J Braz Comput Soc (2011) 17:163–174
DOI 10.1007/s13173-011-0038-2

ORIGINAL PAPER

Learning to cooperate in the Iterated Prisoner's Dilemma by means of social attachments

Ana L.C. Bazzan · Ana Peleteiro · Juan C. Burguillo

Received: 16 February 2011 / Accepted: 19 July 2011 / Published online: 31 August 2011

Abstract The Iterated Prisoner's Dilemma (IPD) has been used as a paradigm for studying the emergence of cooperation among individual agents. Many computer experiments show that cooperation does arise under certain conditions. In particular, the spatial version of the IPD has been used and analyzed to understand the role of local interactions in the emergence and maintenance of cooperation. It is known that individual learning leads players to the Nash equilibrium of the game, which means that cooperation is not selected. Therefore, in this paper we propose that when players have social attachment, learning may lead to a certain rate of cooperation. We perform experiments where agents play the spatial IPD considering social relationships such as belonging to a hierarchy or to coalition. Results show that learners end up cooperating, especially when coalitions emerge.

Keywords Game theory · Iterative Prisoner's Dilemma · Reinforcement learning · Agent-based simulation

A previous version of this paper appeared at BWSS 2010, the Brazilian Symposium on Social Simulation.

A.L.C. Bazzan (✉)
PPGC/Instituto de Informática, Universidade Federal do Rio Grande do Sul, Caixa Postal 15064, CEP 91501-970
Porto Alegre, RS, Brazil
e-mail: bazzan@inf.ufrgs.br

A. Peleteiro · J.C. Burguillo
Telematics Engineering Department, University of Vigo, 36310 Vigo, Spain

A. Peleteiro
e-mail: apeleteiro@det.uvigo.es

J.C. Burguillo
e-mail: J.C.Burguillo@det.uvigo.es

1 Introduction

The concept of evolution of cooperation has been successfully studied using metaphors such as the Iterated Prisoner's Dilemma (IPD), which is the repeated version of the PD game. In the PD, two individuals are questioned separately over their involvement on a crime. They have a simple choice: either to confess a mutual crime (thereby implicating the other) and accept the consequences, or to deny all involvement and hope that the other does likewise. We remark that by confessing there is a penalty reduction as it will be clear when the payoff matrix is introduced.

This game models a more general situation in which two individuals have to decide, in an isolated way, whether to cooperate or to defect. The payoff, however, stems from the *joint* decision. The game is so formulated that mutual cooperation yields the highest joint payoff, but there is a high incentive for individual defection.

Nowadays, the IPD game has been applied to a variety of disciplines: economy, biology, artificial intelligence, social sciences, e-commerce, etc. It also occurs in colloquial situations.[1]

In a classical work, Axelrod [2] has shown that cooperation can emerge in a society of individuals with selfish motivations. An interesting spatial version of the IPD has been suggested and analyzed in [24], where Nowak and May try to understand the role of local interactions in the emergence and maintenance of cooperation. In their formulation, players imitate their most successful neighbors so that a reasonable rate of cooperation is observed. However, this pure deterministic approach, in which players have no memory, was later criticized by Huberman and Glance [14] who have

[1] http://freakonomics.blogs.nytimes.com/2010/09/17/the-prisoners-dilemma-makes-a-reality-tv-appearance/

run an asynchronous updating of strategies, claiming that in this case the coexistence of cooperators and defectors was no longer possible.

These two works have motivated an enormous number of extensions considering several types of relationships among players. In general, the more strong and nonlocal this relationship, the higher the rate of cooperation.

In a sense, reinforcement learning (RL) can be seen as a way to break the effects of that determinism criticized by [14], because one may design learners with different sensing capabilities, with different action selection strategies, and/or with different ways to store their utility estimates, as proposed by Sandholm and Crites [29]. However, RL causes an agent to learn an individually optimal policy meaning that its behavior is a best response to the strategies of the other players. Hence, ultimately it is expected that all agents converge to mutual defection since this is the best possible response in the IPD. The excessive concern with learning Nash equilibria in multiagent encounters has been criticized, e.g., in [31, 32]. Shoham and colleagues single out some problems due to focusing on what they call the "Bellman heritage." According to these authors, it seems that most of the research so far has focused on the play to which agents converge, not on the payoff agents obtain.

This is particularly the case regarding the use of RL in the IPD. Indeed, results reported in [29] were not encouraging: "clear cooperation seldom emerged in experiments with two learners even though the discount factor was set high to stimulate cooperation" [29]. Although in some situations, agents need not to seek cooperation, in the IPD in particular, the payoff matrix is so formulated that mutual cooperation leads to the highest payoff for the society. In the present paper, we assume that agents seek to maximize this reward.

In the work of [29], another sensitive issue arises: only a two-player IPD game is considered, in which both players are aware of the joint actions. This assumes full and reliable communication among these two agents. If such approach is extended to a scenario with dozens or hundreds of agents, the outcome is likely to degrade.

Therefore, further investigations around the RL approach by [29] are necessary in order to find out whether or not agents can learn to cooperate. To the best of our knowledge, there has been no further attempt to address the spatial n-player IPD game using RL, especially tackling emergence of cooperation. In [35], IPD is used, but the aim there is to analyze the dynamics of multiagent learning in multistate problems. Therefore, they have modified the IPD game which is then represented by two payoff matrices (two states). In this modified game, the Nash equilibria in both states are neither mutual defection nor mutual cooperation. Thus, comparisons to other approaches are not straightforward. Moreover, their approach works for two-player games only.

In the present paper, we claim that emergence of cooperation in the IPD using RL can only be achieved if agents are equipped with some kind of behavior that we shall denote here as social attachment. These attachments may be spatial relationships (e.g., [24]), small-world (e.g., [1]), or emotions toward group attachments (e.g., [5]). The reason for this claim is that such approaches have proved to improve cooperation, though in none of these cases learning was used. Here we propose two approaches to couple social attachment and RL: hierarchical organization and coalition. The former is based on a preexisting organizational structure, while in the latter the structure itself emerges out of the agents's interactions. Thus, the main question here is whether these two approaches can support cooperation in a defection-prone environment.

The remainder of the paper is structured as follows: Firstly, in Sect. 2, we give more details about the IPD and point out other relevant works. After, in Sect. 3, we discuss related works on multiagent RL in order to give an idea of the challenges behind this problem. In Sect. 4, we formalize the two approaches for implementing the social attachments among agents, as discussed previously. Section 5 describes the scenario used for the empirical validation and analyzes the results of our experiments. Finally, the conclusions and future works are presented in Sect. 6.

2 Prisoner's dilemma

The two-agent PD is an abstraction of some kinds of social situations. Obviously, it is just a proxy for abstract investigations. Nevertheless, it is useful as it serves as a kind of benchmark that allows comparisons to be made.

The important characteristics of the PD, as mentioned, are as follows. Two suspects of a crime (agents or players) are questioned separately (no communication between them) over their involvement on a crime. They have a simple choice, either to remain silent, i.e., cooperate (C) or to confess a criminal action made by both of them (thereby implicating the other), i.e., to defect (D). This game models a more general situation in which two individuals have to decide, *in an isolated way*, whether to cooperate or defect. The payoff however stem from the *joint* decision.

The PD is a metaphor for acting in a socially responsible way (C) or according to self-interest (D), which is harmful to both agents. To see why this is so, consider the payoff matrix shown in Table 1. It represents the payoff (also known as utility or reward) a player obtains depending on its own action and on the opponent's one. This matrix is common-knowledge, i.e., both agents know it, both know that the both know the matrix, and so on. In this matrix, T means the temptation to defect, R is the reward for mutual cooperation, P is the punishment for mutual defection,

Table 1 Payoff-matrix for the IPD game (payoffs are for j_1/j_2)

		Agent j_1	
		C	D
Agent j_2	C	R/R	T/S
	D	S/T	P/P

and S is the sucker's payoff. To be defined as a PD, the game must respect the following constraints: $T > R > P > S$ and $2R > T + S$.

Given these constraints, in the one-shot PD, the optimal action for both agents is to select D because there is a risk of ending up with S if selecting C when the opponent selects D. Choosing D ensures the highest payoff for any agent no matter what the opponent does. In fact, it is easy to show that mutual defection (DD) is the unique Nash equilibrium in this game. This is also the outcome when both players know that they are going to play the game for a given number of times, because the optimal action in the last round is to play D and, by induction, playing D in all previous rounds is the best thing to do.

Collectively speaking, mutual cooperation would do better though. For instance, if $R = 3$, $T = 5$, $P = 1$, and $S = 0$, mutual cooperation has the *social* utility of 6, whereas all other combinations of actions are worse off.

In practice, agents often encounter each other more than once (and for an unknown number of times), in a repeated PD (or, as it is more commonly referred, an iterated PD). Here, it may be beneficial to cooperate in some rounds, even if one is selfish, in the hope of a reciprocal behavior. Therefore, the IPD has been used as a paradigm for studying emergence of cooperation among individuals.

Many computer experiments show that cooperation does arise under specific conditions. This was verified in a computer tournament [2], in which the winner was the strategy called Tit-for-Tat (TFT): it begins cooperating and then repeats the previous action of the opponent. It is cooperative but retaliates defection, returning to cooperation after the opponent cooperates. It must be noted that this strategy poses a burden for resource-bounded agents because it assumes that they are able to remember past encounters and are able to compute relatively sophisticated strategies, especially in the case of some more complex extensions of TFT.

Therefore, Nowak and May [24] have proposed a simple approach where no memory is necessary and imitation plays a key role. It works in a two-dimensional spatial array where agents are either cooperators or defectors. Hence, there is no need of processing strategies to decide how to play. In every round, players interact with the immediate neighbors and after a round is over, each site is occupied by the best player in the neighborhood (including the previous owner itself). Best player means the one that has collected the highest payoff.

The work by Nowak and May is important because it has determined that players do not need to play the game with the whole population. By making this assumption, different equilibria are likely to be established in different neighborhoods. Thereafter, this idea has appeared also in [22] and numerous other papers.

As previously mentioned, the purely deterministic approach introduced by [24] was later criticized [14]. To address this issue, other works were proposed. Lindgren and Nordahl [19] achieved patterns of coexistence using a cellular automata to update strategies on the sites of the grid, which is able to avoid the synchronization. Abramson and Kuperman [1] place agents in a small-world network, i.e., agents interact not only with their closest neighbors but also with agents located somewhere in the network (in particular, their network is a ring). Kim et al. [16] have shown that the small-world metaphor affects cooperation in the IPD not only when the agents are placed in a ring. Rescuing the two-dimensional spatial configuration of Nowak and May, Kim et al. investigated the effect of the connections of a given node to other (distant) nodes. This paper also introduces the notion of an influential node, i.e., a node which can be compared to a mass media person who influences others. The influential node cannot be influenced by those linked to it though.

In another line of research (the previously discussed are all based only on spatial interaction), [5] use sentiments like generosity toward others and guilt for not having played fair with someone to prevent IPD players from trying to maximize the gain in the short term only. The results have shown that in a society where agents have emotions, to behave rationally (in the classical sense of game theory) may not be the best attitude both for the individual and for the social group.

Two works have combined the spatial interaction and the sentiment-based approach. In [4], agents may depict various types of emotions (the so-called OCC model by [25] was used to categorize the emotions) while playing the IPD. Results showed that the ratio of cooperators is slightly higher when agents make decisions using emotions. Other works have analyzed the performance of agents that can display an altruistic behavior toward its acquaintances (altruistic agents), which were interested in the good performance of their group as a whole since the best performance in the group would be imitated.

Approaches based on RL are not popular in the IPD literature probably due to the convergence to the Nash equilibrium that means mutual defection. In fact, this was verified already in 1995 by [29], when RL started turning popular in multiagent systems. This may explain the lack of interest relating RL and IPD, while RL has been successfully applied to other games, as discussed in the next section.

3 Multiagent learning

Before we discuss the multiagent RL (MARL) problem, it is necessary to introduce the basic, monoagent version of it. Usually, RL problems in which there is only one agent are modeled as Markov Decision Processes (MDPs). These are described by a set of states, S, a set of actions, A, a reward function $R(s,a) \to \mathbb{R}$, and a probabilistic state transition function $T(s,a,s') \to [0,1]$. At each time step t, an experience tuple $\langle s,a,s',r\rangle$ denotes the fact that the agent was in state $s \in S$, performed action $a \in A$ and ended up in $s' \in S$ with reward r. We drop the index t here.

Given an MDP, the goal is to learn a policy π^*, which is a mapping from states to actions such that the expected value of the sum of the discounted future reward is maximized.

In this paper, we use Q-learning, a popular model-free RL algorithm, which is useful when agents do not have a model of the state transition function T. Q-Learning works by estimating state–action values, the Q-values, which are numerical estimators of quality for a given pair of state and action. More precisely, a Q-value $Q(s,a)$ represents the maximum discounted sum of future rewards an agent can expect to receive if it starts in s, chooses action a and then continues to follow an optimal policy. The Q-Learning algorithm approximates $Q(s,a)$ as the agent acts in a given environment. The update rule for each experience tuple $\langle s,a,s',r\rangle$ is as in (1), where α is the learning rate and γ is the discount for future rewards. If all pairs state–action are visited infinitely often, then Q-learning is guaranteed to converge to the correct Q-values with probability one [37].

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big) \quad (1)$$

Next, we discuss the problems posed by a high number of agents in a multiagent reinforcement learning (MARL) scenario.

In game theoretic terms, multiagent learning can be translated into the problem of learning to play best replies. Similarly to the monoagent RL problem that is modeled by an MDP, the multiagent version is modeled by an MMDP (multiagent MDP), also known as stochastic game (SG). In an SG, if every agent is playing a best reply to its opponents, the combination of strategies is called a Nash equilibrium. Here, agents may or may not know the structure of the game they are playing, meaning their joint states, joint actions, and rewards. As put by Sandholm [28], MARL is very important in case agents do not know the structure of the game which is the case in most SG. This problem thus involves two learning tasks: learning the structure of the game and learning how to play. Several approaches to these tasks have been proposed. Since a comprehensive discussion is not possible here, we refer the reader to [26, 31] and references therein. We remark that the SG formalism is not the only approach to MARL. For a discussion on multiagent learning in general, see [32].

There are basically two ways to handle a MARL problem. The first option is that a central entity updates a huge Q-table where each entry is retrieved by the combination of each agent's state and action pair, i.e., by a joint state and joint action combination. Hence, the vector that contains the reward of all agents is determined as $\mathbf{r} : S_1 \cdots \times S_n \times A_1 \times \cdots \times A_n \to \mathbb{R}^n$, where n is the number of agents. This of course is not feasible for more than a handful of agents given that the size of the table is exponential in n. Moreover, it makes the strong assumption that such a central entity exists in the first place, which comes in total contradiction to the very basic idea of distributed, multiagent systems. Hence, this formalization is rarely seen in multiagent systems.

The second option is that each agent individually keeps its own Q-table where an entry is simply the combination of its state and action, i.e., a pair. Of course this table may grow as well if the number of states and/or actions is too high but it does not increase as fast as in the first option. The problem with this latter approach is that it is not efficient. The main reason is due to the fact that while one agent is trying to model the environment (other agents included), the others are doing the same and potentially changing the environment they share. This yields an environment that is inherently nonstationary. Therefore, at least in the general case, the convergence guarantee, as previously known from monoagent reinforcement learning (e.g. Q-learning), no longer holds. This means the convergence to a Nash equilibrium is not guaranteed.

In summary, considering a high number of agents in MARL turns the problem inherently more complex. This complexity has many consequences. In particular, it has implications in the computational complexity, in the convergence to the Nash equilibrium, or both.

Fulda and Ventura [11] have isolated three factors that can cause a system to behave poorly: suboptimal individual convergence, action shadowing, and the equilibrium selection problem. The latter was discussed initially by Claus and Boutilier [8] and is especially an issue in common-reward coordination games. One characteristic of these games is that there will always exist a Pareto optimal equilibrium,[2] but this needs not be unique. This structure of the game often leads agents to miscoordinate. Action shadowing occurs when one individual action appears better than another, even though the second individual action is potentially superior. This can occur because agents maintain Q-values only for individual actions, but receive rewards based on the joint action executed. Action shadowing is likely to occur where failed coordination attempts are punished, as in

[2] Pareto optimal implies that no joint action improves any agent's payoff without making another agent worse off.

penalty games. The relevant factor in the present paper is the suboptimal individual convergence. It is related to the presence of multiple agents in the environment and the consequent lack of convergence guarantee as already discussed. Suboptimal here may mean either nonconvergence to a Nash equilibrium, or in our case, convergence to a poor social equilibrium. Next we refer to some classical approaches remarking that most of them address the nonconvergence to Nash equilibria.

The case of zero-sum stochastic games[3] was discussed by [20] and attempts of generalizations to general-sum SG appeared in [13] (Nash-Q). Littman introduced Friend-or-Foe Q-learning (FFQ) [21], which learns to play Nash equilibria if the overall stochastic game has a global optimum or a saddle point. The algorithm requires that each agent is told whether it is facing a friend or a foe. In particular, the Friend-Q's guarantees are considerably weaker than the Foe-Q due to incompatible coordination equilibria.

Wang and Sandholm [36] propose the Optimal Adaptive Learning (OAL) algorithm, which creates virtual games for each matrix game. In these virtual games, suboptimal Nash equilibria are eliminated. This means that Nash equilibria of the original game must be computed in the first place. Although in practice Nash equilibria can be found for reasonably large games, it is unknown whether a Nash equilibrium can be found in worst-case polynomial time. Besides, virtual games are solved exponentially in the number of agents, and it assumes perfect monitoring, i.e., all agents observe all joint actions, thus turning it nonefficient for a high number of agents. This assumption is particularly strong and here we subscribe to the view of Stone and Veloso [33] who argue that complete communication reduces a multiagent system to a central process. On the other hand, OAL guaranteed finds the global optimum in fully cooperative SG.

Brafman and Tennenholtz [7] follow a model-based approach for the same problem. It assumes a priori coordination of the agents' learning processes (e.g., agreement over a joint exploration phase followed by a settlement on the joint policy that has yielded maximum reward). This results in a near-optimal polynomial-time algorithm in the number of actions of the agents. Recently, Kuminov and Tennenholtz [17] have introduced a near-optimal polynomial algorithm that considers imperfect monitoring in a two-player game where it is assumed that player 1 does not know the payoff matrices or the action taken by player 2; player 2, however, is fully informed about both the payoff matrices and the history of the game.

Suboptimal individual convergence can also be approached by some forms of nonexplicit biased exploration. Three are worth mentioning here as two are related to our hierarchical approach (Sect. 4.2), and one is applied in the IPD. The first was proposed in [38, 39] where Zhang et al. introduce a supervision framework to speed up the convergence of MARL algorithms. Hierarchically superior agents keep abstract states of lower-level agents. This view is used to generate rules (that agents must follow) or suggestions (these are optional), passed down to local agents. In the paper, it is not clear how rules are formed and whether or not they are domain-dependent.

The second form of biased exploration is due to Hines and Larson [12], who use repeated games (these are SG with a single state) where agents can follow the advice of a mediator that makes suggestions to the agents as to what actions to take. However, the authors do not deal with coordination games with multiple states, and the combination with Q-learning was left as future work.

Finally, it is worth to mention that biasing exploration has also been applied in the IPD. Babes et al. [3] show that agents playing the IPD can both lead (i.e., encourage adaptive opponents to cooperate), and follow (i.e., adopt a best-response strategy when paired with a fixed opponent. However, in this paper, the authors consider only two opponents, i.e., the spatial version is not explored.

4 Methods

As mentioned in the introduction, our aim is to investigate whether cooperation can emerge in the IPD when agents learn by reinforcement. From the previous sections, we have seen that mutual cooperation is not a Nash equilibrium in this game, and thus is related to suboptimal individual convergence. We believe that the exploration made by the agents has to be biased toward joint actions that yield higher social payoff. In the IPD, this means mutual cooperation.

Given this, we consider three different formalisms for the SG, including the two previously mentioned (hierarchical organization and coalition). In the third one (used for comparison purposes), agents just play and learn individually. Following the terminology proposed by Claus and Boutilier in [8], we call these agents independent learners (henceforth IL's).

In all cases, we use the spatial configuration proposed by [24] (a set N of agents placed on a square lattice), and Q-learning as learning method. The three formalisms are detailed in the next subsections.

4.1 Independent learners and the IPD

An n-agent IPD game is a tuple (N, S, A, R) where:

$N = 1, \ldots, j, \ldots, n$ is the set of agents
$S = \times S_j$ is the discrete state space (each S_j corresponds to the set of states of an agent j)

[3] In a game with zero sum, the sum of payoffs of all players is zero; the IPD for instance is a *nonzero* sum game as seen from Table 1.

Algorithm 1 Individual learning

```
for all j ∈ N do
    initialize Q-values, list of neighbors
end for
while not time out do
    for all j ∈ N do
        when in state s_j, select a random action with probability ε or greedy action with probability 1 − ε
    end for
    for all k in neighborhood of j do
        play a_j against a_k
        receive reward
        update Q_j {// (1)}
    end for
end while
```

$A = \times A_j$ is the discrete action space (each A_j corresponds to the set of actions of an agent j)
R_j is the reward function (R_j determines the payoff for agent j as $r_j : S_1 \times \cdots S_n \times A_1 \times \cdots \times A_n \to \mathbb{R}$)
T is the transition probability map (set of probability distributions over the state and action spaces).

For the specific case discussed here, the set of actions and respective payoffs (rewards) are as in Table 1 and agents play the IPD game with m other agents (neighbors) at each time step. The state is given by the payoff matrix.

As mentioned, IL's interact and learn by using Q-learning in an independent way. Hence, the main parameters are: the learning rate α, and the discount rate γ.

For action selection, we use ε-greedy: the agent selects the action with highest Q-value with probability $1 - \varepsilon$, and explores selecting an action randomly with probability ε. Depending on the action selected by each pair of agents, a reward is given to the agent, and the Q-value for the particular pair (s, a) is updated (1). In practice, this means that each agent plays m two-person games as it is common in n-person spatial IPD. Each agent updates its Q-table considering the rewards received by playing with the m interacting neighbors. This is formalized in Algorithm 1.

As discussed before, individual learning is not efficient in the IPD (see also results presented in Sect. 5). Hence, the next two sections discuss alternatives, starting with the one based on a hierarchy.

4.2 Playing the IPD in a hierarchy

The first approach for biasing exploration toward a socially higher reward is to use hierarchically superior agents to give recommendations to agents they supervise in a kind of organizational control. This has been used successfully by [6, 38, 39] in completely different domains, leading to more efficient equilibrium selection.

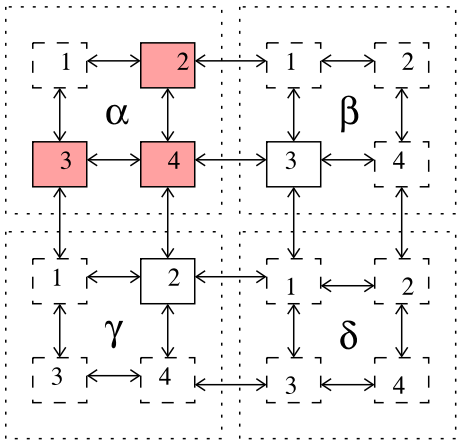

Fig. 1 Two-level organization: 16 agents $(\alpha_1, \ldots, \beta_1, \ldots, \delta_4)$ in the lower level, supervised by α, β, γ, and δ in the second level; *full-line boxes* mean agents with whom α_4 is interacting; *white boxes* mean defection (D)

The idea is to have two sets of agents: supervisors and low-level agents. The former supervise a group of the latter. Low-level agents behave basically as ILs (Sect. 4.1), unless they are given recommendations that regard action selection. This way a low-level agent L_j plays the IPD game repeatedly with m other neighbors. We stress that these neighbors are not necessarily those that belong to L_j's group, i.e., L_j's interactions transcend its own group. In fact, because each low-level agent L_j only communicates with its supervisor, it is not even necessary that L_j knows it belongs to a group.

As illustration, we refer to Fig. 1 in which low-level agents are divided in groups $(\alpha_1, \ldots, \alpha_4, \beta_1, \ldots, \beta_4, \gamma_1, \ldots, \gamma_4, \delta_1, \ldots, \delta_4)$ that are supervised by 4 supervisors $(\alpha, \beta, \gamma, \delta)$. In this figure, an arrow indicates that the two agents sharing it play the IPD. This, however, does not mean that they communicate explicitly as noncommunication between players is one of the assumptions underlying the IPD. Notice that despite the fact that a group exists (e.g., $\alpha_1, \ldots, \alpha_4$), each of these members have interactions outside the groups (e.g., α_4 also interacts with β_3). Conversely, α_4 does not interact with α_1. This is a real-world situation that makes the game more complex and the MARL more difficult. For instance in an organization, interactions not only occur inside a department; they also happen among agents from different departments. Otherwise, coordination would be much simpler.

The supervised learning works as in Algorithms 2 to 4, which are explained next.[4] Before, we remark that supervisors do not actually play the game, thus they are not included in the set N of low-level agents. In fact, supervisors must be seen as facilitators or tutors that will observe the local agents' in their groups from a broader perspective, and

[4] We drop the initialization steps as they are the same as in Algorithm 1.

Algorithm 2 Individual learning stage (stage 1)

1: **while** $t \leq \Delta_{ind}$ **do**
2: **for all** $L_j \in \mathcal{L}$ **do**
3: when in state s_j, select a random action with probability ε or greedy action with probability $1-\varepsilon$, and receive reward obtained by playing against each m-th neighbor
4: update Q_j^{ind} m times {// (1)}
5: **end for**
6: **for all** $S_i \in \mathcal{S}$ **do**
7: observe state, action, and reward for each L_j
8: compute the average reward $\overline{r}$ (among L_j's)
9: **if** tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$ not yet in the base of cases **then**
10: add tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$
11: **else**
12: **if** $\overline{r} > \overline{r_{old}}$ **then**
13: replace by tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$
14: **end if**
15: **end if**
16: **end for**
17: **end while**

eventually recommend actions to them. This recommendation is based on a group perspective, in opposition to the purely local perspective of low-level agents.

Besides the parameters already used by the IL, others that are now necessary are: the set of low-level agents $N = \mathcal{L} = \{L_1, \ldots, L_n\}$; the set $\mathcal{S} = \{S_1, \ldots\}$ of supervisor agents; the threshold τ (explained below); Δ_{ind} (time period during which each L_j learns and acts independently, updating the Q-table Q_j^{ind}); Δ_{tut} (time period during which each S_i prescribes an action to each L_j in its group based on cases observed so far); and Δ_{crit} (time period during which each L_j can act independently or follow the recommendation of the supervisor). These time periods are henceforth called stages 1, 2, and 3, respectively.

Stage 1 is described in Algorithm 2. During Δ_{ind} time steps, the N low-level agents play the IPD as ILs and their supervisors only observe them. Each $L_j \in N$ learns a policy; each supervisor S_i observes its low-level agents and records information to a base of cases. This information consists of *joint* states, *joint* actions, and rewards. Thus, this base is composed by the tuples $\langle \mathbf{s}, \mathbf{a}, \overline{r} \rangle$ where $\overline{r}$ is *averaged* over all supervised agents. The case that has yielded the highest $\overline{r}$ so far is kept in the base (line 13 of Algorithm 2).

The second stage (Algorithm 3) takes further Δ_{tut} time steps. In this stage, each S_i: (i) observes the joint state of its low-level agents; (ii) retrieves $\mathbf{a}^t$ for which $\overline{r}$ is the highest. It is important to note that in any case the local Q-tables continue to be updated. The main difference to stage 1 is that at stage 2, low-level agents are committed to the action prescribed by the supervisor, even when the expected reward is not as good as the computed Q-values.

Algorithm 3 Tutoring stage (stage 2)

1: **while** $\Delta_{ind} < t \leq \Delta_{ind} + \Delta_{tut}$ **do**
2: **for all** $S_i \in \mathcal{S}$ **do**
3: communicate with supervisor at upper level; get similar cases; add to case base
4: given $\mathbf{s}^t$, find $\mathbf{a}^t$ in case base for which $\overline{r}$ is highest; communicate a_j^p to each L_j {//where a_j^p is action prescribed by the supervisor for this agent}
5: **end for**
6: **for all** $L_j \in \mathcal{L}$ **do**
7: perform action a_j^p communicated by supervisor {or follow local policy if supervisor has not prescribed any action}
8: receive reward obtained by playing against each m-th neighbor
9: update Q_j^{ind} m times {// (1)}
10: **end for**
11: **for all** $S_i \in \mathcal{S}$ **do**
12: observe state, action, and reward for each L_j
13: compute the average reward (among L_j's) $\overline{r}$
14: **if** tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$ not yet in case base **then**
15: add tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$
16: **else**
17: **if** $\overline{r} > \overline{r_{old}}$ **then**
18: replace by tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$
19: **end if**
20: **end if**
21: **end for**
22: **end while**

In the third stage (which takes Δ_{crit} steps, as in Algorithm 4), low-level agents need not follow the prescribed action. Rather, after comparing the expected reward $\overline{r}$ that was communicated by the supervisor, with the locally computed Q-value for this particular prescribed action, each agent may select the action associated with its local policy. This means that the low-level agent will only select the prescribed action if this is at least as good as the expected Q-value (here considering a tolerance factor τ as in line 7 in Algorithm 4). No matter whether the low-level agents do follow the prescription or not, the supervisor is able to observe actions and rewards, and update its base of cases.

4.3 Playing the IPD in coalitions

The approach presented in the previous subsection has the drawback that groups must be given a priori, i.e., it works in static environments and/or when there is a clear grouping factor. It does not allow the emergence of such groups in a dynamic way.

One issue that has attracted many attention in multiagent systems is how to partition or organize a multiagent system in an effective way. Several approaches to this exist in the

Algorithm 4 Critique stage (stage 3)

while $\Delta_{ind} + \Delta_{tut} < t \leq \Delta_{ind} + \Delta_{tut} + \Delta_{crit}$ **do**
  for all $S_i \in \mathcal{S}$ **do**
    given $\mathbf{s}^t$, find $\mathbf{a}^t$ in case base for which $\overline{r}$ is maximal; communicate a_j^p to each L_J plus expected reward $\overline{r}$
  end for
  for all $L_j \in \mathcal{L}$ **do**
    {//compare Q_j^{ind} and r^e:}
    if $\overline{r} \times (1 + \tau) > Q_j^{ind}$ **then**
      perform a_j^p against each m-th neighbor {// where a_j^p is action prescribed by the supervisor for this agent}
      receive reward
      update Q_j^{ind} m times
    else
      perform a_j^{ind} against each m-th neighbor {// where a_j^{ind} is ε-greedy selected following local policy}
      receive reward
      update Q_j^{ind} m times
    end if
  end for
  for all $S_i \in \mathcal{S}$ **do**
    observe state, action, and reward for each L_j
    compute the average reward (among L_j's) $\overline{r}$
    if tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$ not yet in case base **then**
      add tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$
    else
      if $\overline{r} > \overline{r_{old}}$ **then**
        replace by tuple $\langle \mathbf{a}^t, \mathbf{s}^t, \overline{r} \rangle$
      end if
    end if
  end for
end while

multiagent systems literature, but here we focus on coalition formation because it is a well-established approach from game theory, having solid mathematical grounds. Unfortunately, partitioning agents in coalitions that lead to an efficient utility is not a trivial problem. In the general case, the number of coalition structures ($O(|N|^{|N|})$) is so large that it cannot be enumerated for more than a few agents [30]. Therefore, it is necessary to use domain knowledge and/or games with particular structures and where agents have particular characteristics (e.g., they form a network in which the neighborhood plays a role) to solve the problem of coalition formation in a reasonable efficient way. For example, coalitions among neighbors make sense and help them to collect a much higher payoff. In the spatial IPD game, only coalitions among neighboring agents are initially formed. Thus, the number of coalition structures is manageable (it is much smaller than $|N|^{|N|}$). This does not mean that coalitions are restricted to four or five agents. Rather, they may grow as agents in the initially formed coalitions may propose to their immediate neighbors to join and so forth.

These facts have motivated our second approach for achieving a socially higher reward in the IPD, namely the formation of coalitions of cooperators. A preliminary version of this approach was tested with positive results in [27], where the focus was to compare Q-learning and learning automata [23] techniques.

The coalitional approach is also based on ILs. These, however, have here a different set of actions to choose from. Instead of just selecting C or D as described in the two previous subsections, now actions are to act as IL and play C, to act as IL and play D, and to be in a coalition or not. Agents may leave the coalition whenever they want, thus becoming an IL.

When belonging to a coalition, an agent cooperates with other members of the coalition, thus it plays C. The action to be played with non-members (outsiders) is decided collectively, by means of a voting process. Hereby, each agent votes to play the action which is the best according to its own individual Q-table. Each vote is weighted by its Q-value. This has the same effect as if the whole coalition would keep a Q-table with the sum of Q-values over all its agents, followed by a greedy action selection. Again, this is used only when playing against outsiders. Inside a coalition, the agent does not have to decide which is the action against its coalition mates, as it is assumed that they all cooperate. This assumption is a reasonable one because since actions are public inside the coalition, noncooperators would be seen as someone betraying their coalition members. This would cause the "black sheep" to be expelled from the coalition and suffer retaliation (D) in future plays.

Although this procedure is simple, it has been used in a similar way in, e.g., [15]. In this particular case, however, every agent locally chooses the action yielding the maximum value, and from these maximum values, the action corresponding to the highest value is chosen.

In our case, each agent decides which is the best action to take, based on the local policy (Q-value). After the agents have individually done that, they vote and the action that receives more votes is the one that the coalition is going to perform. Algorithm 5 indicates how the learning proceeds.

5 Experiments and analysis

For each of the formalisms introduced in Sects. 4.1 to 4.3, we have run experiments using the same payoff matrix and spatial configuration. The results of these experiments are presented and analyzed next.

Table 2 summarizes the parameters used and their values. A star indicates that values were varied and are then

Algorithm 5 Coalition learning

```
for all L_j ∈ N do
    initialize Q-values, list of neighbors
end for
while not time out do
    when in state s_j, select greedy action a_j with probability 1 − ε or a random action with probability ε
    if L_j's selected action is to be in coalition then
        join coalition
        play C with coalition members
        vote to select how to play with outsiders
        play winner action with outsiders
    else
        play as IL
    end if
    receive reward and update Q-values (m times) (1)
end while
```

Table 2 Parameters and their values

Parameter	Description	Value
T	temptation to defect	5
R	reward for mutual cooperation	3
S	sucker's payoff	0
P	punishment for mutual defection	1
$N = \|\mathcal{L}\|$	number of agents	$*$
m	nb. neighbors	4
α	learning coefficient	0.5
γ	discount rate	0
ε	greedy action selection	$*$
g	size of group	4
Δ_{ind}	stage 1	50
Δ_{tut}	stage 2	10
Δ_{crit}	stage 3	40
τ	intolerance factor	$*$

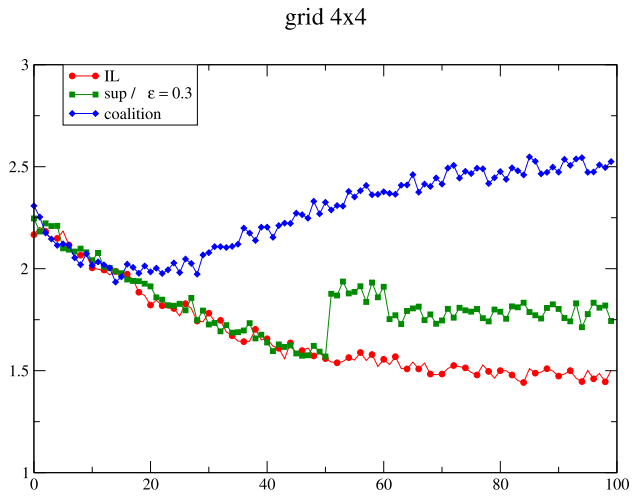

Fig. 2 Grid 4×4: Average *reward* along *time*, for independent learners, supervised learning, and coalition-based learning ($\tau = 0$)

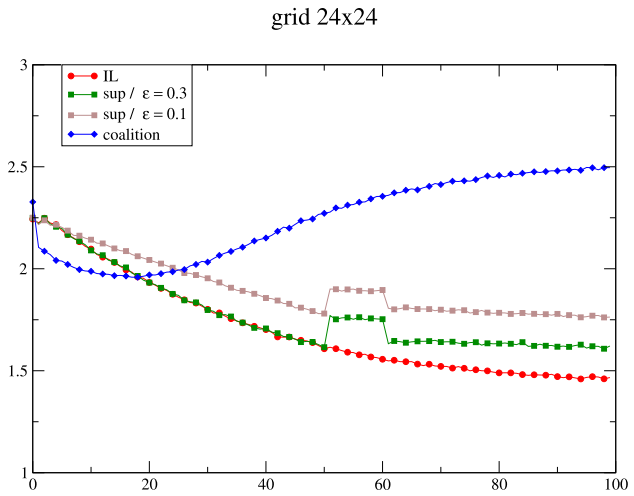

Fig. 3 Grid 24×24: Average *reward* along *time*, for independent learners, supervised learning, and coalition-based learning ($\tau = 0$)

reported in the appropriate section. We indicate some parameters that are specific of the supervision-based method (light grey). The others were used in the three variants.

We have performed experiments with two different grid sizes: 4×4 and 24×24. This way, $N = 16$ and $N = 576$ respectively. This aimed at demonstrating that the pattern remains, no matter the number of agents. The values T, R, S, and P for the payoff matrix (Table 1) are given in Table 2 and are commonly used in experiments regarding the IPD.

All experiments were repeated 100 times and the simulations run for 100 time steps or 100 action selections. Although we do not show error bars in the resulting plots, the standard deviations are at most 20% for the grid of size 4 and less than 5% in the case of the grid size 24.

To demonstrate that ILs perform poorly because they end up learning the Nash equilibrium and thus converging to mutual defection, we have run the first series of experiments changing N and also ε. Unless noted, we give results for $\varepsilon = 0.3$ but note that the pattern of mutual defection does not change significantly. In fact, the exploration rate cannot prevent this behavior, as already noticed in [29].

Figure 2 depicts how the average reward over all $N = 16$ agents changes along time. Similarly, Fig. 3 plots this evolution for $N = 576$.

Mutual cooperation would lead to an average reward of $R = 3$, while mutual defection leads to average reward of $P = 1$. The learning curve for the ILs is marked by circles in Fig. 2. Observing this curve, we notice that the value of this reward at step 100 is above 1. This happens due to the exploration that agents still perform as ε was not decreased with time (no annealing). This means that on average half of the $\varepsilon \times N$ agents were cooperating by chance, which yields

a payoff of $T = 5$ to the opponent, thus slightly increasing the average reward.

The approach based on supervision improves this picture (plots marked with squares) but not to the extent that was verified in other games and scenarios (e.g. [6] and in another attempt using coordination games). The reason is that supervision is more efficient to guide agents to equilibrium selection when more than one exists. In coordination games for instance, where two or more equilibria exist, the bias imposed by the supervisor can guide agents to a more efficient selection as the supervisor is able to record good coordinated actions (and recommend them later), while avoiding the recording of miscoordinations, which are then not recommended.

In the IPD, the problem is that the supervisors observe few mutual cooperations,[5] recommend them, but once the supervised agents select C and are defected, they tend to reject this recommendation in the future.

This is exactly what is seen in Fig. 2, looking at the curve marked with squares. During stage 1 (first $\Delta_{ind} = 50$ time steps), the behavior is the same as for the ILs already discussed. During this time, the supervisors eventually record good cases occurred in the group they supervise. For the next $\Delta_{tut} = 10$ steps, these good cases are recommended, low-level agents must select such actions (this is mandatory in stage 2), but the reward achieved is not high enough to drive individual agents away from a higher expected payoff associated with defection. Hence, once this period is over (around $t = 60$), low-level agents tend to refuse recommendations. We have performed simulations with various values for τ but no tolerance level is able to keep agents from defecting.

The same discussion is valid for $N = 576$ (Fig. 3). Besides the curve for $\varepsilon = 0.3$, the curve for $\varepsilon = 0.1$ is also shown as it has led higher rewards in stage 1, when less agents explore. Hence, some mutual cooperation remains for a longer time. Supervisors are able to observe better cases and recommend them thus improving the cooperation level. A similar behavior is not observed for small N since here, the influence of a single defection is much higher than when N is big.

The lesson taken from this set of experiments is that supervision improves the picture over the IL but not as much as it would be desired, namely a value closer to R.

The approach using coalition, as expected, is more efficient as it explores the flexibility of the emergence of groups that indeed are willing to cooperate because this has proven good in the past (otherwise the ε-greedy action selection would not lead agents to cooperate). Comparing this variant to the one based on supervision the differences are clear.

[5]The tendency to observe full mutual cooperation in the group decreases with the increase in the group size.

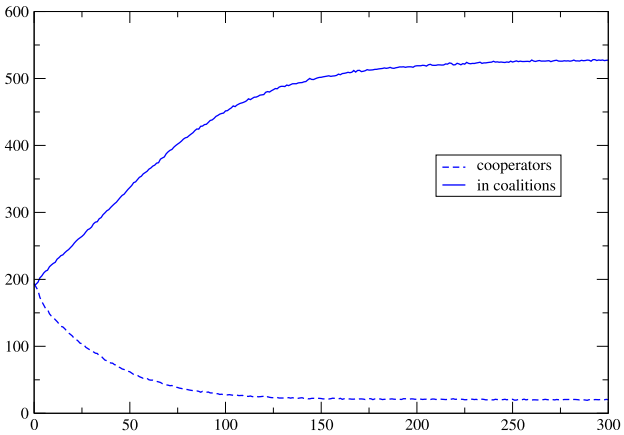

Fig. 4 Grid 24 × 24: Number of cooperators and number of agents that form coalitions, along time

In the latter, if an agent is willing to cooperate but happens to be in a "bad" group (regarding behavior), it will learn to defect on the neighbors as well. However, the coalition is a much more flexible structure that emerges only among those that have experienced cooperation as rewarding in the past and thus want to continue following this action.

Results corroborate this. In both Fig. 2 and Fig. 3, one sees that the coalition approach ends up rewarding the collectivity. It starts with the worst performance among the three approaches (curves marked with diamonds) because agents are exploring the possibilities (and they have more actions to explore). But it establishes itself as supportive of cooperation. Moreover, this happens relatively early (around time step 30). From this point on, the number of agents belonging to coalitions increase and so the average reward. The fact that this average reward does not fully reach the value of $R = 3$ (it falls 0.5 short), is explained by two issues. First, experimentation is still performed (with probability ε); second, clusters of defections establish that are difficult to break.

Figure 4 shows the number of agents in coalition and the number of cooperators along time for the grid 24 × 24. Toward the end of the simulation, almost all cooperators belong to coalitions. Therefore, the difference between the total number of agents (576) and the cooperators (both in coalitions and acting independently) corresponds to the number of defectors. These are few as it can be observed.

6 Conclusions and future work

Despite the obvious limitations and simplifications, the two-agent PD can be seen as an abstraction of very simple social situations that deal with cooperation. However, this game can be better appreciated in the spatial and repeated version, the so-called n-player IPD. This is played in a square lattice.

Previous results using some forms of social attachment (e.g., spatial neighborhood) reported a reasonable rate of cooperation. However, when pure RL is used, cooperation vanishes because agents learn that mutual defection is the Nash equilibrium for this game.

The present paper has then departed from the use of pure RL, claiming that some form of social attachment must be employed to bias the exploration agents perform during the learning process. Two kinds of biases were tried. First, we have used a method that has proven successful in other kinds of games, namely the supervised learning. The second method is by allowing agents to form coalitions that sustain cooperation.

In order to validate these methods, a series of experiments using them were performed. In such experiments, we have changed the number of agents, the exploration rate, and the values of other parameters. Results show that both methods were able to depict mutual cooperation. However, the rate of cooperation was higher when coalitions were used (e.g., the average reward agents received was close to the highest possible, R). The reason for the coalitional method performing better than the supervision-based one is twofold. First coalitions may *emerge* among agents that have experimented some benefits in past encounters. Groups in the supervised learning are fixed, thus they do not fully support the dynamics of the game. The second reason is that the supervision-based approach works better for the cases in which more than one equilibria exist—which is not the case of the IPD—and their selection must be coordinated.

These results show that learning to cooperate is feasible when the exploration is biased. Bias may stem either by some agent(s) having a broader view (and hence knowledge) as it is the case in the hierarchical method, or by agents forming groups that support mutual cooperation. Obviously, the latter is preferred but we remark that there are other games, e.g., the coordination game mentioned, where the hierarchical approach is more efficient given that it helps agents to distinguish between two or more equally good equilibria.

The methods and results presented and discussed here can be employed in scenarios that involve data networks such as P2P, sensor, and vehicular networks. In all of these cases, dealing with free-riders and malicious agents is an important issue. These issues are partially discussed in [9, 10, 18, 34].

As future work, it is intended to investigate reputation among agents, as this continues in the direction of bringing in the IPD game more issues that are socially backed. This could be important, for instance, when agents have to decide which action to play against outsiders. If these outsiders have a reputation degree, this could be used in such decision-making. Also, reputation may turn important among members of coalitions themselves, for instance to decide when coalitions must be dissolved.

References

1. Abramson G, Kuperman M (2001) Social games in a social network. Phys Rev E 63
2. Axelrod R (1984) The evolution of cooperation. Basic Books, New York
3. Babes M, Cote EMD, Littman ML (2008) Social reward shaping in the prisoner's dilemma. In: Padgham L, Parkes D, Müller J, Parsons S (eds) Proc. of the 7th int. joint conf. on aut. agents and multiagent systems, IFAAMAS, May 2008, pp 1389–1392
4. Bazzan ALC, Bordini RH (2001) A framework for the simulation of agents with emotions: Report on experiments with the iterated prisoners dilemma. In: Müller JP, Andre E, Sen S, Frasson C (eds) Proceedings of the fifth international conference on autonomous agents, Montreal, Canada, May 2001. ACM, New York, pp 292–299
5. Bazzan ALC, Bordini RH, Campbell JA (1999) Moral sentiments in multi-agent systems. In: Intelligent agents V. Lecture notes in artificial intelligence, vol 1555. Springer, Berlin, pp 113–131. Also appeared as Proc. of the workshop on agent theories, architecture and languages (ATAL98), Paris, July 1998
6. Bazzan ALC, de Oliveira D, da Silva BC (2010) Learning in groups of traffic signals. Eng Appl Artif Intell 23:560–568
7. Brafman RI, Tennenholtz M (2002) Efficient learning equilibrium. In: NIPS, pp 1603–1610
8. Claus C, Boutilier C (1998) The dynamics of reinforcement learning in cooperative multiagent systems. In: Proceedings of the fifteenth national conference on artificial intelligence, pp 746–752
9. Costa-Montenegro E, Burguillo-Rial JC, González-Castaño FJ, Vales-Alonso J (2007) Agent-controlled sharing of distributed resources in user networks. In: Lee RST, Loia V (eds) Computational intelligence for agent-based systems. Studies in computational intelligence, vol 72. Springer, Berlin, pp 29–60
10. Costa-Montenegro E, Burguillo-Rial JC, Gil-Castiñeira F, González-Castaño FJ (2011) Implementation and analysis of the bittorrent protocol with a multi-agent model. J Netw Comput Appl 34:368–383
11. Fulda N, Ventura D (2007) Predicting and preventing coordination problems in cooperative Q-learning systems. In: Proceedings of the 20th international joint conference on artificial intelligence (IJCAI), pp 780–785
12. Hines G, Larson K (2008) Learning when to take advice: A statistical test for achieving a correlated equilibrium. In: McAllester DA, Myllymäki P (eds) UAI. AUAI Press, Menlo Park, pp 274–281
13. Hu J, Wellman MP (1998) Multiagent reinforcement learning: Theoretical framework and an algorithm. In: Proc. 15th international conf. on machine learning. Kaufmann, Los Altos, pp 242–250
14. Huberman BA, Glance NS (1993) Evolutionary games and computer simulations. Proc Natl Acad Sci USA 90:7716–7718
15. Humphrys M (1997) Action selection methods using reinforcement learning. PhD thesis, Cambridge
16. Kim BJ, Trusina A, Holme P, Minnhagen P, Chung JS, Choi MY (2002) Dynamic instabilities induced by asymmetric influence: Prisoner's dilemma game in small-world networks. Phys Rev E 66
17. Kuminov D, Tennenholtz M (2008) As safe as it gets: Near-optimal learning in multi-stage games with imperfect monitoring. In: Proceeding of the ECAI. IOS Press, Amsterdam, pp 438–442
18. Lin R, Kraus S, Shavitt Y (2007) On the benefits of cheating by self-interested agents in vehicular networks. In: Proceedings of the 6th international joint conference on autonomous agents and multiagent systems (AAMAS 2007). ACM, New York, pp 327–334
19. Lindgren K, Nordahl M (1994) Evolutionary dynamics of spatial games. Physica D 75:292–309

20. Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th international conference on machine learning, ML, New Brunswick, NJ. Kaufmann, Los Altos, pp 157–163
21. Littman ML (2001) Friend-or-Foe Q-learning in general-sum games. In: Proceedings of the eighteenth international conference on machine learning (ICML01), San Francisco, CA, USA. Kaufmann, Los Altos, pp 322–328
22. Mailath G, Samuelson L, Shaked A (1993) Correlated equilibria as network equilibria. Discussion paper, University of Bonn
23. Narendra KS, Thathachar MAL (1989) Learning automata: an introduction. Prentice-Hall, Upper Saddle River
24. Nowak MA, May RM (1992) Evolutionary games and spatial chaos. Nature 359:826–829
25. Ortony A, Clore GL, Collins A (1988) The cognitive structure of emotions. Cambridge University Press, Cambridge
26. Panait L, Luke S (2005) Cooperative multi-agent learning: The state of the art. Auton Agents Multi-Agent Syst 11(3):387–434
27. Peleteiro A, Burguillo JC, Bazzan ALC (2010) Enhancing cooperation in the ipd with learning and coalitions. In: Proc. of the 2nd Brazilian workshop on social simulation, S. Bernardo do Campo. SBC, Porto Alegre
28. Sandholm T (2007) Perspectives on multiagent learning. Artif Intell 171(7):382–391
29. Sandholm TW, Crites RH (1995) Multiagent reinforcement learning in the iterated prisoner's dilemma. Biosystems 37:147–166
30. Sandholm T, Larson K, Andersson M, Shehory O, Tohmé F (1999) Coalition structure generation with worst case guarantees. Artif Intell 111(1–2):209–238
31. Shoham Y, Powers R, Grenager T (2007) If multi-agent learning is the answer, what is the question? Artif Intell 171(7):365–377
32. Stone P (2007) Multiagent learning is not the answer. It is the question. Artif Intell 171(7):402–405
33. Stone P, Veloso M (2000) Multiagent systems: A survey from a machine learning perspective. Auton Robots 8(3):345–383
34. Vinyals M, Rodríguez-Aguilar JA, Cerquides J (2011) A survey on sensor networks from a multiagent perspective. Comput J 54:455–470
35. Vrancx P, Tuyls K, Westra RL (2008) Switching dynamics of multi-agent learning. In: Padgham L, Parkes D, Müller J, Parsons S (eds) Proceedings of the 7th international joint conference on autonomous agents and multiagent systems, Estoril, vol 1. pp 307–313
36. Wang X, Sandholm T (2002) Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In: Advances in neural information processing systems (NIPS-2002), vol 15
37. Watkins CJCH, Dayan P (1992) Q-learning. Mach Learn 8(3):279–292
38. Zhang C, Abdallah S, Lesser VR (2008) Efficient multi-agent reinforcement learning through automated supervision (extended abstract). In: Padgham L, Parkes D, Müller J, Parsons S (eds) Proceedings of the 7th international joint conference on autonomous agents and multiagent systems, Estoril, vol 3. pp 1365–1368
39. Zhang C, Abdallah S, Lesser V (2009) Integrating organizational control into multi-agent learning. In: Sichman JS, Decker KS, Sierra C, Castelfranchi C (eds) Proceedings of the 8th international conference on autonomous agents and multiagent systems (AAMAS), Budapest, Hungary