *et al.*, 2017] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets. In *ICML*, 2017.

[Becker *et al.*, 2003] Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V. Goldman. Transition-independent decentralized Markov decision processes. In *AAMAS*, 2003.

[Bellman, 1957] Richard Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5), 1957.

[Bernstein *et al.*, 2000] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of Markov decision processes. In *UAI*, 2000.

[Bertsekas, 2007] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, 3rd edition, 2007.

[Boutilier *et al.*, 1999] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *JAIR*, 11, 1999.

[Boutilier, 1996] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Theoretical Aspects of Rationality and Knowledge*, 1996.

[Claes *et al.*, 2015] Daniel Claes, Philipp Robbel, Frans A. Oliehoek, Daniel Hennes, Karl Tuyls, and Wiebe Van der Hoek. Effective approximations for multi-robot coordination in spatially distributed tasks. In *AAMAS*, 2015.

[Claes *et al.*, 2017] Daniel Claes, Frans A. Oliehoek, Hendrik Baier, and Karl Tuyls. Decentralised online planning for multi-robot warehouse commisioning. In *AAMAS*, 2017.

[Dibangoye *et al.*, 2013] Jilles S. Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. Optimally solving Dec-POMDPs as continuous-state MDPs. In *IJCAI*, 2013.

[Foerster *et al.*, 2018] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI*, 2018.

[Gmytrasiewicz and Doshi, 2005] Piotr J. Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. *JAIR*, 24, 2005.

[Goldman and Zilberstein, 2003] Claudia V. Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS*, 2003.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS 27*, 2014.

[Guestrin *et al.*, 2002a] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. In *NIPS 14*, 2002.

[Guestrin *et al.*, 2002b] Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *ICML*, 2002.

[Guestrin *et al.*, 2003] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored MDPs. *JAIR*, 19, 2003.

[Hansen *et al.*, 2004] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, 2004.

[Irissappane *et al.*, 2016] Athirai Irissappane, Frans A. Oliehoek, and Jie Zhang. A scalable framework to choose sellers in e-marketplaces using POMDPs. In *AAAI*, 2016.

[Kaelbling *et al.*, 1996] Leslie Pack Kaelbling, Michael Littman, and Andrew Moore. Reinforcement learning: A survey. *JAIR*, 4:237–285, 1996.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *AIJ*, 101(1-2), 1998.

[Katt *et al.*, 2017] Sammie Katt, Frans A. Oliehoek, and Christopher Amato. Learning in POMDPs with Monte Carlo tree search. In *ICML*, 2017.

[Kok and Vlassis, 2006] Jelle R. Kok and Nikos Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *JMLR*, 7, 2006.

[Koller and Parr, 1999] Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured MDPs. In *IJCAI*, 1999.

[Kuyer *et al.*, 2008] Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. Multiagent reinforcement learning for urban traffic control using coordination graphs. In *ECML*, 2008.

[Littman, 1997] Michael L. Littman. Probabilistic propositional planning: Representations and complexity. In *AAAI*, 1997.

[MacDermed and Isbell, 2013] Liam C. MacDermed and Charles Isbell. Point based value iteration with optimal belief compression for Dec-POMDPs. In *NIPS*, 2013.

[Messias *et al.*, 2011] João V. Messias, Matthijs Spaan, and Pedro U. Lima. Efficient offline communication policies for factored multiagent POMDPs. In *NIPS 24*, 2011.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[Nair *et al.*, 2005] Ranjit Nair, Pradeep Varakantham, Milind Tambe, and Makoto Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *AAAI*, 2005.

[Nayyar *et al.*, 2011] Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Optimal control strategies in delayed sharing information structures. *IEEE Transactions on Automatic Control*, 56(7), 2011.

[Oliehoek and Amato, 2014a] Frans A. Oliehoek and Christopher Amato. Best response Bayesian reinforcement learning for multiagent systems with state uncertainty. In *Multi-Agent Sequential Decision Making in Uncertain Domains*, 2014.

[Oliehoek and Amato, 2014b] Frans A. Oliehoek and Christopher Amato. Dec-POMDPs as non-observable MDPs. IAS technical report IAS-UVA-14-01, Intelligent Systems Lab, University of Amsterdam, 2014.

[Oliehoek and Amato, 2016] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016.

[Oliehoek *et al.*, 2008a] Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *JAIR*, 32, 2008.

[Oliehoek *et al.*, 2008b] Frans A. Oliehoek, Matthijs T. J. Spaan, Shimon Whiteson, and Nikos Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *AAMAS*, 2008.

[Oliehoek *et al.*, 2012] Frans A. Oliehoek, Stefan Witwicki, and Leslie P. Kaelbling. Influence-based abstraction for multiagent systems. In *AAAI*, 2012.

[Oliehoek *et al.*, 2013a] Frans A. Oliehoek, Matthijs T. J. Spaan, Christopher Amato, and Shimon Whiteson. Incremental clustering and expansion for faster optimal planning in decentralized POMDPs. *JAIR*, 46, 2013.

[Oliehoek *et al.*, 2013b] Frans A. Oliehoek, Shimon Whiteson, and Matthijs T. J. Spaan. Approximate solutions for factored Dec-POMDPs with many agents. In *AAMAS*, 2013.

[Oliehoek *et al.*, 2015] Frans A. Oliehoek, Matthijs T. J. Spaan, and Stefan Witwicki. Factored upper bounds for multiagent planning problems under uncertainty with non-factored value functions. In *IJCAI*, 2015.

[Oliehoek *et al.*, 2017] Frans A. Oliehoek, Rahul Savani, José Gallego-Posada, Elise Van der Pol, Edwin D. De Jong, and Roderich Groß. GANGs: Generative Adversarial Network Games. *ArXiv e-prints*, 2017.

[Oliehoek, 2010] Frans A. Oliehoek. *Value-Based Planning for Teams of Agents in Stochastic Partially Observable Environments*. PhD thesis, University of Amsterdam, 2010.

[Oliehoek, 2013] Frans A. Oliehoek. Sufficient plan-time statistics for decentralized POMDPs. In *IJCAI*, 2013.

[Panella and Gmytrasiewicz, 2014] Alessandro Panella and Piotr Gmytrasiewicz. Learning policies of agents in partially observable domains using Bayesian nonparametric methods. In *Multi-Agent Sequential Decision Making in Uncertain Domains*, 2014.

[Papadimitriou and Tsitsiklis, 1987] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3), 1987.

[Peshkin *et al.*, 2000] Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie P. Kaelbling. Learning to cooperate via policy search. In *UAI*, 2000.

[Pfrommer, 2016] Julius Pfrommer. Graphical partially observable monte-carlo planning. In *Learning, Inference and Control of Multi-Agent Systems*, 2016.

[Puterman, 1994] Martin L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[Pynadath and Tambe, 2002] David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *JAIR*, 16:389–423, 2002.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *ArXiv e-prints*, 2018.

[Robbel *et al.*, 2016] Philipp Robbel, Frans A. Oliehoek, and Mykel J. Kochenderfer. Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs. In *AAAI*, 2016.

[Ross *et al.*, 2011] Stéphane Ross, Joelle Pineau, Brahim Chaibdraa, and Pierre Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *JMLR*, 12, 2011.

[Scharpff *et al.*, 2016] Joris Scharpff, Diederik M. Roijers, Frans A. Oliehoek, Matthijs T. J. Spaan, and Mathijs de Weerdt. Solving transition-independent multi-agent MDPs with sparse interactions. In *AAAI*, 2016.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning. In *AAMAS*, 2018.

[Van der Pol and Oliehoek, 2016] Elise Van der Pol and Frans A. Oliehoek. Coordinated deep reinforcement learners for traffic light control. In *Learning, Inference and Control of Multi-Agent Systems*, 2016.

[Varakantham *et al.*, 2007] Pradeep Varakantham, Janusz Marecki, Yuichi Yabu, Milind Tambe, and Makoto Yokoo. Letting loose a SPIDER on a network of POMDPs: Generating quality guaranteed policies. In *AAMAS*, 2007.

[Varakantham *et al.*, 2014] Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. Decentralized stochastic planning with anonymity in interactions. In *AAAI*, 2014.

[Wiggers *et al.*, 2016] Auke J. Wiggers, Frans A. Oliehoek, and Diederik M. Roijers. Structure in the value function of two-player zero-sum games of incomplete information. In *ECAI*, 2016.

[Witwicki and Durfee, 2010] Stefan J. Witwicki and Edmund H. Durfee. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *ICAPS*, 2010.

[Witwicki *et al.*, 2012] Stefan Witwicki, Frans A. Oliehoek, and Leslie P. Kaelbling. Heuristic search of multiagent influence space. In *AAMAS*, 2012.