Communication in Collaborative Multi-Agent Reinforcement Learning

Élie Michel
Department of Computer Science
École normale supérieure
Paris, France
elie.michel@ens.fr

Abstract

While Reinforcement Learning (RL) studies how a single agent optimizes its reward when interacting with a stationary environment, we review which hypothesis of this model can be relaxed and how far. We show that a very flexible framework to open RL is Multi-Agent Systems and Stochastic Game and then focus on the notion of communication within collaborative setups.

Keywords Collaborative Reinforcement Learning, Game Theory, Multi-Agent Systems, Decentralized Decision Making, Distributed Control.

This report is written as part of the validation process of the MVA lecture on Reinforcement Learning by Alessandro Lazaric[1].

1 Introduction

Reinforcement Learning (RL) [SB98] studies how an agent interacting with an environment through a given set of action can determine a *good* policy with respect to a notion of *reward*. This trial-and-error approach to decision-making has been applied to a wide variety of problems ranging from robotics [KBP13] to economics [TK02] with impressive results, especially in the field of games [SHM+16].

We've studied during the lecture the application of RL to the case of an unique agent interacting with a stationary environment. This problem is well-studied, so we wanted to open up the reflexion to less explored horizons and see how far the paradigm of RL can be generalized to other setups.

In Section 2, we presents different possible axes of generalization of the RL problem and explain why we focus on Multi-Agent RL (MARL). Then, Section 3 explicits the theoretical setup for MARL. Section 4 gives an overview of the many challenges specific to the multi-agent aspect and narrows again the scope of this study to Collaborative MARL. Within this perspective, Section 5 explores the notion of communication inherent to collaboration and its different aspects. Section 6 gives some actual application cases and finally Section 7 concludes this study.

2 Generalizing Reinforcement Learning

Different axes of generalization of the traditional RL problem (Figure 1) can be explored. We present in this section the possibility of relaxing the stationarity of the agent policy, the stationarity of the the environment and the uniqueness of the agent.

[1] http://researchers.lille.inria.fr/~lazaric/Webpage/MVA-RL_Course16.html

XXth Conference on Neural Information Processing Systems (NIPS 20xx), Paris, France – 2017.

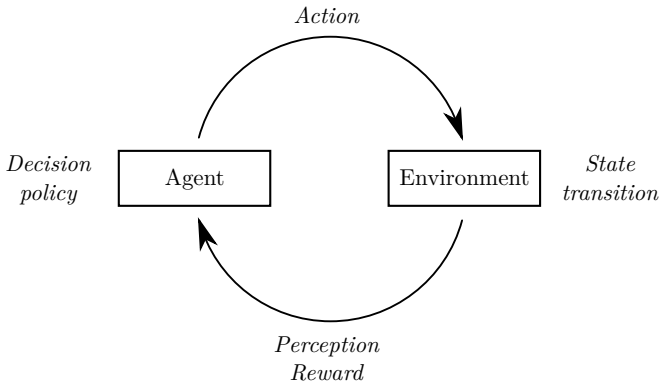

Figure 1: Traditional scheme of Reinforcement Learning. The usual RL task consists in determining a decision policy. Other problems are sometimes studied, like inferring the reward (Inverse RL) or the perception (in cognitive science).

2.1 Non-stationary policies

The stationarity of the agent policy becomes an issue as soon as the agent needs to *remember* what happened in the past to take a decision.

This is required when the agent must observe the inertia of elements of its environment. For instance, the velocity of the ball in a Pong game cannot be known by looking at one time frame of the game screen only. More generally, it is very common in robotics [KBP13] to get a reward based on the position while acting on derivatives only (through force and torque).

The easiest way solve this case is to extend the environment state to a fixed short term history, like the last 2 or 3 observed states of the agent's trajectory, in order to get the local dynamic of a system. This solution works for any need for short term memory but exponentially complexes the problem.

Another way to look at this problem is to consider that the actual state of the game consolidates all the orders of derivative describing its dynamic. Actually, the environment would not be considered as stationary otherwise. For instance, modeling the Pong game requires to store at any time the velocity of the ball, so it is actually part of the environment state. Since the agent cannot see this information at a given time, this is a case *partial observation* (Section 7 of [KLM96]).

Whether it is because the environment has a very complex dynamics or the agent a too partial observation, the short term history might not be enough for the agent to decide the right action. Non stationary policies can hence be used. One can see them as an augmentation of the environment state space by a *learned agent-state space* comparable to arbitrary memory. This learned space compresses elements of history that the agent might need in a potentially continuous form, as in Recurrent Neural Networks, and can foster long term features as with Long-Short Term Memory [HS97].

Online learning, like in multi-armed bandits problems, can also be considered as a case of non-stationary policy. In this case, run time is also train time, so the policy is always changing.

2.2 Non-stationary environment

Unless we are doing online learning, non-stationary policies are theoretically relevant only in non-stationary environments. Partially observe a stationary environments is actually equivalent to fully observe a non-stationary environments. Agent may have access to a model-based knowledge of how the environment evolves, like in physically grounded dynamic environments.

Another property of the environment as presented in Figure 1 that can change in time is the reward function, e.g. the target of the agent (Adaptive Learning). The evolution of the perception process is

less explored, as observation is in a majority of cases a total snapshot of the environment state, but this could be seen as kind of visual glare effect.

2.3 Multiple agents

Independently from RL, Multi-Agent Systems, also called Agent-Based Model in social sciences [HCSB11], have been studied a lot. Learning has been hence naturally introduced into this domain as a generalization of RL, giving birth to Multi-Agent Reinforcement Learning (MARL).

From the point of view of a given agent, other agents are part of the environment. Since other agents are learning as well, this problem can be compared to RL in non-stationary environment. But MARL considers all the agent, which introduces new issues like the notions of local and global goals.

As pointed out in [BBDS08], a multi-agent setup can also be a way of looking in a different perspective at a centralized decision taking problem that could actually be modeled as a traditional RL but features some local pattern.

The case of two adversarial agents with opposite goals in a stateless environment, commonly called *zero-sum game*, has been studied during the lecture, but the field of MARL is actually much wider, as we'll see in Section 4. Since non-stationary environments can be considered as particular cases of MARL where the non-stationarity is modeled by an extra agent, we focus bellow on the MARL modeling.

3 Formalism for MARL

3.1 Markov Decision Process

We expressed the environment for traditional RL as a Markov Decision Process (MDP). An MDP is a tuple $(\mathcal{S}, \mathcal{A}, \tau, \rho)$ where $\mathcal{S}$ is environment's state space, $\mathcal{A}$ is the agent's action space, $\tau : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$[2] is the state transition probability function and $\rho : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function. An MDP is *finite* when $\mathcal{S}$ and $\mathcal{A}$ are finite sets.

For a time step t, we note $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}$ and $r_t \in \mathbb{R}$ respectively the state, the action taken and the reward obtained. The agent behavior is modeled by a *policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. When the policy is deterministic, we note it $\bar{\pi} : \mathcal{S} \to \mathcal{A}$. Similarly, a deterministic state transition function is noted $\bar{\tau} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$.

3.2 Stochastic Game

The generalization of MDPs applied to multi-agent setups [Lit94] is called Stochastic Game (SG) or Markov Game. An SG of n agent is a tuple $(\mathcal{S}, \mathcal{A}_1, \dots, \mathcal{A}_n, \tau, \rho_1, \dots, \rho_n)$ where the environment's state-space is still shared but each agent i has its own action set $\mathcal{A}_i$ and its own reward ρ_i.

Even if agents are symmetric, i.e. have qualitatively identical action spaces, it is important to distinct an action a as performed by agent i from the "same" action as performed by j because they don't act from the same "position"[3]

The state transition function is not agent-dependent because the environment state is what is shared among agents, but it takes the action of each agent as input, namely $\tau : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_n \times \mathcal{S} \to [0, 1]$. We note for convenience $\hat{\mathcal{A}} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$. In general, the reward functions also depend on the action taken by each agent, $\rho_i : \mathcal{S} \times \hat{\mathcal{A}} \times S \to \mathbb{R}$, but are different for different agents.

[2] Formally, τ should map $\mathcal{S} \times \mathcal{A}$ to probability distributions over $\mathcal{S}$. In the case of finite MDPs, it is enough to use this notation if we assume that for all s and a, $\sum_{s' \in \mathcal{S}} \tau(s, a, s') > 0$.

[3] Interestingly, the notion of *position* can be defined, beyond its geometrical meaning, as what makes the "same" action have a different impact on the environment.

Example Zero-sum game modeled by a SG with two agents in a stateless environment, i.e. $\mathcal{S} = \{s\}$ (so $\bar{\tau}(s, a_1, a_2) = s$) and $\bar{\rho}_1(s, a_1, a_2) = -\bar{\rho}_2(s, a_1, a_2)$. A stateless SG is sometimes called a *static* game.

A policy for an agent of a SG is a map $\pi_i : \mathcal{S} \times \mathcal{A}_i \to [0, 1]$ or $\bar{\pi}_i : \mathcal{S} \to \mathcal{A}_i$ if it is deterministic. Here again, we can define a joint version of the policy $\hat{\pi} : \mathcal{S} \times \hat{A} \to [0, 1]$.

3.3 Fully Cooperative Stochastic Game

It is important to note that if all agents share the same target $\rho_1 = \rho_2 = \cdots = \hat{\rho}$, the tuple $(\mathcal{S}, \hat{\mathcal{A}}, \tau, \hat{\rho})$ is a valid MDP and traditional RL could be applied to find a join policy $\hat{\pi}$. This is called a fully cooperative SG.

Nevertheless, there remains a capital difference between applying single-agent RL to the MDP formulation of the problem and applying MARL to the SG formulation: MARL can handle the case of *decentralized learning* in which each agent learns its part of the policy independently or with limited access to other agent's learning. Even when centralized learning is possible, it can be interesting to structure the problem as a SG than as a huge product MDP.

Furthermore, the equivalence between a fully cooperative SG and an MDP is lost as soon as we consider that agents have a partial and different observation. A Partially Observed MDP (POMDP) is *not* equivalent to a *decentralized* POMDP [DABC16] if we impose constraints on how agents can communicate. Even when an agent fully measures the environment state, it is possible that agents do not see each other's actions. This is also a case of partial observation which does not make sense in the MDP formalism.

Remark As well as the MDP model is used in other contexts the RL such as Inverse RL or behavioral studies in cognitive science, SG models are also used beyond the case of MARL. It can for instance be used for studying relation between the micro and the macro levels in economics and social sciences [Tro09].

4 Challenges of MARL

Dealing with multiple agents is very difficult in general and even with fixed policies emerging behaviors are really hard to predict. So research focuses on restricted classes of problems. An important one is the case of adversarial stateless or stage games, in which there is no global environment, only agent-agent interactions. The simplest examples are two players zero-sum or general-sum games, but the study of these systems is more generally called *Game Theory*. A list of other typical classes of MARL based problems is given in [BBDS08], along with general considerations about what questions are specific to MARL compared to RL. We review those questions in the remaining of this section.

4.1 Need for stochasticity

In a finite MDP, there exists a *deterministic* optimal policy (the greedy policy for Q-learning's Q^*). But this breaks in general as soon as we add a second agent. Let's consider the worker/inspector problem defined as follow:

Example There are two agents A (the worker) and B (the inspector) with respective action spaces $\mathcal{A}_A = \{W, R\}$ (for Work and Rest) and $\mathcal{A}_B = \{I, N\}$ (for Inspect or Not). Agent A is always rewarded with an income r unless it is inspected while resting. Agent B pays the income to A and gets a reward of p if A works. Additionally, it pays an inspection cost c when choosing to inspect A.

Possible deterministic policies for the worker are "always working" or "always resting" and for the inspector are "always inspecting" or "never inspecting". But it is easy to check that none of these possibilities is optimal, and one of the agents will always have an incentive to change its strategy. For instance, if the worker is never inspected it will stop working, but if the worker always rests,

the inspector will have interest to start inspecting, so that there is no possible equilibrium in *pure strategies* (i.e. deterministic).

This notion of stable policies is formalized in Game Theory as the *Nash equilibrium*, which generally is in *mixed strategies*, meaning that the stable policies are probabilistic. There might be zero, one or more Nash equilibria in a given problem.

4.2 Local and global goals

A well-known difficulty of MARL is the definition of the goal to optimize. There is a different reward for each agent, and there are many ways of consolidating them into a single goal. Agents may have contradictory, or at least correlated goals.

The problem is of course easier to solve for fully cooperative settings than for fully competitive ones, but there are mixed setting or even more subtle cases like agents which are collaborative but competing for a resource. Furthermore, there can even be a global goal not directly coming from agents, like in regulation design that intends to shape local agent reward in order to make emerging behavior match a given global goal.

To the typical exploration-exploitation trade-off, the presence of joint learning in MARL adds the *adaptation-stability trade-off*. Even in adversarial setups, agents might have interest to make their behavior predictable (*stability*) at the same time as they evolve with or against other agents' changes (*adaptation*). Without stability, a MARL algorithm would really struggle to converge.

4.3 Model-free and model-based

MARL also makes the definition of *model-free* and *model-based* setting more complex. An RL problem is called model-based when there is some domain-knowledge manually introduced about the underlying MDP, and model-free otherwise. But in MARL there are two very different aspects of the SG that can be modeled: the (stationary) environment and the (evolving) other agents.

The modeling of other agents, called *agent-awareness*, is reviewed more in details in Section 5.3. It introduces another interesting difference between collaborative MARL and the equivalent centralized RL problem presented in Section 3.3: the MARL problem can be agent-aware but model-free with respect to the environment itself. The equivalent RL problem loses this distinction and is model-based without specifying that the model is only coming for agent-agent interactions.

On another side of the spectrum, the problem may feature a full model for the environment and be hence so model-based that there is no nothing left to learn about the environment. This is called *Optimal Control* [KBP13]. With a single agent, this is thus no longer RL, but with multiple agents there is still the agent-agent interactions to learn. This usually involves collaborative agents and is used for instance in swarm robotics [BFBD13].

Among the diversity of MARL problems, we limit the scope of this study to fully collaborative agents as done in [PL05]. This suggests us to focus on the solutions for *communication* between agents.

5 Communication, teaching, or imitation

We define communication in the more general way as what enables agents to share information. This includes both active or passive communication. Active communication involves *showing* explicit messages while passive communication is about *observing* what other agents do. In both cases, it requires a minimum of agent-awareness.

Communication consists for the agents in "hacking" on a subset of their action space to start using it as a communication mean. [FAdFW16] even hard-codes a set of actions as being for communication only (no influence on immediate reward). Along with the action space $\mathcal{A}_i$, agents have a message

space $\mathcal{M}_i$ that is not an argument of ρ. The communicative action space stops being used for its consequence on the environment itself and is used for the consequence it has on other's perception.

Learning communication is difficult without constraint because agents need to agree on a *meaning* for signs, e.g. what state would cause an agent to send a given message. This is known as the symbol grounding problem [Har90]. In a model for language emergence, [VD07] presents a list of principle for symbol grounding inspired by the literature on children's language acquisition.

We first explicit why there is a need for communication, and show first means of coordination. We then focus on the different levels of agent awareness and finally present applications with deep reinforcement learning.

5.1 Need for communication

There are two main reasons to need communication. The first one is to *synchronize* agent actions, and the second one is to *complete perception* in the case of decentralized POMDPs.

Coordination-free methods exist, like Team-Q learning [Lit01] or Distributed Q-learning algorithm [LR00] that deal with collaborative MARL but assume that the argmax of Q in Q-Learning is unique, which is a big constraint. It is indeed very common to experience ties in practice because most problems have symmetry properties. See Example 2 from [BBDS08] for a nice illustration of the tie breaking problem.

Communication is also a workaround of the problem of incomplete perception of the full environment state or other agent actions. This is pointed out by [FAdFW16] (who forgets the synchronization aspect of communication, by the way).

5.2 Hardwired communication

Most simple attempts to break ties include "social" conventions and role assignment. This is not really communication because it is completely predetermined. For instance, agents are ordered by priority and fully deterministic so that each agent can guess what the previous on has chosed. There starts to be a communication process when prioritary agents are not especially deterministic but have the ability to tell their choice to next agents to chose. More complex communication include learned role assignments [PLL98].

Some methods make more assumptions on how agents can communicate and additively decompose the state-action function Q into local members $Q(x) = Q_1(s, a_1, a_2) + Q_2(s, a_2, a_3) + \dots$. The graph whose nodes are agent indices and an edge is present between i and j iff $Q_k(s, a_i, a_j)$ is part of the previous Q sum is called the *coordination graph*. It is usually part of the *a priori* model provided with the problem, but can also be learned [KHBV05]. The Q_k can then be locally optimized.

5.3 Observation and agent-awareness

Passive, or *indirect* communication like between requires the observation of other agents by modeling their potential behavior, which involves *a priori agent-awareness*.

Agent-awareness addresses a fundamental question of Multi-Agent Systems, which is peer recognition. Are the agents conscious of having the same *nature* as their pairs? Do the agent recognize the phenomena of their observed environment as being the action of other agents? The human brain has hard-coded mechanism recognizing in other people's action what it could have performed itself through Human *Mirror Neuron System* [RFDC09].

Agent-awareness ranges from the simple fact of knowing that the non-stationarity of the observation comes from the existence of other agents, potentially along with a static model of how they work, to the precise dynamic modeling of other agents. An example of the latter for collaborative systems is Joint Action Learners [CB98a], but this is also applied in adversarial setups as *opponent modeling* [CM95].

5.4 Communication in Deep Reinforcement Learning

As repeated in [BBDS08], an important concern of the research in MARL is about the scalability of the algorithm. Indeed, the computational resources are sometimes already a problem with a single agent, so it is even more critical with many agents.

Single-agent RL deals with huge or continuous state spaces using Deep Q-Networks to represent the state-action function Q in a compressed way. This is sometimes called Deep Reinforcement Learning. This approach can be applied to MARL, either on a small number of agents [TMK+15] or on many agents for traffic light regulation for instance [vdP16].

And it has also been applied recently to the process of learning communication by [FAdFW16] and [SSF16]. Opponent Modeling is also reaching Deep Reinforcement Learning [HBGKDI16].

[FAdFW16] proposes to learn communication between multiple fully cooperative agents in a partially observed environment as a neural network. It tries two settings, one with and one without back-propagation through the communications (*Differentiable Inter-Agent Learning*), which is interpreted as communication feedback, as human beings do, like approbation gestures, etc. [SSF16] presents a very similar approach.

In both cases, Deep Q-learning is used for communication but the learning is centralized. [FAdFW16] notes though that the centralization is at learn time while run time has a decentralized execution. But without decentralization at learn time, the problem can be interpreted as an MDP problem. The interesting point of this way of looking at it is that, since communication network is shared among all agents, the overall network can be seen as a Convolutional Neural Network.

6 Applications

Now that we reviewed the main challenges of Collaborative MARL, we present some possible applications. [BBDS08] interestingly notes that in some of them the agent learn to *serve* a resource that they own, while in other cases the agents learn when to *exploit* a passive resource.

For instance, in urban traffic control problems [SCGC08] [KG14] [vdP16] the agents are traffic lights that are sending to their neighbors a flow of cars.

On another hand, multi-robot applications [Mat97] must deal with space sharing because a given position in space can generally be occupied by at most one agent. We can cite again swarm robotics [BFBD13] and its predecessor, the boids [Rey87] which includes agent avoiding policies, but also the agreement problem in self reconfiguring robots [VKR09].

Quantitative social sciences is also interested in MARL problem, either for economy or with initiatives such as the NewTies project [GdBB+06], which had the ambition of creating a complete toy virtual society and study how agent learn to interact.

There are also interesting problems with small numbers of agents, like elevator scheduling [CB98b] or distributed control like collaborative pendulum stabilization.

Finally, resource sharing like load balancing of problems in networks of sensors like packet routing like cognitive radio networks [BLJ13] are very practical applications of MARL.

7 Conclusion

We have followed in this document a direction of opening of the RL paradigm to a wider range of problems. We have seen that a particularly relevant track to explore is the multi-agent setting and reviewed what new challenges it raises, to finally focus on communication process as part of collaborative systems. This is an active field of research, and though there are interesting results there remains a lot to find. Deep MARL started addressing the problem of computational resource but is at this time still limited to centralized learning, which is a quite restrictive setup.

References

[BBDS08] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*, 2008.

[BFBD13] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[BLJ13] Mario Bkassiny, Yang Li, and Sudharman K Jayaweera. A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys & Tutorials*, 15(3):1136–1159, 2013.

[CB98a] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, (s 746):752, 1998.

[CB98b] Robert H Crites and Andrew G Barto. Elevator group control using multiple reinforcement learning agents. *Machine Learning*, 33(2-3):235–262, 1998.

[CM95] David Carmel and Shaul Markovitch. Opponent modeling in multi-agent systems. In *International Joint Conference on Artificial Intelligence*, pages 40–52. Springer, 1995.

[DABC16] Jilles Steeve Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. Optimally solving dec-pomdps as continuous-state mdps. *Journal of Artificial Intelligence Research*, 55:443–497, 2016.

[FAdFW16] Jakob N Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

[GdBB+06] Nigel Gilbert, Matthijs den Besten, Akos Bontovics, Bart GW Craenen, Federico Divina, et al. Emerging artificial societies through learning. *Journal of Artificial Societies and Social Simulation*, 9(2), 2006.

[Har90] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

[HBGKDI16] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1804–1813, 2016.

[HCSB11] Alison J Heppenstall, Andrew T Crooks, Linda M See, and Michael Batty. *Agent-based models of geographical systems*. Springer Science & Business Media, 2011.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[KBP13] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, page 0278364913495721, 2013.

[KG14] Mohamed A Khamis and Walid Gomaa. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, 29:134–151, 2014.

[KHBV05] Jelle R Kok, Eter Jan Hoen, Bram Bakker, and Nikos Vlassis. Utile coordination: Learning interdependencies among cooperative agents. In *EEE Symp. on Computational Intelligence and Games, Colchester, Essex*, pages 29–36, 2005.

[KLM96] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[Lit94] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157, pages 157–163, 1994.

[Lit01] Michael L Littman. Value-function reinforcement learning in markov games. *Cognitive Systems Research*, 2(1):55–66, 2001.

[LR00] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

[Mat97] Maja J Matarić. Reinforcement learning in the multi-robot domain. In *Robot colonies*, pages 73–83. Springer, 1997.

[PL05] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434, 2005.

[PLL98] MV Nagendra Prasad, Victor R Lesser, and Susan E Lander. Learning organizational roles for negotiated search in a multiagent system. *International Journal of Human-Computer Studies*, 48(1):51–67, 1998.

[Rey87] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34, 1987.

[RFDC09] Giacomo Rizzolatti, Maddalena Fabbri-Destro, and Luigi Cattaneo. Mirror neurons and their clinical relevance. *Nature Clinical Practice Neurology*, 5(1):24–34, 2009.

[SB98] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[SCGC08] As' ad Salkham, Raymond Cunningham, Anurag Garg, and Vinny Cahill. A collaborative reinforcement learning approach to urban traffic control optimization. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 560–566. IEEE Computer Society, 2008.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[SSF16] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. *arXiv preprint arXiv:1605.07736*, 2016.

[TK02] Gerald Tesauro and Jeffrey O Kephart. Pricing in agent economies using multi-agent q-learning. *Autonomous Agents and Multi-Agent Systems*, 5(3):289–304, 2002.

[TMK+15] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *arXiv preprint arXiv:1511.08779*, 2015.

[Tro09] Klaus G Troitzsch. Perspectives and challenges of agent-based simulation as a tool for economics and other social sciences. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 35–42. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[VD07] Paul Vogt and Federico Divina. Social symbol grounding and language evolution. *Interaction Studies*, 8(1):31–52, 2007.

[vdP16] Elise van der Pol. Deep reinforcement learning for coordination in traffic light control. Master's thesis, Master's Thesis. University of Amsterdam, 2016.

[VKR09] Paulina Varshavskaya, Leslie Pack Kaelbling, and Daniela Rus. Efficient distributed reinforcement learning through agreement. In *Distributed Autonomous Robotic Systems 8*, pages 367–378. Springer, 2009.