Multiagent Learning: Basics, Challenges, and Prospects

Karl Tuyls, Gerhard Weiss

■ *Multiagent systems (MAS) are widely accepted as an important method for solving problems of a distributed nature. A key to the success of MAS is efficient and effective multiagent learning (MAL). The past 25 years have seen a great interest and tremendous progress in the field of MAL. This article introduces and overviews this field by presenting its fundamentals, sketching its historical development, and describing some key algorithms for MAL. Moreover, main challenges that the field is facing today are identified.*

Multiagent systems (MAS) are distributed systems of independent actors, called agents, that cooperate or compete to achieve a certain objective. These agents may be computer programs, robots, or even humans. Many technological challenges of today's society involve complex dynamics and high degrees of uncertainty and are characterized by the fact that they are situated in the real physical world and consequently have an inherently distributed nature. Examples include automated driving, distributed traffic light control, robot soccer, and coordination of large swarms of robots. Because of their complexity it becomes impossible to engineer optimal solutions by hand, that is, defining beforehand which behavior is optimal in which situation. Moreover, agents need to take into account not only changing circumstances but possibly also the interactions with and behavior of other agents present in the system. Consequently, agents should be able to learn to behave optimally from experience, the environment, and interaction with other agents. Multiagent learning (MAL) is the field that integrates machine-learning techniques in MAS and studies the design of algorithms to create such adaptive agents.

The most commonly studied technique for MAL is reinforcement learning (RL). Single-agent RL is usually described within the framework of Markov decision processes (MDPs). Some standalone RL algorithms (for example, Q-learning) are guaranteed to converge to the optimal strategy, as long as the environment the agent is experiencing is *Markovian* and the agent is allowed to try out sufficient actions. Although MDPs provide a solid mathematical framework for single-agent learning, they do not offer the same theoretical grounding for MAL. When multiple *adaptive* agents interact with each other, the reward an agent receives may depend on the actions taken by those other agents, rendering the Markov property invalid since the environment is no longer stationary. Each agent is therefore faced

Copyright © 2012, Association for the Advancement of Artificial Intelligence. All rights reserved. ISSN 0738-4602

with a moving-target problem: what needs to be learned by an agent depends on and changes with what has been learned by the respective other agents. Therefore extensions of the MDP framework have been considered such as Markov games and joint action learners (Littman 1994, Claus and Boutilier 1998). In these approaches learning happens in the product space of the set of states and action sets of the different agents. Such approaches experience difficulties with large state-action spaces when the number of agents, states, and actions increase. Furthermore, a shared joint action space approach is not always applicable; for instance, in situations with incomplete information it is not necessarily possible to observe what actions the other agents take. Today, tackling complex real-world problems is the holy grail of MAL research, that is, how to efficiently handle many states, many agents, and continuous strategy spaces. For this purpose multiagent learning should rely on a scalable theory, that is, a foundational framework within which MAL algorithms can be designed for both small and large-scale agent systems.

This article reviews the current state of affairs in the field of MAL and is intended to offer a bird's-eye perspective on the field by reflecting on the nature and the foundations of MAL. Rather than surveying, the purpose of this article is to introduce the basics of MAL, to identify the main challenges the MAL field needs to tackle, to stimulate discussion of the foundations of MAL, and to identify promising future research directions in which we believe the field needs to develop. For the sake of completeness, there are several recent articles available that do an excellent job on surveying MAL; see, for example, Busoniu, Babuska, and De Schutter (2008); Panait and Luke (2005); Yang and Gu (2009). Good sources for tracking the field are the conferences on autonomous agents and multiagent systems (AAMAS), machine-learning conferences (the International Conference on Machine Learning — ICML, the European Conference on Machine Learning — ECML) and corresponding journals (*Journal of Autonomous Agents and Multi-Agent Systems*, *Journal of Machine Learning Research*, *Machine Learning*). There are also contributions to conferences like the International Joint Conference on Artificial Intelligence (IJCAI), the AAAI Artificial Intelligence Conference (AAAI), the European Coordinating Committee for Artificial Intelligence Conference (ECCAI), the Genetic and Evolutionary Computation Conference (GECCO), and Artificial Life.

In this article, we begin by taking a closer look at the fundamentals of MAL. More specifically, we first describe the basic setup and then delve deeper into the nature of MAL. We continue by describing some of the milestones of the field by sketching a historical perspective on MAL and discussing some of its landmark algorithms. Then we investigate what the current active topics are and how these could be extended using influences from other domains. Finally, we conclude.

Fundamentals of Multiagent Learning

In this section we introduce the basic formal setting of multiagent learning, necessary to understand the remainder of the article. Specifically, we briefly sketch stochastic or Markov games the most commonly used framework to describe the multiagent learning setting. After discussing the standard MAL setting we delve deeper into the nature of multiagent learning, investigate its complexity, and look into classifications and characterizations of MAL research.

The Basic Setup

Reinforcement learning finds its roots in animal learning. It is well known that we can teach a dog to respond in a desired way by rewarding and punishing it appropriately. For example we can train it to search for victims of criminal acts or of natural disasters such as earthquakes. Dogs can be trained for this by stimulating them to search for hidden dummy items with a specific scent and rewarding them each time they locate the object. Based on this reward or external feedback signal the dog adapts to the desired behavior and gradually learns to search for items or victims on command. Reinforcement learning is based on the observation that rewarding desirable behavior and punishing undesirable behavior leads to behavioral change. More generally, the objective of a reinforcement learner is to discover a policy, that is, a mapping from situations to actions, so as to maximize the reinforcement it receives. The reinforcement is a scalar value that is usually negative to express a punishment and positive to indicate a reward.

Markov Decision Processes

Most single-agent RL research is based on the framework of Markov decision processes (MDPs) (Puterman 1994). MDPs are sequential decision-making problems for fully observable worlds. MDPs are defined by a tuple (S, A, T, R), where S is a finite set of states and A is a finite set of actions available to an agent. An MDP has the *Markov property*: the future dynamics, transitions, and rewards fully depend on the current state, that is, an action a in state $s \in S$ results in state s' based on a transition function $T : S \times A \times S \to [0, 1]$. The probability for ending up in state s' after doing action a in state s is denoted as $T(s, a, s')$. The reward function $R : S \to \Re$, returns the reward $R(s, a)$ after taking action a from state s.

The transition function T and reward function R

together are referred to as the model of the environment. The learning task in an MDP is to find a policy $\pi : S \rightarrow A$ for selecting actions with maximal expected (discounted) future reward. The quality of a policy is indicated by a *value function* V^{π}. The value $V^{\pi}(s)$ specifies the total amount of reward that an agent may expect to accumulate over the future, starting from state s and then following the policy π. In a discounted infinite horizon MDP, the expected cumulative reward (that is, the value function) is denoted as:

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s\right] \qquad (1)$$

A discount factor $\gamma \in [0, 1)$ is introduced to ensure that the rewards returned are bounded (finite) values. The variable γ determines the relevance of future rewards in the update.

The value for a given policy π, expressed by equation 1, can iteratively be computed by the *Bellman Equation* (Bellman 1957). One typically starts with arbitrary chosen value functions, and at each iteration for each state $s \in S$, the value functions are updated based on the immediate reward and the current estimates of V^{π}:

$$V_{t+1}^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_t^{\pi}(s') \qquad (2)$$

The goal of an MDP is to find the optimal policy, that is, the policy that receives the most reward. The optimal policy $\pi^*(s)$ is such that $V^{\pi^*}(s) \geq V^{\pi}(s)$ for all $s \in S$ and all policies π.

When the model of the environment is unknown (as is usual), reinforcement learning can be used to directly map states to actions. Q-learning (Watkins 1989) is the most famous example of model-free temporal difference learning algorithms. The updating of the Q-values of the state action pairs is given by:

$$Q(s,a) \rightarrow (1-\alpha)Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a')\right] \qquad (3)$$

where α is the learning rate, and γ the discount-rate.

Crucial for the entire learning process is how actions are selected, typically referred to as the *exploration-exploitation* dilemma. Given estimates of the values of each action, the question becomes how to select future actions. Through exploration the reinforcement learner discovers new actions and their potential value and uses this to improve its policy. Through exploitation the agent selects the best action available at that time instance, and as such maximizes the reward the agent receives. An important question then is how to balance exploration and exploitation.

One way to proceed is to behave *greedily* most of the time but once in a while select a random action to make sure the better actions are not missed in the long term. This approach is called the ϵ-*greedy* exploration method, in which the greedy option is chosen with high probability $1 - \epsilon$, and with a small probability ϵ a random action is played.

Another alternative is to use a "softmax" approach, or Boltzmann exploration, where the good actions have an exponentially higher probability of being selected and the degree of exploration is based on a *temperature* parameter τ. An action a_j is chosen with probability:

$$p_j = \frac{e^{\frac{Q(s,a_j)}{\tau}}}{\sum_i e^{\frac{Q(s,a_i)}{\tau}}} \qquad (4)$$

The selection of the temperature parameter is used to balance exploration and exploitation.

Markov Games

Once multiple agents are interacting through their learning processes, the basic MDP model is no longer sufficient. Markov or stochastic games generalize both repeated games and MDPs to the more general case of multiple states (repeated games are stateless) and multiple agents (basic MDPs consider only one agent). In each stage, the game is in a specific state featuring a particular payoff function and an admissible action set for each player. Players take actions simultaneously and hereafter receive an immediate payoff depending on their joint action. A transition function maps the joint action space to a probability distribution over all states, which in turn determines the probabilistic state change. Thus, similar to a Markov decision process, actions influence the state transitions. A formal definition of Markov games goes as follows (with some details omitted). A Markov game is a tuple (P, S, A, R, T) where

P is a set of n players;

S is a set of k states;

A is the set of joint actions, that is, $A = A_1 \times A_2 \ldots \times A_n$ with A_i being the finite set of actions available to player i;

$R : S \times \mathrm{A} \mapsto \Re^n$ is a payoff function, that is, a function that maps each joint action carried out by the agents in some state to an immediate real-valued payoff for each player (R is called payoff function); and

$T : S \times A \times S \mapsto [0, 1]$ is a transition probability function, that is, a function that gives the probability of transitioning from state s to state s' under the players' joint action a.

Markov games were first used as a framework for multiagent learning in Littman (1994). There exist other (more extensive) formal frameworks for multiagent learning such as for instance decentralized MDPs (dec-MDPs), and decentralized POMDPs (dec-POMDPs), in which it is no longer assumed that agents have perfect knowledge of the system state (Bernstein, Zilberstein, and Immerman 2000).

The Nature of MAL

In order to get insight into what multiagent learning is about, it is crucial to understand the true nature of MAL. The multiagent learning problem is the problem of an agent, situated in a stochastic game (or similar framework as the one described above), that needs to learn to behave optimally in the presence of other (learning) agents, facing the complexities of incomplete information, large state spaces, credit assignment, cooperative or competitive settings, and reward shaping (Wolpert and Tumer 2001; Tumer, Agogino, and Wolpert 2002; Agogino and Tumer 2008). Behaving optimally is usually defined in terms of game-theoretic solution concepts such as Nash equilibrium, Pareto optimality, and evolutionarily stable strategies. For an elaborate discussion on these concepts we refer to Busoniu, Babuska, and De Schutter (2008); Tuyls and Nowe (2005); Weibull (1996); and Vega-Redondo (2003). Informally, a Nash equilibrium can be described as follows: If there is a set of strategies for a game with the property that no player can increase the payoff by changing its strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute a Nash equilibrium. While we define the multiagent learning problem here as a learning problem with a number of complexity factors, not so often researchers tackle the multiagent learning problem in a situation combining all of these factors. There have been many attempts to characterize and refine the goals and purposes of MAL, resulting in various taxonomies and classifications of MAL techniques (see for example, Busoniu, Babuska, and De Schutter [2008] and ’t Hoen et al. [2006] for broader overviews and references to representative techniques). In what follows we overview the most common classifications.

Available classifications usually differ in the criteria they apply to characterize MAL. An example of such a criterion is the *type of task,* which leads to the prominent distinction between *cooperative learning* and *competitive learning*. In the case of cooperative learning, the agents have a joint task and their common learning goal is to improve or optimize task execution in terms of (for example) task completion time and quality. In other words, the agents have the same reward function and the agents’ learning goal is to maximize their utility as a group. In the case of competitive learning, the agents have conflicting tasks (so that not all of them can be completed, for example, due to resource limitations or because the agents’ goals are in direct opposition) and each agent’s learning goal is to ensure the best possible execution of its own task. In competitive learning scenarios the agents have individual reward functions and each of them is selfish in that it aims at maximizing its own utility even if this is only possible at the cost of the other agents and their individual utilities. As noted by Busoniu, Babuska, and De Schutter (2008) and ’t Hoen et al. (2006), the cooperative-competitive distinction is not sharp with regard to the behavior of agents: a cooperative agent may encounter a situation in which it has to behave temporarily in a selfish way (while all involved agents have the same goal and are willing to cooperate, they may want to achieve their common goal in different ways); and a competitive agent may encounter a situation in which a temporary coalition with its opponent is the best way to achieve its own goal. Several refinements of the type-of-task criterion have been proposed. For instance, in Panait and Luke (2005) cooperative learning has been further differentiated into team learning (also called coordination-free learning) and concurrent learning (also called coordination-based learning): the former assumes that a single agent identifies a set of appropriate behaviors for a team of agents, and the latter assumes that agents run multiple learning processes concurrently where each process concerns the improvement of an individual agent’s behavior or a (relatively independent) subtask of the agents’ joint task.

Another standard classification criterion for MAL is a learning agent’s degree of *awareness* of the other agents and their learning processes, resulting in characterizations that range from “fully unaware” to “fully aware.” As noted by Busoniu, Babuska, and De Schutter (2008), this criterion is strongly related to the agents’ learning goals: while some learning goals (for example, overall stability) may be achievable with no or little awareness of the other agents’ behavior, others (for example, behavioral adaptation to other agents) may only be achievable with high or full awareness. Other classification criteria include the degree of *homogeneity* of the agents’ learning algorithms, the homogeneity of the agents themselves, and the *prior knowledge* a learning agent has about the task, the environment and the other agents (for example, Busoniu, Babuška, and De Schutter [2008] and ’t Hoen et al. [2006]). Related to the criterion of prior knowledge is the criterion of *usage of models* of (for example) the task or the other agents’ strategies, which leads to the distinction of model-based and model-free MAL techniques, that is, having or learning a model of the environment dynamics and/or an opponent model predicting the behavior of other agents. For further considerations on model-based and model-free learning approaches see, for example, Shoham, Powers, and Grenager (2007) and Yang and Gu (2009) .

A different characterization of MAL, based on an examination of available MAL literature and formulated in terms of proposed research agendas, was introduced in Shoham, Powers, and Grenager (2007). In that article five possible goals of MAL research (and thus, indirectly, of five types of MAL techniques) are identified that are claimed to provide clear motivations and success criteria for MAL: computational; descriptive; normative; prescriptive, cooperative; or prescriptive, noncooperative.

In *computational* MAL techniques, learning algorithms are viewed as an iterative way to compute properties of a game. In *descriptive MAL techniques*, learning algorithms are used as a means to formally investigate learning by natural entities (humans, animals, organizations). In *normative* MAL techniques, learning algorithms give a means to determine which sets of learning rules are in equilibrium with one another.

In *prescriptive, cooperative* MAL techniques, learning algorithms describe how agents should learn in order to achieve distributed control of dynamic systems. In *prescriptive, noncooperative* MAL techniques, learning algorithms describe how agents should act to obtain high rewards. This classification is useful in that it offers a novel perspective on MAL that is complementary to the perspectives of the other classifications. Critical reflections of this classification can be for instance found in Stone (2007) and Tuyls and Parsons (2007).

Milestones

This section sketches main research developments in MAL over the past 20 to 25 years and elaborates on a number of algorithms that have been important and trend setting for the field. The intention behind this section is to give a useful overall picture of MAL, including its history and its state of the art, rather than being comprehensive.

A Historical Perspective

The development of the MAL field can roughly be divided into two periods, which we call the "startup" and "consolidation" periods. Both periods are discussed in more detail below.

The Startup Period

The startup period, from the late 1980s until about 2000, was characterized by a broad exploration of the concept of MAL and its possible realizations. The end of the 1980s were dominated by the first investigations of what was then called adaptive parallel computation inspired by nature. Techniques explored in these first multiagent learning contexts were early ant systems and flocking or herding behavior (Manderick and Moyson 1988; Colorni, Dorigo, and Maniezzo 1992; Banerjee 1992), evolutionary computation (Manderick and Spiessens 1989; Steels 1987, 1988; Paredis 1995), social learning (Boyd and Richerson 1985; Laland, Richerson, and Boyd 1993), neural networks (Pfeifer et al. 1989), and interactive and imitation learning (Galef 1988, Steels 1996). Many of these early results and techniques developed further within what is now known as the artificial life field. At that time the first artificial life conferences (such as A-life, PPSN — parallel problem solving from nature, and so on) were organized. Many techniques investigated within artificial life are still highly relevant for MAL but are currently not so often explored for this purpose. Moreover, recent results show there exist formal links between these techniques and popular MAL techniques such as reinforcement learning. A prime example is the formal link between coevolutionary algorithms and multiagent Q-learning established through the replicator equations from evolutionary game theory (EGT) (Panait, Tuyls, and Luke 2008), but also relations between swarm intelligence and reinforcement learning are being discovered (Lemmens and Tuyls 2009).

The first multiagent *reinforcement learning* efforts appeared soon with the work of Whitehead, Tan, and Littman (Whitehead 1991, Tan 1993, Littman 1994). Shortly after these publications the first dedicated workshops on MAL were organized and journal special issues appeared (for example, Huhns and Weiss volume 33(2–3) [1998]; Sen [1996]; Weiss [1996, 1997, 1998]; Weiss and Sen [1996]). At the end of this startup phase the first general understanding of the role of learning in multiagent settings emerged, for example, Stone and Veloso (2000) and the first textbooklike treatment of MAL became available (Sen and Weiss 1999). The insights and results gained in those years were the input for the second period, from about 2000 until today.

The Consolidation Period

While research during the first period was more like a breadth-first paradigmatic exploration, research conducted in the second phase was more like a depth-first exploration characterized by a focus on certain multiagent learning techniques (especially reinforcement learning in a game-theoretic context) and on theoretical foundations of MAL. Articles that give a good overview of MAL methods and techniques developed during this second phase are, for example, Busoniu, Babuška, and De Schutter (2008); 't Hoen et al. (2006), and Shoham, Powers, and Grenager (2007).

Next we describe a number of algorithms that we find exemplary state-of-the-art algorithms of both the startup and the consolidation period. As we cannot discuss all of them and be comprehensive in this article, we have chosen to describe one algorithm of the startup period (JAL) and two algorithms from the consolidation period in more detail. More precisely we describe Nash-Q, a direct follow-up algorithm of the startup period, and dis-

$$\begin{array}{c} \\ a_0 \\ a_1 \end{array} \begin{array}{c} \begin{array}{cc} b_0 & b_1 \end{array} \\ \begin{pmatrix} r_1 & r_2 \\ r_3 & r_4 \end{pmatrix} \end{array}$$

Figure 1. Example Repeated Matrix Game.

cuss the gradient ascent family of algorithms developed in the consolidation period. We certainly do not wish to give the impression that these algorithms are the only landmark algorithms.

State of the Art Algorithms of Both Periods

Joint action learning has been introduced in the context of cooperative repeated games; see Claus and Boutilier (1998). A joint action learner (JAL) is an agent that learns Q-values for joint actions in a cooperative repeated game, in contrast to independent learners that learn Q-values only for individual actions. This entails that such an agent stores and adapts Q-values for joint actions **a** with **a** a vector $\langle a_1, ..., a_n\rangle \in A_i \times ... \times A_n$ composed of the individual actions $a_i \in A_i$ of agent *i*. This implies that each agent can observe the actions of other agents.

Instead of carrying out Q-learning in the individual action space the JAL agent now learns in the joint action space. Since we consider stateless repeated games the update rule of Q-learning can be simplified to

$$Q(a) = Q(a) + \alpha(r - Q(a)) \tag{5}$$

In this stateless setting, we assume a Q-value, that is $Q(a)$, providing an estimate of the value of taking action *a*. At each time step a JAL agent *i* takes an action a_i belonging to joint action *a*. The sample $\langle a, r\rangle$ is the "experience" obtained by the agent: joint action *a* was performed resulting in reward *r*; for instance when the agents involved in the game illustrated in figure 1 play joint action $\langle a0$ $, b0\rangle$ they will receive reward $r_1 \cdot \alpha$ is the typical learning rate to control step sizes of the learning process. It is important to realize that a JAL agent is now learning values for all joint actions and no longer individual actions. For instance in the two-player two-action game example of figure 1 the joint action learner will learn Q-values for the tuples $\langle a_i , b_j\rangle$ with $i, j \in \{0, 1\}$ instead of for its individual actions a_i as an independent learner does.

Suppose that agent 1 (or the row player) has Q-values for all four joint actions, then the reward the agent can expect to accumulate will depend on the strategy adopted by the second (or column) player. Therefore a JAL agent will keep a model of the strategies of other agents *i* participating in the game such that he or she can compute the expected value of joint actions in order to select good subsequent actions balancing exploration and exploitation. A JAL then assumes that the other players *i* will choose actions in accordance with the model he keeps on the strategies of the other players. Such a model can be simply implemented through a fictitious play approach, in which one estimates the probability with which an agent will play a specific action based on the frequencies of the agent's past plays. In such a way expected values can be computed for the actions of a JAL based on the joint actions. For instance in the example we would have the following expected value *EV* for the first player's actions:

$$EV(a_i) = \sum_{b_j \in b_0, b_1} Q(b_j \cup \{a_i\}) Pr^1_{b_j} \tag{6}$$

with $Pr^1{}_{bj}$ the probability with which player 1 believes the other player will choose actions b_j implemented through a fictitious play approach. Using these EV values player 1 can now implement, for example, a Boltzmann exploration strategy for action selection.

Nash-Q Learning

Nash-Q, an algorithm introduced by Hu and Wellman (2000, 2003), aims to converge to a Nash equilibrium in general-sum stochastic games. In essence the algorithm extends the independent Q-learning algorithm to the multiagent case using the Markov game framework. The optimal Q-values in this algorithm are the values that constitute a policy or strategy for the different agents that are in Nash equilibrium. The Nash equilibrium serves as the solution concept the agents aim to reach by learning iteratively. To achieve this each Nash Q-learning agent maintains a model of other agents' Q-values and uses that information to update its own Q-values.

The Nash-Q learning algorithm also considers joint actions (such as JAL) but now in the context of stochastic games (containing multiple states). In an *n*-agent system, the Q-function for an agent becomes $Q(s, a_1, ..., a_n)$, rather than the single-agent Q-function, $Q(s, a)$. Given these assumptions Hu and Wellman define a Nash Q-value as the expected sum of discounted rewards when all agents follow specified Nash equilibrium strategies from the next period on. The algorithm uses the Q-values of the next state to update those of the current state. More precisely, the algorithm makes updates with future Nash equilibrium payoffs, whereas single-agent Q-learning updates are based on the agent's own maximum payoff. To be able to learn these Nash equilibrium payoffs, the agent must observe not only its own reward but also those of others (as was the case in the JAL algorithm).

The algorithm proceeds as follows. The learning agent, indexed by *i*, learns its Q-values by starting with arbitrary values at time 0. An option is to let $Q^{i}{}_{0}(s, a_1, ..., a_n) = 0$ for all $s \in S, a_1 \in A_1, ..., a_n \in A_n$. At each time *t*, agent *i* observes the current state, and takes its action. After that, it observes its own reward, actions taken by all other agents, rewards of others, and the new state s'. Having this information it then computes a Nash equilibrium $\pi^1 (s'), ..., \pi^n(s')$ for the stage game $(Q^1{}_t(s'), ..., Q^n{}_t(s'))$, and updates its Q-values according to:

$$Q^i_{t+1}(s, a_1, ..., a_n) = \\ (1 - \alpha_t)Q^i_t(s, a_1, ..., a_n) + \alpha_t[r^i_t + \beta NashQ^i_t(s')]$$

where $NashQ_i^t(s') = \pi^1(s') \dots \pi^n(s').\ Q^i{}_t(s')$

$NashQ_i^t(s')$ is agent *i*'s payoff in state s' for the selected equilibrium. In order to calculate the Nash equilibrium$(\pi^1 (s'), ..., \pi_n (s'))$, agent *i* needs to know $Q^1{}_t(s'), ..., Q^n{}_t(s')$. Since this information about other agents' Q-values is not available, this has to be learned as well. Since *i* can observe other agents' immediate rewards and previous actions it can use that information to learn the other agents' Q-functions as well.

The algorithm is guaranteed to converge to Nash equilibrium, given certain technical conditions hold. Littman tackled these restrictive conditions of Nash-Q and introduced *Friend or Foe Q-learning* (Littman 2001), which converges to Nash equilibrium with fewer restrictions than Nash-Q. For more details on Nash-Q we refer to Hu and Wellman (2003).

Gradient Ascent Algorithms

Infinitesimal gradient ascent (IGA) (Singh, Kearns, and Mansour 2000) is a policy gradient learning algorithm based on the limit of infinitesimal learning rates. It is shown that the average payoff of IGA converges to the pure Nash equilibrium payoff in two-agent, two-action matrix games, although policies may cycle in games with mixed equilibria. Each agent *i* participating in a game updates its policy π_i such that it follows the gradient of expected payoffs. The IGA algorithm has been generalized into the generalized infinitesimal gradient ascent (GIGA) algorithm beyond two actions using the regret measure by Zinkevich (2003). Regret measures how much worse an algorithm performs compared to the best static strategy, with the goal to guarantee at least zero average regret, that is no regret, in the limit. Since GIGA reduces to IGA in two-player, two-action games, it does not achieve convergence in all types of games. As a response to the fact that the IGA algorithm does not converge in all two-player two-action games, IGA-WoLF (win or learn fast) was introduced by Bowling (Bowling and Veloso 2002) in order to improve the convergence properties of IGA. The policies of Infinitesimal Gradient Ascent with WoLF learning rates are proven to converge to the Nash equilibrium policies in two-agent, two-action games (Bowling and Veloso 2002). The learning rate is made large if WoLF is losing. Otherwise, the learning rate is kept small as a good strategy has been found. In contrast to other reinforcement learning algorithms, IGA-WoLF assumes that the agents possess a lot of information about the payoff structure. In particular, sometimes agents are not able to compute the gradient of the reward function that is necessary for this algorithm because that information is not available. Another well known gradient-ascent type algorithm is the policy hill climber (PHC) explained in Bowling and Veloso (2002). PHC is a simple adaptive strategy, based on an agent's own actions and rewards, which performs hill climbing in the space of mixed policies. It maintains a Q-table of values for each of its base actions, and at every time step it adjusts its mixed strategy by a small step toward the greedy policy of its current Q-function. Also the PHC-WoLF algorithm needs prior information about the structure of the game. Related algorithms to infinitesimal gradient ascent have been devised to tackle this issue, such as for instance the weighted policy learner (WPL) algorithm of Abdallah and Lesser (2008). The GIGA-WoLF algorithm extended the GIGA algorithm with the WoLF principle (Bowling 2004), improving on its convergence properties. The algorithm basically keeps track of two policies, one of which is used for action selection and the other for approximating the Nash equilibrium.

Challenges in Multiagent Learning

In this section we discuss three main challenges we believe the field is currently facing and needs to tackle to continue its successful previous development.

Classification Limitations

The field of MAL is, so to say, still seeking for its identity (which is not surprising for such a young field). This search is important because significant and continuing progress in a research field can only be expected if it is clear what the research should be about. In response to this, several classifications of MAL have been proposed (see the previous Nature of MAL subsection). We believe that these characterizations fail fully to capture the essence and potential range of MAL: they all focus on the characterization of existing MAL approaches (and are very useful in this respect) and thus on "what *is* available," but say nothing about "what *could* (or should) be available." In this way, they are not appropriate for identifying important facets and forms of MAL that so far are not covered (or have been largely ignored) in the MAL field. To make this more concrete, we hark back to a classification scheme initially proposed in Weiss and Dillenbourg (1999) with the intention to work out key differences among single-agent and multiagent learning. This scheme distinguishes three types of MAL: multiplied learning, divided learning, and interactive learning.

In *multiplied learning,* there are several agents that learn independently of one another; they may interact but their interactions do not change the way the individual agents learn. An agent learns "as if it were alone" and thus acts as a "generalist" capable of carrying out all activities that as a whole constitute a learning process.

In *divided learning* a single-agent learning task or algorithm is divided a priori (that is, before any learning process starts) among multiple agents according to functional aspects of the algorithm or characteristics of the data to be processed (for example, their geographical distribution). The agents have a shared overall learning goal (which is not the case in multiplied learning). Interaction is required for putting the individual learning results together, but (as in the case of multiplied learning) this interaction only concerns the input and output of the agents' individual learning processes. An agent acts as a "specialist," being responsible for a specific part of an overall learning process.

In *interactive learning,* the agents are engaged in a single learning process that becomes possible through a shared understanding of the learning task and goal. The shared understanding is not necessarily available at the beginning of the learning process, but may emerge during the learning process as a result of knowledge-intensive interaction (for example, in the form of consensus building, argumentation, and mutual explanation, on the basis of an advanced communication language). The intrinsic purpose of interaction is jointly to construct, in a flexible and dynamic way, a successful learning path and thus a solution to the learning task. In the case of multiplied and divided learning the primary purpose of interaction is to provide input (data and information) to separate, encapsulated learning processes of individual agents. An agent involved in interactive learning does not so much act as a generalist or a specialist but as a regulator who influences the path of a joint learning process and as an integrator who synthesizes possibly conflictive perspectives of the different agents involved in this learning process.

Examples of MAL approaches from which this classification was derived are Haynes, Lau, and Sen (1996); Vidal and Durfee (1996) (multiplied learning), Sen, Sekaran, and Hale (1994) and Weiss (1993) (divided learning), and Bui, Kieronska, and Venkatesh (1996) and Nagendra Prasad, Lesser, and Lander (1996) (interactive learning). Although the field of MAL started up with research on all three learning types, it quickly focused on multiplied and divided learning and today nearly all available MAL approaches are variants of these two learning types. Representative examples of current research on multiplied and divided learning are Meng and Han (2009) and Chakraborty and Stone (2010), respectively. In contrast, interactive learning has largely been neglected and plays only a marginal role in current MAL research. As an effect, in recent years only very few MAL approaches have been proposed that fall into the "interactive learning" class (an example is Ontañón and Plaza [2010]). In a sense interactive learning is the most sophisticated form of multiagent learning and the field needs to make the effort to delve into it. Perhaps the field was not mature enough for this effort in the past, but today it is. A useful starting point for this effort and a valuable source of inspiration is the vast amount of literature on collaborative learning in groups of humans that is available in disciplines such as social and educational psychology (for example, see Smith and Gregor [1992]), because interactive learning is conceptually very close to this type of human-human learning. In Weiss and Dillenbourg (1999) three key processes are identified — dialogue-based conflict resolution, mutual regulation, and explanation — which are fundamental to collaborative learning but were not realized by MAL approaches available at the time that article was written. Interestingly, these processes still do not play a noticeable part in MAL.

Extending the Scope

Today the MAL field is dominated by work on reinforcement learning and, specifically, by research conducted at the intersection of reinforcement learning and game theory. Approximately 90 percent of the multiagent learning research presented at the last three AAMAS conferences (2009, 2010, and 2011) is situated at this intersection (counting the number of papers explicitly situated at this intersection in the multiagent learning sessions). A positive effect of this dominant interest is that tremendous progress has been achieved, and a remarkable theoretical understanding of multiagent reinforcement learning has been developed in this area in the previous 10 years or so, for example, Busoniu, Babuska, and De Schutter (2008); Yang and Gu (2009); Tuyls, 't Hoen, and Vanschoenwinkel (2006). Maybe a less positive effect is that the field, so to say, entangled itself in a "reinforcement learning and game theory" perspective on MAL that is probably too narrow in its conceptual and formal scope to embrace multiagent learning. We agree with Stone's (2007) remark: "However, from the AI perspective, the term 'multiagent learning' applies more broadly than can be usefully framed in game theoretic terms."

To broaden its current scope, the MAL field needs to open up again to a wider range of learning paradigms, as was the case in the early days (see previous subsection on historical perspective), building on the experience of the past 25 years and drawing explicit connections between the different paradigms to tackle more complex problems. Two paradigms that we consider as particularly appropriate in this respect are *transfer learning* (for example, Taylor and Stone [2009, 2011]) and *swarm intelligence* (Dorigo and Gambardella 1997, Dorigo and Stützle 2004). Transfer learning is, roughly speaking, the field that studies the transfer of knowledge learned in one task domain to another, related one. What makes transfer learning very interesting from the MAL perspective is that this transfer can take place between distinct learning entities, be it agents (software or robots) or humans. Thereby three transfer directions can be distinguished — from an agent to another agent, from an agent to a human, and from a human to an agent. By opening up to transfer learning, a fruitful linkage would be established between the MAL field on the one hand and psychology (where transfer learning has been a subject of study for more than a hundred years) and areas such as human computer interaction. Moreover, this would also establish a linkage to learning techniques that are closely related to transfer learning such as imitation learning (for example, Price and Boutilier [2003]), learning from demonstration (for example, Breazeal et al. [2006]) and multitask learning (for example, Ando and Zhang [2005]). Currently very little multiagent transfer learning work is available (for example, Ammar and Taylor [2011], Proper and Tadepalli [2009], Wilson et al. [2008], Wilson, Fern, and Tadepalli [2010]).

Swarm intelligence is a bioinspired machine-learning technique (He et al. 2007; Colorni, Dorigo, and Maniezzo 1992), largely based on the behavior of social insects (for example, ants and honeybees), that is concerned with developing self-organized and decentralized adaptive algorithms. The type and form of learning in swarm intelligence is characterized by a large population of cognition limited agents that locally interact. Rather than developing complex behaviors for single individuals, as is done in reinforcement learning, swarm intelligence investigates the emerging (intelligent) behavior of a group of simple individuals that achieve complex behavior through their interactions with one another. Consequently, swarm intelligence can be considered as a cooperative multiagent learning approach in that the behavior of the full set of agents is determined by the actions of and interactions among the individuals. Swarm intelligence and reinforcement learning are closely related, as both techniques use iterative learning algorithms based on trial and error and a "reinforcement signal" to find optimal solutions. The key difference though is how the reinforcement signal is used to modify an individual's behavior. Currently the most well-known swarm intelligence algorithms are pheromone-based (stigmergic), such as Ant Colony Optimization. For an overview, we refer to Bonabeau, Dorigo, and Theraulaz (1999) and Dorigo and Stützle (2004). Recently, interest has grown in nonpheromone-based approaches, mainly inspired by the foraging behavior of honeybees (Lemmens and Tuyls 2009, Alers et al. 2011).

In addition to transfer learning and swarm intelligence, we see several other learning paradigms that are inherently related to the concept of MAL and thus should attain much more attention in the MAL field than they have received so far. These paradigms are *coevolutionary learning* (Paredis 1995, Ficici and Pollack 1998), that is the field that investigates and develops learning algorithms inspired by natural evolution, using operators like selection, mutation, crossover, and others; *multiview learning* (Christoudias, Urtasun, and Darrell 2008), that is machine-learning methods that use redundant views of the same input data; *multistrategy learning* (Michalski and Tecuci 1995), that is, an approach in which two or more learning strategies are combined into one learning system; and *parallel inductive learning,* that is, the domain that studies how to exploit the parallelism present in many learning algorithms in order to scale to more complex problems (Provost and Kolluri 1999).

These paradigms have been the subject of research in the field of (single-agent) machine learning for years, and opening up to them would not only broaden the scope of the MAL field but also further strengthen the ties between single-agent and multiagent learning research.

Multiagent Learning in Complex Systems

The field of MAL has often dealt with rather simple applications, usually in toy-world scenarios or drawn from game theory and mostly with only a few (typically two) learning agents involved. We think this simplification makes sense and is helpful for getting a better understanding of principle possibilities, limitations, and challenges of MAL in general and of specific MAL techniques in particular. But it is not sufficient. In addition, the MAL field needs to focus more than it currently does on complex and more realistic applications (and is mature enough for doing so) for two main reasons. First, eventually this is the only way to find out whether and to what extent MAL can fulfill the expectations in its benefits; and second, this is the best way to stimulate new ideas and MAL research directions that otherwise would not be explored. There is a broad range of potential real-life application domains for MAL such as ground and air traffic control, distributed surveillance, electronic markets, robotic rescue and robotic soccer, electric power networks, and so on. Some work has already successfully explored the integration of *learning* in these real-world domains and has shown promising results that justify a stronger focus by the community on complex systems; for example, see Richter, Aberdeen, and Yu (2006); Kalyanakrishnan, Liu, and Stone (2006); Abbeel et al. (2007); Agogino and Tumer (2012). Stone (2007) stated that it is still unclear whether complex multiagent learning problems can be handled at all. We prefer to formulate it in this way: it is unclear whether this is possible on the basis of the current perspective on MAL. In fact, currently there is *no* compelling evidence that multiagent learning in complex systems is not possible. To take a broader and more interdisciplinary approach to MAL, as proposed in this article, is an important step toward efficient multiagent learning in complex applications.

Conclusions

Multiagent learning is a young and exciting field that has already produced many research results and has seen a number of important developments in a relatively short period of time. In this article we have reviewed the field starting by sketching its history and most important developments since the end of the 1980s. This article continues by introducing the basics of the field and delving deeper into the nature of multiagent learning, answering the question what MAL is really about. We described some of the milestones of the field and looked into the current and future challenges. Over the past years MAL has seen great progress at the intersection of game theory and reinforcement learning due to its strong focus on this intersection. However, in order to overcome some of the current issues, as identified in this article, we are convinced the field should also take a broader and more interdisciplinary approach to MAL, which is an important step toward efficient multiagent learning in complex applications. As an example we have discussed the potential value of transfer learning and swarm intelligence for MAL. Moreover, extending the scope needs to be done by building on the experience of the past 25 years and by drawing explicit connections between the different paradigms in order to tackle more complex problems. We believe MAL is also in need of shifting some of its focus to more complex and more realistic applications for two main reasons. First, eventually this is the only way to find out whether and to what extent MAL can fulfill the expectations in its benefits of creating such applications; and second, this is the best way to stimulate new ideas and MAL research directions that otherwise might not be explored.

Acknowledgments

We would like to thank the anonymous reviewers and the editors of this issue for their insightful comments and suggestions. Furthermore, we wish to thank our colleagues for valuable discussions and feedback.

References

Abbeel, P.; Coates, A.; Quigley, M.; and Ng, A. Y. 2007. An Application of Reinforcement Learning to Aerobatic Helicopter Flight. In *Neural Information Processing Systems,* volume 19. Cambridge, MA: The MIT Press.

Abdallah, S., and Lesser, V. R. 2008. A Multiagent Reinforcement Learning Algorithm with Non-Linear Dynamics. *Journal of Artificial Intelligence Research* 33: 521–549.

Agogino, A. K., and Tumer, K. 2012. A Multiagent Approach to Managing Air Traffic Flow. *Journal of Autonomous Agents and MultiAgent Systems* 24(1): 1–25.

Agogino, A. K., and Tumer, K. 2008. Analyzing and Visualizing Multiagent Rewards in Dynamic and Stochastic Environments. *Journal of Autonomous Agents and MultiAgent Systems* 17(2): 320–338.

Alers, S.; Bloembergen, D.; Hennes, D.; de Jong, S.; Kaisers, M.; Lemmens, N.; Tuyls, K.; and Weiss, G. 2011. Bee-Inspired Foraging in an Embodied Swarm. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems.* Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Ammar, H. Y. B., and Taylor, M. E. 2011. Common Subspace Transfer for Reinforcement Learning Tasks. Paper presented at the AAMAS 2011 Workshop on Adaptive Learning Agents and Multiagent Systems, Taipei, Taiwan, May 2.

Ando, R. K., and Zhang, T. 2005. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research* 6(11): 1817–1853.

Banerjee, A. 1992. A Simple Model of Herd Behavior. *Quarterly Journal of Economics* 107(3): 797–817.

Bellman, R. 1957. *Dynamic Programming.* Princeton, NJ: Princeton University Press.

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The Complexity of Decentralized Control of Markov Decision Processes. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence,* 32–37. San Francisco: Morgan Kaufmann.

Bonabeau, E.; Dorigo, M.; and Theraulaz, G. 1999. *Swarm Intelligence: From Natural to Artificial Systems.* New York: Oxford University Press.

Bowling, M. 2004. Convergence and No-Regret In Multiagent Learning. In *Advances in Neural Information Processing Systems,* volume 17, 209–216, Cambridge, MA: MIT Press.

Bowling, M., and Veloso, M. 2002. Multiagent Learning Using a Variable Learning Rate. *Artificial Intelligence* 136(2): 215–250.

Boyd, R., and Richerson, P. 1985. *Culture and the Evolutionary Process.* Chicago: The University of Chicago Press.

Breazeal, C.; Berlin, M.; Brooks, A.; Gray, J.; and Thomaz, A. 2006. Using Perspective Taking to Learn from Ambiguous Demonstrations. *Robotics and Autonomous Systems* 54(5): 385–393.

Bui, H.; Kieronska, D.; and Venkatesh, S. 1996. Learning Other Agents' Preferences in Multiagent Negotiation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* 114–119. Menlo Park, CA: AAAI Press.

Busoniu, L.; Babuska, R.; and De Schutter, B. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38(2): 156–172.

Chakraborty, D., and Stone, P. 2010. Convergence, Targeted Optimality and Safety in Multiagent Learning. In *Proceedings of the 27th International Conference on Machine Learning.* Madison, WI: Omnipress.

Christoudias, C.; Urtasun, R.; and Darrell, T. 2008. Multiview Learning in the Presence of View Disagreement. In *Proceedings of the Twenty-Fourth Annual Conference on Uncertainty in Artificial Intelligence,* 88–96. Corvallis, OR: Association for Uncertainty in Artificial Intelligence Press.

Claus, C., and Boutilier, C. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of the 15th National Conference on Artificial Intelligence,* 746–752. Menlo Park, CA: AAAI Press.

Colorni, A.; Dorigo, M.; and Maniezzo, V. 1992. Distributed Optimization by Ant Colonies. In *Towards a Practice of Autonomous Systems: Proceedings of the First*

European Conference on Artificial Life, ed. F. J. Varela and P. Bourgine, 134–142. Cambridge, MA: MIT Press.

Dorigo, M., and Gambardella, L. M. 1997. Ant Colony Systems: A Cooperative Learning Approach to the Travelling Salesman Problem. *IEEE Transactions on Evolutionary Computation* 1(1): 53–66.

Dorigo, M., and Stützle, T. 2004. *Ant Colony Optimization*. Cambridge, MA: The MIT Press/Bradford Books.

Ficici, S. G., and Pollack, J. B. 1998. Challenges in Coevolutionary Learning: Arms-Race Dynamics, Openendedness, and Mediocre Stable States. In *Proceedings of the 6th International Conference on Artificial Life,* 238–247. Cambridge, MA: The MIT Press.

Galef, B. 1988. Imitation in Animals: History, Definition, and Interpretation of Data from the Psychological Laboratory. In *Social Learning: Psychologic and Biological Perspectives,* ed. T. Zentall, and B. Galef. Hillsdale, NJ: Lawrence Erlbaum Associates.

Haynes, T.; Lau, K.; and Sen, S. 1996. Learning Cases to Compliment Rules for Conflict Resolutions in Multiagent Systems. In *Adaptation, Coevolution and Learning in Multiagent Systems*. Papers from the 1996 AAAI Symposium, ed. S. Sen. Technical Report SS-96-01, 51–56. Menlo Park, CA: AAAI Press.

He, X.; Zhu, Y.; Hu, K.; and Niu, B. 2007. A Swarm-Based Learning Method Inspired by Social Insects. In *Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence: Third International Conference on Intelligent Computing,* volume 4682 of Lecture Notes in Computer Science, ed. D.-S. Huang, L. Heutte, and M. Loog,, 525–533. Berlin: Springer.

Hu, J., and Wellman, M. P. 2000. Experimental Results on Q-Learning for General-Sum Stochastic Games. In *Proceedings of the Seventeenth International Conference on Machine Learning,* 407–414. Morgan Kaufmann Publishers Inc.

Hu, J., and Wellman, M. P. 2003. Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning Research* 4(11): 1039–1069.

Huhns, M., and Weiss, G., eds. 1998. Special Issue on Multiagent Learning. *Machine Learning Journal* 33(2–3).

Kalyanakrishnan, S.; Liu, Y.; and Stone, P. 2006. Half Field Offense in RoboCup Soccer: A Multiagent Reinforcement Learning Case Study. In *RoboCup 2006: Robot Soccer World Cup XI,* Lecture Notes in Computer Science 4434, 72–85. Berlin: Springer

Laland, K.; Richerson, P.; and Boyd, R. 1993. Animal Social Learning: Toward a New Theoretical Approach. In *Perspectives in Ethology,* ed. P. Klopfer, P. Bateson, and N. Thomson. New York: Plenum press.

Lemmens, N., and Tuyls, K. 2009. Stigmergic Landmark Foraging. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems,* 497–504. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Littman, M. 1994. Markov Games as a Framework for Multiagent Reinforcement Learning. In *Proceedings of the Eleventh International Conference on Machine Learning,* 157–163. San Francisco: Morgan Kaufmann.

Littman, M. L. 2001. Friend-or-Foe Q-Learning in General-Sum Games. In *Proceedings of the Eighteenth International Conference on Machine Learning,* 322–328. San Francisco: Morgan Kaufmann.

Manderick, B., and Moyson, F. 1988. The Collective Behavior of Ants: An Example of Self-Organization in Massive Parallelism. Paper presented at the AAAI Spring Symposium on Parallel Models of Intelligence. March 1988.

Manderick, B., and Spiessens, P. 1989. Fine-Grained Parallel Genetic Algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms,* 428–433. San Francisco: Morgan Kaufmann.

Meng, W., and Han, X. 2009. Parallel Reinforcement Learning Algorithm and Its Application. *Computer Engineering and Applications* 45(34): 25–28.

Michalski, R., and Tecuci, G., eds. 1995. *Machine Learning. A Multistrategy Approach.* San Francisco: Morgan-Kaufmann.

Nagendra Prasad, M.; Lesser, V.; and Lander, S. 1996. Learning Organizational Roles in a Heterogeneous Multiagent System. In *Proceedings of the Second International Conference on Multiagent Systems,* 291–298. Menlo Park, CA: AAAI Press.

Ontañón, S., and Plaza, E. 2010. Multiagent Inductive Learning: An Argumentation-Based Approach. In *Proceedings of the 27th International Conference on Machine Learning,* 839–846. Madison, WI: Omnipress.

Panait, L., and Luke, S. 2005. Cooperative Multiagent Learning: The State of The Art. *Autonomous Agents and MultiAgent Systems* 11(3): 387–434.

Panait, L.; Tuyls, K.; and Luke, S. 2008. Theoretical Advantages of Lenient Learners: An Evolutionary Game Theoretic Perspective. *Journal of Machine Learning Research* 9: 423-457

Paredis, J. 1995. Coevolutionary Computation. *Artificial Life* 2(4): 355–375.

Pfeifer, R.; Schreter, Z.; Fogelman-Soulie, F.; and Steels, L. 1989. *Connectionism in Perspective*. Amsterdam: Elsevier Science Publishers.

Price, B., and Boutilier, C. 2003. Accelerating Reinforcement Learning Through Implicit Imitation. *Journal of Artificial Intelligence Research* 19: 569–629.

Proper, S., and Tadepalli, P. 2009. Multiagent Transfer Learning Via Assignment-Based Decomposition. In *Proceedings of the 2009 International Conference on Machine Learning and Applications,* 345–350. Los Alamitos, CA: IEEE Computer Society.

Provost, F., and Kolluri, V. 1999. A Survey of Methods for Scaling Up Inductive Algorithms. *Data Mining and Knowledge Discovery* 3(2): 131–169.

Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* New York: John Wiley and Sons.

Richter, S.; Aberdeen, D.; and Yu, J. 2006. Natural Actor-Critic for Road Traffic Optimisation. In *Neural Information Processing Systems,* volume 18, 1169–1176. Cambridge, MA: The MIT Press.

Sen, S., and Weiss, G. 1999. Learning in Multiagent Systems. In *Multiagent Systems,* ed. G. Weiss. Cambridge, MA: The MIT Press. 259–298.

Sen, S.; Sekaran, M.; and Hale, J. 1994. Learning to Coordinate Without Sharing Information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence,* 426–431. Menlo Park, CA: AAAI Press.

Sen, S., ed. 1996. *Adaptation, Coevolution and Learning in Multiagent Systems.* Papers from the 1996 Spring Symposium. Technical Report SS-96-01. Menlo Park, CA: AAAI Press.

Shoham, Y.; Powers, R.; and Grenager, T. 2007. If Multiagent Learning Is the Answer, What Is the Question? *Artificial Intelligence* 171(7): 365–377.

Singh, S. P.; Kearns, M. J.; and Mansour, Y. 2000. Nash Convergence of Gradient Dynamics in General-Sum Games. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence,* 541–548. San Francisco: Morgan Kaufmann Publishers Inc.

Smith, B. L., and Gregor, J. T. 1992. What Is Collaborative Learning? In *Collaborative Learning: A Sourcebook for Higher Education,* ed. A. Goodsell, M. Maher, V. Tinto, B. L. Smith, and J. MacGregor, 1–49. University Park, PA: National Center on Postsecondary Teaching, Learning, and Assessment, Pennsylvania State University. .

Steels, L. 1987. Massive Parallelism for Artificial Intelligence. *Microprocessing and Microprogramming* 21(1–5): 17–19.

Steels, L. 1988. Artificial Intelligence and Complex Dynamics. In *Concepts and Techniques of Knowledge-Based Systems*. Amsterdam, The Netherlands: North Hollad.

Steels, L. 1996. Self-Organising Vocabularies. In *Proceedings of Artificial Life,* Volume V. Cambridge, MA: The MIT Press, Cambridge.

Stone, P., and Veloso, M. 2000. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots* 8(3): 345–383.

Stone, P. 2007. Multiagent Learning Is Not the Answer. It Is the Question. *Artificial Intelligence* 171(7): 402–405.

’t Hoen, P.; Tuyls, K.; Panait, L.; Luke, S.; and la Poutré, H. 2006. An Overview of Cooperative and Competitive Multiagent Learning. In *Learning and Adaptation in MultiAgent Systems,* Lecture Notes in Artificial Intelligence volume 3898, ed. K. Tuyls, P. ’t Hoen, K. Verbeeck, and S. Sen, 1–50. Berlin: Springer

Tan, M. 1993. Multiagent Reinforcement Learning: Independent Versus Cooperative Agents. In *Proceedings of the Tenth International Conference on Machine Learning,* 330–337. San Francisco: Morgan Kaufmann.

Taylor, M. E., and Stone, P. 2009. Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research* 10(1): 1633–1685.

Taylor, M. E., and Stone, P. 2011. An Introduction to Inter-Agent Transfer for Reinforcement Learning. *AI Magazine* 32(1): 15–34.

Tumer, K.; Agogino, A.; and Wolpert, D. 2002. Learning Sequences of Actions in Collectives of Autonomous Agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems,* 378–385. New York: Association for Computing Machinery.

Tuyls, K., and Nowe, A. 2005. Evolutionary Game Theory and Multiagent Reinforcement Learning. *The Knowledge Engineering Review* 20(1): 63–90.

Tuyls, K., and Parsons, S. 2007. What Evolutionary Game Theory Tells Us About Multiagent Learning. *Artificial Intelligence* 171(7): 406–416.

Tuyls, K.; ’t Hoen, P.; and Vanschoenwinkel, B. 2006. An Evolutionary Dynamical Analysis of Multiagent Learning in Iterated Games. *Journal of Autonomous Agents and Multiagent Systems* 12(1): 115–153.

Vega-Redondo, F. 2003. *Economics and the Theory of Games*. Cambridge: Cambridge University Press.

Vidal, J., and Durfee, E. 1996. The Impact of Nested Agent Models in an Information Economy. In *Proceedings of the 2nd International Conference on Multiagent Systems,* 377–384. Menlo Park, CA: AAAI Press.

Watkins, C. 1989. Learning with Delayed Rewards. Ph.D. Dissertation, Cambridge University, Cambridge, UK.

Weibull, J. W. 1996. *Evolutionary Game Theory.* Cambridge, MA: The MIT Press.

Weiss, G. 1993. Learning to Coordinate Actions In Multiagent Systems. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence,* 311–316. San Francisco: Morgan Kaufmann.

Weiss, G. 1996. Adaptation and Learning in Multiagent Systems: Some Remarks and a Bibliography. In *Adaption and Learning in Multiagent Systems,* volume 1042 of Lecture Notes in Artificial Intelligence, ed. G. Weiss and S. Sen, S. Berlin: Springer-Verlag.

Weiss, G., ed. 1997. *Distributed Artificial Intelligence Meets Machine Learning,* volume 1221 of Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag.

Weiss, G., ed. 1998. Special Issue on Learning in Distributed Artificial Intelligence Systems. *Journal of Experimental and Theoretical Artificial Intelligence* 10(3).

Weiss, G., and Dillenbourg, P. 1999. What Is “Multi” in Multiagent Learning? In *Collaborative Learning: Cognitive and Computational Approaches,* ed. P. Dillenbourg, 64–80. Pergamon Press.

Weiss, G., and Sen, S., eds. 1996. *Adaption and Learning in Multiagent Systems,* volume 1042 of Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag.

Whitehead, S. D. 1991. A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning. In *Proceedings of the Ninth National Conference on Artificial Intelligence,* 607–613. Menlo Park, CA: AAAI Press.

Wilson, A.; Fern, A.; Ray, S.; and Tadepalli, P. 2008. Learning and Transferring Roles in Multiagent. In *Transfer Learning for Complex Tasks: Papers from the AAAI Workshop*. AAAI Technical Report WS-08-13. Menlo Park, CA: AAAI PRess.

Wilson, A.; Fern, A.; and Tadepalli, P. 2010. Bayesian Role Discovery for Multiagent Reinforcement Learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent System,* 1587–1588. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Wolpert, D. H., and Tumer, K. 2001. Optimal Payoff Functions for Members of Collectives. *Advances in Complex Systems* 4(2/3): 265–279.

Yang, E., and Gu, D. 2009. Multirobot Systems with Agent-Based Reinforcement Learning: Evolution, Opportunities and Challenges. *International Journal on Modelling, Identification and Control* 6(4): 271–286.

Zinkevich, M. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the Twentieth International Conference on Machine Learning,* 928–936. Menlo Park, AAAI Press.

Karl Tuyls works as an associate professor in artificial intelligence at the Department of Knowledge Engineering, Maastricht University (The Netherlands) where he leads a research group on swarm robotics and learning in multiagent systems (Maastricht Swarmlab). Previously, he held positions at the Vrije Universiteit Brussel (Belgium), Hasselt University (Belgium), and Eindhoven University of Technology (The Netherlands). His main research interests lie at the intersection of reinforcement learning, multiagent or robot systems, and (evolutionary) game theory. He was a coorganizer of several events on this topic, such as the European Workshop on Multiagent Systems (EUMAS’05), the Belgian-Dutch conference on AI (BNAIC’05 and ’09), and workshops on adaptive and learning agents (EGTMAS’03, LAMAS’05, ALAMAS’07, ALAg and ALAMAS’08, ALA’09). He is associate editor of the *International Journal of Agent Technologies and Systems* and the *Journal of Entertainment Computing* and served on the program committees of various conferences and workshops. In 2000 he was awarded the Information Technology prize in Belgium and in 2007 he was elected best junior researcher (TOPDOG) of the faculty of humanities and sciences, Maastricht University, the Netherlands.

Gerhard Weiss is a full professor and head of the Department of Knowledge Engineering (DKE), Faculty of Humanities and Sciences, Maastricht University. Before joining Maastricht University in 2009, he was the scientific director of Software Competence Center Hagenberg GmbH, Austria, and an assistant professor at the Department of Computer Science of Technical University Munich, Germany. He received his Ph.D. in computer science from Technical University Munich and his Habilitation degree from Johannes-Kepler University Linz, Austria. His main research interests are in artificial intelligence, multiagent technology, and knowledge-based interaction among computer-based systems. He is an editorial board member of several international computer science and AI journals. He was a board member of the International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS) and of two European networks of excellence (Agentlink and Exystence). He has served as a referee for various governmental and private research funding organizations and as a scientific consultant for companies and industrial organizations.