Multiagent Reinforcement Learning with Spiking and Non-Spiking Agents in the Iterated Prisoner's Dilemma

Vassilis Vassiliades, Aristodemos Cleanthous, and Chris Christodoulou

Department of Computer Science, University of Cyprus,
75 Kallipoleos Ave., P.O. Box 20537, 1678 Nicosia, Cyprus
{v.vassiliades,aris,cchrist}@cs.ucy.ac.cy

Abstract. This paper investigates Multiagent Reinforcement Learning (MARL) in a general-sum game where the payoffs' structure is such that the agents are required to exploit each other in a way that benefits all agents. The contradictory nature of these games makes their study in multiagent systems quite challenging. In particular, we investigate MARL with spiking and non-spiking agents in the Iterated Prisoner's Dilemma by exploring the conditions required to enhance its cooperative outcome. According to the results, this is enhanced by: (i) a mixture of positive and negative payoff values and a high discount factor in the case of non-spiking agents and (ii) having longer eligibility trace time constant in the case of spiking agents. Moreover, it is shown that spiking and non-spiking agents have similar behaviour and therefore they can equally well be used in any multiagent interaction setting. For training the spiking agents, a novel and necessary modification enhances competition to an existing learning rule based on stochastic synaptic transmission.

1 Introduction

Multiagent Reinforcement Learning (MARL) is a problem that has been studied extensively during the last few years. The problem lies in the dynamic environment created by the presence of another learner. In MARL there could be different kinds of situations: fully competitive (which could be modelled with zero-sum games), fully cooperative (which could be modelled with team games) or a mixture of both (which could be modelled with general-sum games). Each situation has different problems, so an active community has been designing algorithms to address all of them. Some examples include minimax-Q [1], Nash-Q [2], Joint Action Learners [3], FoF-Q (Friend-or-Foe Q) [4], WoLF-IGA (Win or Lose Fast - Infinitesimal Gradient Ascent) [5], CE-Q (Correlated Equilibria Q) [6], FMQ (Frequency Maximum Q) [7], GIGA-WoLF (Generalised IGA - WoLF) [8] and AWESOME (Adapt When Everybody is Stationary Otherwise Move to Equilibrium) [9][1]. A lot of work is focused in deriving theoretical guarantees,

[1] For a comprehensive coverage of MARL algorithms see [10] and references therein.

based on different sorts of criteria such as rationality and convergence [11] or targeted-optimality, safety and auto-compatibility [12]. Since the problem is not very well-defined, Shoham et al. [13] attempted to classify the existing work by identifying five distinct research agendas. They argued that when researchers design algorithms they need to place their work under one of these categories. Subsequently some work did focus on specific agendas (e.g., [14]), but more agendas were proposed [15]. In addition, the original agendas [13] have been criticised that they may not be distinct, since they may complement each other ([16], [17]). Stone [18] extended the criticism by arguing that the game theoretic approach is not appropriate in complex multiagent problems. Despite these criticisms, our study lies in the original prescriptive non-cooperative agenda [13], which asks how the agents should act to obtain high rewards for a given environment.

Reinforcement learning (RL) has successfully been applied to spiking neural networks (NNs) in recent years. These techniques try to incorporate reward distribution according to the biological processes of neurons. Although their degree of experimental justification varies and they need to be further assessed, all these methods are biologically plausible and provide the basis for applying RL on biologically realistic neural models as well as the inspiration for further and better integration of RL into the spiking models. Imaginative approaches to the subject include: (i) the reinforcement of irregular spiking [19], where the learning rule performs stochastic gradient ascent on the expected reward by correlating the fluctuations in irregular spiking with a reward signal and (ii) the employment of a spike-timing-dependent synaptic plasticity rule [20], in order to achieve RL by modulating this plasticity through reinforcement signals ([21],[22]). In our study, we use a variation of Seung's RL on spiking NNs [23], which reinforces the stochasticity present in the process of synaptic transmission. To the best of our knowledge, this is the first time that a spiking neural model with biologically plausible learning simulates a game theoretical situation.

The current study investigates cooperation between self-seeking reward agents in a non-cooperative setting. This situation can be modelled with the Iterated Prisoner's Dilemma (IPD) which is a general-sum game. Although the cooperative outcome is a valid equilibrium of the IPD, our study does not aim to assess the strength of the learning algorithms to attain equilibria of the game or best responses to any given strategy. Instead, we focus on mutual cooperation and see whether it can be achieved by spiking and simple non-spiking agents trained with RL and attempt to compare them. It is very interesting and beneficial to understand how and when cooperation is achieved in the IPD's competitive and contradictive environment, as it could then become possible to prescribe optimality in real life interactions through cooperation, analogous to the IPD. In its standard one-shot version, the Prisoner's Dilemma [24] is a game summarized by the payoff matrix of Table 1. There are 2 players, Row and Column. Each player has the choice of either to "Cooperate"(C) or "Defect" (D). For each pair of choices, the payoffs are displayed in the respective cell of the payoff matrix of Table 1. In game theoretical terms, where rational players are assumed, DD is the only Nash equilibrium outcome [25], whereas only the cooperative (CC) out-

Table 1. Payoff matrix of the Prisoner's Dilemma game with the values used in our experiments. Payoff for the Row player is shown first. R is the "reward" for mutual cooperation. P is the "punishment" for mutual defection. T is the "temptation" for unilateral defection and S is the "sucker's" payoff for unilateral cooperation. The only condition imposed to the payoffs is that they should be ordered such that T>R>P>S.

	Cooperate (C)	Defect (D)
Cooperate (C)	R(=4),R(=4)	S(=-3),T(=5)
Defect (D)	T(=5),S(=-3)	P(=-2),P(=-2)

come satisfies Pareto optimality [26]. The "dilemma" faced by the players in any valid payoff structure is that, whatever the other does, each one of them is better off by defecting than cooperating. The outcome obtained when both defect however is worse for each one of them than the outcome they would have obtained if both had cooperated. In the IPD, an extra rule (2R>T+S) guarantees that the players are not collectively better off by having each player alternate between C and D, thus keeping the CC outcome Pareto optimal. Moreover, contrary to the one shot game, CC can be a Nash equilibrium in the infinite version of the IPD.

As pointed out in [27] "perfectly predicting the environment is not enough to guarantee good performance", because the level of performance depends partly on properties of the environment. In our case, we believe that the property of the environment which plays a significant role in the CC outcome is the reward function, since it specifies the type and strength of the reinforcement the agents receive. By experimenting with the payoff matrix we observed that for the non-spiking agents it is beneficial to mix positive and negative values. The payoff values used in both spiking and non-spiking simulations are shown in Table 1.

The remainder of the paper is organised as follows. Section 2 describes our methodology for both spiking and non-spiking simulations. The results are presented and analysed in Section 3, while the last section gives the conclusions.

2 Methodology

2.1 Agents as Lookup Tables and Non-Spiking Neural Networks

We mainly focus on Q-learning [28], a simple reinforcement learning algorithm. There are two ways of storing the Q values (i.e., the estimates of how good the state-action pairs are), either in a lookup table (LT) or inside a function approximator, such as a NN. When the search space is small, LTs are preferred, because they are much faster. However, for large search spaces, function approximators are the only choice, because it is impossible to store every possible state-action pairs, due to memory constraints. According to Sandholm and Crites [29], LTs yield better results and are faster than simple recurrent NNs in the IPD. The multiagent system can be either *homogeneous* or *heterogeneous*. A homogeneous system exists when both agents employ the same strategy or learning algorithm, with the same or similar parameters, which in our case are the discount factor

(γ) and the memory type which could either be an LT or a feedforward NN. In contrast, a heterogeneous system exists when either both agents employ the same algorithm with dissimilar parameters, or different algorithms. The opposing agents could be TD (Temporal Difference) [30], SARSA (State-Action-Reward-State-Action) [31] or Q-learning. The agents are provided only with incomplete information: they receive only the state of the environment, i.e., the actions of both the agent and the opponent in the previous round, but not the payoffs associated with each action. A Boltzmann exploration schedule is utilised, as it gives a good balance between exploration and exploitation. Specifically, an action a_i is selected from state s with probability $p(a_i)$ given by equation (1):

$$p(a_i) = \frac{e^{Q(s,a)/t}}{\sum_{a \in \{C,D\}} e^{Q(s,a)/t}} \tag{1}$$

where the temperature t is given by $\mathrm{t} = 1 + 10 \times 0.995^n$, with n being the number of games played so far. The constants 1, 10 and 0.995 are chosen empirically. Each NN input is represented by 4 bits (thus 4 nodes), because according to the experiments in [29], this "unary encoding resulted in faster learning than a two-bit binary" one. Each bit is active depending on whether the previous action of the agent and the opponent was C or D. The network has 2 linear outputs corresponding to the estimates of the actions available in the game. One hidden layer with 3 sigmoidal activation function units is used and the network is trained by backpropagation as in [32], but with a single network.

2.2 Agents as Spiking Neural Networks

The game simulation is repeated with the two players implemented by two spiking NNs. The networks' architecture is depicted in Fig. 1. Each network has two layers of leaky integrate-and-fire (LIF) neurons. The equation and values of the parameters used for modelling the LIF neurons are the same as in [23], apart from the value of the mean weight of the conductance used for the excitatory synapses which is set to 14nS. Both networks receive a common input of 60 Poisson spike trains grouped in four neural populations. The networks learn simultaneously but separately where each network seeks to maximise its own accumulated reward. Learning is implemented through reinforcement of stochastic synaptic transmission as in [23], where the model is developed along the hypothesis that microscopic randomness is harnessed by the brain for the purposes of learning. Briefly, within the model's framework, each synapse acts as an agent pursuing reward maximisation through the actions of releasing or not a neurotransmitter upon arrival of a presynaptic spike. Each synapse records its recent actions through a dynamical variable, the eligibility trace [33], the time constant of which essentially reflects the way each synapse integrates time-related events. The input to the system is presented for 500ms and encodes the decisions the two networks had at the previous round. One can identify here a cyclic procedure which starts when the networks decide, continues by feeding this information

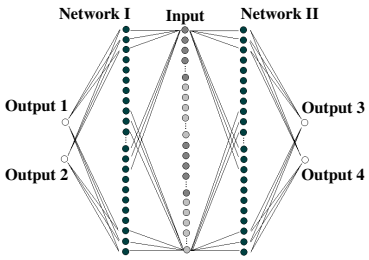

Fig. 1. Two spiking Neural Networks of Hedonistic Synapses and multilayer-type architecture compete in the IPD. Each network has two layers of hedonistic synapses that make full feedforward connections between three layers of neurons: the 60 shared input neurons, 60 leaky integrate-and-fire (LIF) hidden neurons and two LIF output neurons, randomly chosen to be either excitatory or inhibitory.

to the networks during which learning takes place and ends by a new decision. Each network's decision is encoded in the input, by the firing rate of two groups of Poisson spike trains. The first group will fire at 40Hz if the network cooperated and at 0Hz otherwise. The second group will fire at 40Hz if the network defected and at 0Hz otherwise. Consequently, four groups of Poisson spike trains provide the system's input with two groups always being active, preserving thus a balance at the output neurons' firing rates at the beginning of learning. Any significant difference in the output neurons' firing rate at any time should only be induced by learning and not by the differences of the driving input firing rates. At the end of each learning round the networks decide whether to cooperate or defect for the game's next round, according to the value each network assigns to the two actions. These values are reflected by the output neurons' firing rates at the end of each learning round. The cooperation value for network I and II is taken to be proportional to the firing rate of output neurons 1 and 3 respectively. Similarly, the defection value for network I and II is taken to be proportional to the firing rate of output neurons 2 and 4 respectively. When the two networks decide their play for the next round of the IPD, they each receive a distinct payoff given their actions and according to the game's payoff matrix (see Table 1). This same payoff is also the global reinforcement signal (scaled down) that will train each network during the next learning round and thus guide the networks to their next decisions. Since the learning algorithm works with positive and negative reinforcements that are directly applied to the synapses and are extracted from the payoff matrix, it is then necessary that the payoff matrix contains both positive and negative values. Each network is reinforced for every spike of their output neuron that was "responsible" for the decision at the last round and hence for the payoff received. The networks thus learn through global reinforcement signals which strengthen the value of an action that elicited a reward and weaken the value of an action that resulted to a penalty.

In order to enhance competition between output neurons during a learning round, Seung's algorithm [23] is extended with additional global reinforcement

signals administered to the networks for every output neuron spike not "responsible" for the decision. In the CD case for example, an additional reward of +1.15 is provided to network I for every spike of output neuron 2 and an additional penalty of -1.15 is provided to network II for every spike of output neuron 3. The value of the action that was not chosen by each network is therefore also updated, by an opposite in sign reinforcement signal. The value of 1.15 applies to all outcomes and is chosen to be small enough such that: (i) any changes to the values of the players' actions are primarily induced by the reinforcement signals provided by the payoff matrix and (ii) not to cause IPD rules' violation.

Overall during a learning round, each network receives global, opposite in sign reinforcements for spikes of both of its output neurons. One of the two signals is due to the game's payoff matrix and its aim is to "encourage" or "discourage" the action that elicited reward or penalty and the other signal is complementary and its aim is to "encourage" or "discourage" the action that could have elicited reward or penalty if it had been chosen in the previous round of the game.

3 Results and Discussion

3.1 Lookup Table and Non-Spiking Neural Network Agents

The step-size parameter α, the backpropagation learning rate η and the discount factor γ were empirically set to 0.9 for all agents, as this showed good behaviour with fast convergence, especially when using LTs. Low α and η values, such as 0.1, prevented in some cases the algorithms from converging to the CC outcome. This may be due to the exploration schedule and more specifically to t (see eqn. 1) dropping at a level where no further exploration takes place. The games were run for 50 trials with 2000 rounds per trial.

As mentioned in Section 2.1, a multiagent learning system may either have homogeneous or heterogeneous settings. In homogeneous settings we test 3 Q-agents in self-play: 1) Q(LT, γ=0.9); 2) Q(LT, γ=0.1) and 3) Q(NN, γ=0.9). In heterogeneous settings we test 6 cases: 1) Q(LT, γ=0.9) vs TD(LT, γ=0.9) - different algorithm; 2) Q(LT, γ=0.9) vs SARSA(LT, γ=0.9) - different algorithm; 3) Q(LT, γ=0.9) vs Q(LT, γ=0.1) - different γ; 4) Q(NN, γ=0.9) vs Q(LT, γ=0.9) - different memory type; 5) Q(NN, γ=0.9) vs TD(LT, γ=0.9) - different algorithm and memory type and 6) Q(NN, γ=0.9) vs SARSA(LT, γ=0.9) - different algorithm and memory type [2]. Fig. 2 depicts the results taken when ranking all these cases based on the percentage of CC. The results suggest that normally more cooperation can be achieved in homogeneous environments. Moreover, LTs with high γ obtain better results (93%) than (i) NNs with high γ (67%) and (ii) LTs with low γ (49%). This may be because (i) LTs are simpler and more suitable than NNs in our simple environment and (ii) a higher γ

[2] It is worth noting that we could not evaluate SARSA and TD agents with NNs, nor Q with NN and a low γ (=0.1) or in some cases with many hidden nodes such as 10, because the algorithm diverges, since the NN weights are driven to infinity. Divergence problems are known to exist in RL with function approximators and some solutions were proposed to deal with them (see [34]; [35] and references therein).

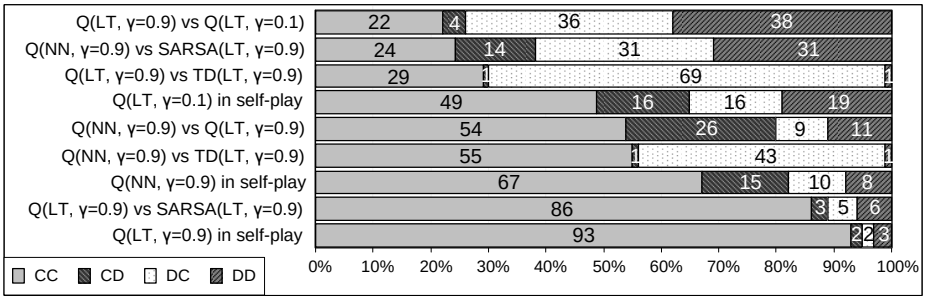

Fig. 2. Performance of Q vs Learning agents in homogeneous and heterogeneous environments ranked based on the CC percentage. Highest CC is achieved when an LT farsighted Q-agent (i.e., with a high discount factor γ) competes in self-play.

makes the agents take future rewards into account more strongly, which leads to an enhanced mutual cooperation. A successful heterogeneous case is the one where a Q agent competes with a SARSA, both with LTs and γ=0.9; in this case CC gets 86%. However, when the Q agent uses a NN, CC drops dramatically to 24% and DD rises from 6% to 31%. The "strongest" algorithm in our case (i.e., Q-learning) accumulates more reward, since DC occurs in 31% of the rounds, whereas CD only in 14%. TD can be considered very weak when evaluated against Q-learning, as the latter manages to learn a better policy that exploits the weakness of its opponent. This is illustrated by the high DC percentages, in contrast with the low CD percentages. The interesting case of Q agents with LTs, with the row player being more farsighted than the column player, shows that being myopic does not mean that more unilateral defection would be played, as indicated by the percentages of DC (36%) and CD (4%). However, one may suggest that in this case a myopic agent in effect finds a worse policy, as it accumulates less payoff than the farsighted one. This is in line with the fact that in general, farsighted agents should accumulate more reward in the long-term than myopic ones. Finally, when an NN is compared with an LT, CC gets 54%. The reason for this could be in the difference in policy learning by the agents given that the LT learns faster than the NN agent and accumulates more reward (CD>DC). Fig. 3a illustrates the accumulated net payoffs over time for two homogeneous cases, one with a farsighted and one with a myopic Q agent, both with LTs evaluated in self-play. It is clearly evident that in the case of the farsighted agent the accumulated payoff of the system is higher than the case of the myopic agent, which together with the results as presented in Fig. 2, support that CC is higher for farsighted agents.

3.2 Spiking Neural Network Agents

For the system configuration described in Section 2.2 a single game of the IPD consists of 200 rounds during which the two networks seek to maximise their individual accumulated payoff by cooperating or defecting at every round of the

game. This simulation aims to investigate the capability of the spiking NNs to cooperate in the IPD. Two simulations were performed with the synapses of the two networks having different eligibility trace time constants which reflect how the networks integrate time related events and thus associated with the "memory" each network has. The values for both networks were set to 20ms and 2ms for the two simulations respectively. Therefore, during the first simulation the networks have a "good memory" whereas in the second one a "weak memory". The results of both simulations are shown in Fig. 3b. The difference in the

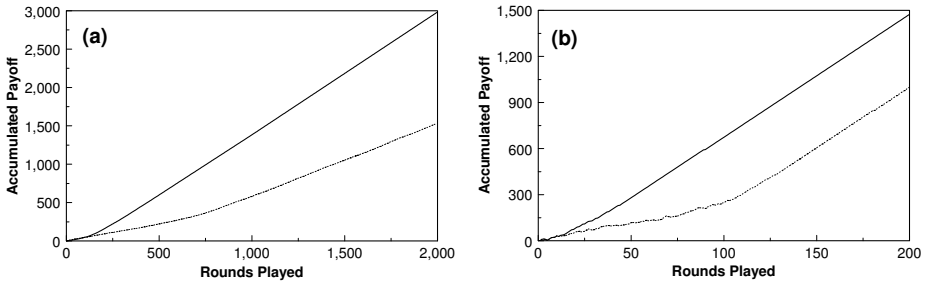

Fig. 3. Accumulated Payoff with: (a) non-spiking Q-agents (LTs) with the two players having a discount factor of 0.9 (*solid line*) and 0.1 (*dotted line*); (b) with spiking NNs with the two players having eligibility trace time constants 20ms (*solid line*) and 2ms (*dotted line*).

system's performance is evident. When the system was configured with 20ms eligibility trace time constants, the accumulated payoff is much higher than the one with 2ms; this results from the difference in the cooperative outcome. With the eligibility trace time constants set at 20ms the two networks learned quickly to cooperate in order to maximise their long-term reward and achieved the CC outcome 182 out of the 200 times. On the contrary, when the system was configured with "weak memory", learning took effect much later during the game (after the 100th round) and thus the system resulted in exhibiting much less cooperation (120 out of 200). However, the system with both configurations eventually managed to learn to cooperate. It is noted that the CC outcome not only persisted during the final rounds of the simulations, but it also did not change after a point (much earlier in the first case) due to the system's dynamics that were evolved by that point in time in such a way to produce CC consistently. Results show that agents' memory influences the cooperative outcome of the game in the sense that it delays it to a great extent. However, a weak memory does not destroy learning as the networks eventually learned to cooperate.

4 Conclusions

Our results indicate that the system accumulates higher cooperative reward when both agents have: (i) higher discount factor, in the case of non-spiking

agents, or (ii) "stronger" memory, as in the case of the spiking agents with longer eligibility trace time constant. One may suggest that the effect of the discount factor in the non-spiking agents is equivalent to the effect of the "synaptic memory" of the spiking agents resulting from the use of eligibility traces, despite the fact that the former refers to future predictions, whereas the latter to past events. Based on this assumption, one could explain the more frequent emerging of the cooperative outcome with high values of eligibility trace time constant and discount factor, in the spiking and non-spiking systems respectively. However, in order to make a direct comparison between the spiking and non-spiking systems we could either use eligibility traces in the Q-learning algorithms [28], or employ TD learning in our spiking NNs as in [36].

According to the results, apart from being desirable for the non-spiking agents to be farsighted in order to achieve the pareto-optimal outcome of the game (as mentioned above), they should also use LTs where possible (since LTs enable the system to converge faster and more accurately) and have learning algorithms of the same or of similar "strength" (such as both Q, or one Q and one SARSA). In the case of spiking agents, it has to be noted that our extension of the reinforcement of stochastic synaptic transmission of Seung [23], by enhancing competition between output neurons (see Section 2.2) through concurrently applying a positive global reinforcement to one output and a negative global reinforcement to the other output is both novel and necessary. More specifically it is essential, so as to avoid a positive feedback effect which would have increased the synaptic strength without bounds, leading to saturation of the synaptic connection and thus preventing further learning from taking place (like the limitation of classical Hebbian learning). Therefore one could conclude that in cases where more than one neuron competes for reinforcement in a spiking NN, the global evaluation signal of Seung's reinforcement of stochastic synaptic transmission [23], should consist of global reward and penalty accordingly, for avoidance of possible synaptic saturation. In addition, it is also desirable for the payoff matrix for the non-spiking agents to mix positive and negative values (as in [37]), which if viewed as another technique of introducing competition into the system (as above), it could explain the enhancement of the cooperative outcome. As mentioned in Section 2.2, this mixture is necessary for the spiking agents.

In general, as it can be seen from the results, the behaviour of spiking and non-spiking agents is in effect similar (Fig. 3). We could therefore argue, that spiking agents could equally well be used in multiagent interactions as non-spiking agents. Certainly, a spiking agent system is more computationally expensive and should only be used when the task in question demands more biologically realistic models. For example, we have used a spiking multiagent system in modelling the high level behaviour of self-control [38].

Acknowledgments. We gratefully acknowledge the support of the University of Cyprus for a Small Size Internal Research Programme grant and the Cyprus Research Promotion Foundation as well as the European Union Structural Funds for grant PENEK/ENISX/0308/82.

References

1. Littman, M.L. In Cohen, W., Hirsh, H., eds.: ICML, M Kaufmann (1994) 157–163
2. Hu, J., Wellman, M.P. JMLR **4** (2003) 1039–1069
3. Claus, C., Boutilier, C. In: AAAI/IAAI, AAAI Press (1998) 746–752
4. Littman, M. In Brodley, C., Danyluk, A., eds.: ICML, M Kaufmann (2001) 322–328
5. Banerjee, B., Peng, J. In Elomaa, T., Mannila, H., Toivonen, H., eds.: ECML. LNCS 2430, Springer (2002) 1–9
6. Greenwald, A.R., Hall, K. In Fawcett, T., Mishra, N., eds.: ICML, AAAI Press (2003) 242–249
7. Kapetanakis, S., Kudenko, D. In Kudenko, D., Kazakov, D., Alonso, E., eds.: Adaptive Agents and Multi-Agent Systems. LNCS 3394, Springer (2005) 119–131
8. Bowling, M.H. In Saul, L.K., Weiss, Y., Bottou, L., eds.: NIPS. (2004) 209–216
9. Conitzer, V., Sandholm, T. Mach. Learn. **67**(1-2) (2007) 23–43
10. Busoniu, L., Babuska, R., De Schutter, B. IEEE Trans SMC C **38** (2008) 156–172
11. Bowling, M., Veloso, M. In Nebel, B., ed.: IJCAI, M Kaufmann (2001) 1021–1026
12. Powers, R., Shoham, Y., Vu, T. Mach. Learn. **67**(1-2) (2007) 45–76
13. Shoham, Y., Powers, R., Grenager, T. Artif. Intell. **171**(7) (2007) 365–377
14. Erev, I., Roth, A.E. Artif. Intell. **171**(7) (2007) 423–428
15. Gordon, G.J. Artif. Intell. **171**(7) (2007) 392–401
16. Fudenberg, D., Levine, D.K. Artif. Intell. **171**(7) (2007) 378–381
17. Tuyls, K., Parsons, S. Artif. Intell. **171**(7) (2007) 406–416
18. Stone, P. Artif. Intell. **171**(7) (2007) 402–405
19. Xie, X., Seung, H. Physical Review E **69**(4) (2004) 41909
20. Abbott, L., Nelson, S. Nature Neuroscience **3** (2000) 1178–1183
21. Florian, R. Neural Comp **19**(6) (2007) 1468–1502
22. Legenstein, R., Pecevski, D., Maass, W. PLoS Comput Biol **4**(10) (2008) e1000180
23. Seung, H. Neuron **40**(6) (2003) 1063–1073
24. Rappoport, A., Chammah, A.: Prisoner's dilemma: a study in conflict and cooperation. Ann Arbor, MI: Univ of Michigan Press (1965)
25. Nash, J. PNAS **36**(1) (1950) 48–49
26. Fudenberg, D., Tirole, J.: Game Theory. Cambridge, MA:MIT Press (1991)
27. Zinkevich, M., Greenwald, A., Littman, M.L. Artif. Intell. **171**(7) (2007) 440–447
28. Watkins, C.: PhD thesis. Univ of Cambridge (1989)
29. Sandholm, T., Crites, R. Biosystems **37**(1-2) (1996) 147–166
30. Sutton, R. Mach. Learn. **3**(1) (1988) 9–44
31. Rummery, G., Niranjan, M.: Technical Report CUED/F-INFENG/TR 166. Cambridge Univ Eng Dept (1994)
32. Lin, L.J. Mach. Learn. **8**(3-4) (1992) 293–321
33. Klopf, A.: The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence. Hemisphere Pub (1982)
34. Baird, L., Moore, A. In Kearns, M., Solla, S., Cohn, D., eds.: NIPS, MIT Press (1998) 968–974
35. Wiering, M. In Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D., eds.: ECML. LNCS 3201, Springer (2004) 477–488
36. Potjans, W., Morrison, A., Diesmann, M. Neural Comp **21**(2) (2009) 301–339
37. Kraines, D., Kraines, V. In Müller, J.P., Wooldridge, M., Jennings, N.R., eds.: ATAL. LNCS 1193, Springer (1996) 219–231
38. Christodoulou, C., Banfield, G., Cleanthous, A. J. Physiol. (Paris) (2009) (in press)