Payoff size variation problem in simple reinforcement learning algorithms[1]

Michal Kvasničk[2]

Abstract. This paper shows that the speed of the reinforcement learning depends on the size of the payoffs, at least when all payoffs are positive. When the speed of learning is too fast, the agents tend to learn to play the actions which they randomly chosen in the first rounds of the learning process. The compositions of the agents' strategies then on the aggregate level resembles the initial individual agent's mixed strategy. This may create artificial effects in the simulations where the size of payoffs depend on the model treatments because the speed of learning cannot be tuned in.

Keywords: reinforcement learning, agent-based simulation, economic experiments, voluntary provision of public goods

JEL classification: C92, D83, H41
AMS classification: 68U, 68W, 91B, 91E

1 Introduction

Standard economic models based on optimizing omniscient agents and instantly attained equilibria are not able to explain many observed phenomena. For instance, they often fail to explain the outcomes of laboratory experiments with human subjects. That is why there are attempts to create alternative formal explanatory models. One approach to this goal is Agent-based computational economics (for a general overview see [14]). The ACE models populated with boundedly rational agents are able to predict not only the eventual equilibrium but also the adjusting process leading to it. Their structure makes them especially useful for modeling the behavior observed in the experiments, see [6]. Since the agents in these models are only boundedly rational they usually have to learn how to act from the feedback provided by their model environment. Thus the learning process constitutes an important part of the models (for a general overview of the learning algorithms used in the ACE model see [4]).

In this paper, we will explore one overlooked property of one of the most often used learning algorithms: the simple reinforcement learning (we use the adjective "simple" to distinguish the reinforcement learning algorithms used in ACE from the more complex algorithms used in the field of artificial intelligence, see [12].) We will claim that the speed of the simple reinforcement learning depends on the size of the payoffs, at least when all payoffs are positive. When the speed of learning is too fast, the agents tend to learn to play the actions which they randomly chosen in the first rounds of the learning process. The compositions of the agents' strategies then on the aggregate level resembles the initial individual agent's mixed strategy. This is no problem in most simulations because the modeler has tools to tune in the speed of learning. However, this may create strange effects in the simulations where the size of payoffs depends on the model treatments—then the speed of learning varies with the treatment, and hence cannot be fine-tuned.

Several parts of the claim has been previously known. Sutton and Barto claim without any proof that the proper setting of the algorithm depends on the size of the payoffs [12, p. 31] and that the eventual mixed strategy may be biased by its prior [12, p. 35]. Fudenberg and Levine claim (also without any proof or elaboration) that there is a positive probability that the algorithm converges to a state "where an inferior choice is played with probability 1" [7, p. 73]. The closest to this paper came Bell [3] who inquired the influence of the initial prior on the algorithm's outcomes with a constant payoff. As far as we know, the influence of the variable payoff size on the outcomes of the algorithm has never been explored.

The rest of the paper proceeds like this: The second section describes the reinforcement learning algorithm and shows the claim analytically in the simplest possible case of one agent playing against a deterministic automaton.

[1]This paper has been created as a part of project of specific research no. MUNI/A/0797/2012 at the Masaryk University.
[2]Masaryk University, Faculty of Economics, Department of Economics, Lipová 41a, Brno, 602 00, the Czech Republic, michal.kvasnicka@econ.muni.cz

The third section provides a more complex example: a simulated version of the provision of public goods experiment where the treatment is the number of agents. It shows that the agents learn to contribute more when the number of agents in the game is higher. The fourth section compares the reinforcement learning to the replicator dynamics and then discusses various ways to solve the problem. It discusses also one more possibility, namely that the discussed effect is not only a computational artifact but a true property of human learning.

2 Speed of reinforcement learning: an analytical example

The basic idea behind the reinforcement learning is that the rewarded behavior is strengthened and the punished behavior is weakened. More specifically, it is assumed that an agent (human or animal) can choose an action from a known discrete set of actions. She has a mixed strategy, i.e. chooses each action with some probability. After choosing an action, the agent earns a payoff that depends on the chosen action and the state of the environment (which can include actions chosen by other agents). After observing the payoff associated with the chosen action, the agent updates her mixed strategy. The probability that the chosen action would be played in the future is increased in proportion to the payoff (it is decreased if the payoff was negative). The future probabilities of all other actions are changed accordingly so that the probabilities of all action sum to unity. The reinforcement learning is classified as "non-conscious" learning (see [4]) because the algorithm does not explicitly model the agent's cognitive reflection. Specifically, it models neither the agent's belief about the state of her environment, nor her belief about the strategies of other agents. Nonetheless, the reinforcement learning was successfully used in modeling the situations where the agents learned consciously (such as in laboratory experiments with human subjects) and where the situation was strategic (e.g. in games), see [4, p. 939]. Very often it produced better predictions of human subjects' behavior in standard games then game theory, see [6, p. 1003].

There are many variants of the simple reinforcement learning algorithm but the differences between them are minor: alternative variants differ only in the speed of learning (it may be constant or slowing down in time) and in the inertia after a change of the environment, see [4, p. 905]. For our exposition, we use the variant of the algorithm taken from [1]. It works like this: The learning proceeds in discrete rounds. In each round t, agent i chooses an action j with probability p_{ij}^t which is calculated from her propensities R_{ij}^t to play the action j as

$$p_{ij}^t = \frac{e^{\lambda R_{ij}^t}}{\sum\limits_{\forall k} e^{\lambda R_{ik}^t}}. \tag{1}$$

After observing the payoff, the propensity to play each action j is updated as

$$R_{ij}^{t+1} = qR_{ij}^t + I_{ij}^t \pi_i^t, \tag{2}$$

where π_i^t is the payoff agent i gained in round t and $I_{ij}^t = 1$ if the action j was chosen in the round t, and $I_{ij}^t = 0$ otherwise. There are three parameters: the initial propensities R_{ij}^1, the "forgetting" parameter $q \in (0, 1]$, and the "focus" parameter $\lambda \geq 0$. The forgetting parameter q allows the agent to change her mind if her environment (and hence payoffs) changed; it also secures the numerical stability of the algorithm. The initial propensities R_{ij}^1 determine the agent's initial mixed strategy; e.g. $R_{i1}^1 = R_{i2}^1 = \ldots = R_{im}^t$ means that in the first round agent i chooses each of her m actions with probability $1/m$. The focus parameter λ is said to determine "the extent to which the agent focuses on choices with higher values of R_{ij}^t" [1, p. 211]. If $\lambda = 0$, then each action is chosen with the same probability. As λ rises, the higher and higher probability is attached to the action with the highest propensity; it is chosen with probability equal to 1 in the limit.

The three parameters together with the size of the payoffs determine the speed of an agent's learning. Intuitively, the speed of learning means how fast the agent's mixed strategy degenerates to the choice of one pure action. This can be measured as the speed with which the entropy of the agent's mixed strategy decreases in time. Entropy E_i^t of agent i's mixed strategy in round t is defined as

$$E_i^t = -\sum_{k=1}^{m} p_{ik}^t \log_m p_{ik}^t \tag{3}$$

where the base of the logarithm is the number m of agent i's actions. Notice that $E_i^t \in [0, 1]$ reaches its maximum when agent i plays each action with the same probability $1/m$, and its minimum when she plays one action with probability 1 and the other actions with probability 0. It has an intermediate value for other mixed strategies and decreases as the agent focuses. We claim that 1) the speed of learning depends beside others also on the payoff

size, and 2) if the speed of learning is too high, the chosen pure action need not to be the right one, i.e. the action with the highest payoff. From some speed of learning, the ex ante probability that the right action would be chosen decreases in the speed of learning and converges to its probability in the initial mixed strategy.

We will show my claim first in the simplest possible environment. Since even this case is not fully analytically tractable, we will discuss analytically only the speed of learning between the first and second round, and the corresponding ex ante probability that the right action is reinforced. The setting is like this: Let us assume a simulation where one agent plays against a deterministic automaton. The agent has two actions, a_1 and a_2. The payoffs of these actions are deterministic, $s\pi_1 > 0$ and $s\pi_2 > s\pi_1$ respectively where s is a "size" of the payoffs (let us say π_1 is normalized to 1). Let us further assume that the agent's initial propensities to play a_1 and a_2 are $R^1_{11} = R^1_{12} = r$ respectively, i.e. she plays each action with probability $1/2$ in the first round.

We have to explore two probabilities: the unconditional probability Ep^2_{12} that the right action a_2 is chosen in the second round and the probability p^- that the action randomly chosen in the first round is chosen also in the second round. Let us start with Ep^t_{12}. If the action a_1 is chosen in the first round, the agent sets the future propensities $R^2_{11} = qr + s\pi_1$ and $R^2_{12} = qr$. Hence, the conditional probability that she chooses action a_2 in the second round when she has chosen a_1 in the first round is then $p^2_{12}|a_1 = e^{\lambda qr}/(e^{\lambda qr+\lambda s\pi_1} + e^{\lambda qr}) = 1/(1 + e^{\lambda s\pi_1})$. On the other hand, if the action a_2 is chosen in the first round, the agent sets $R^2_{11} = qr$ and $R^2_{12} = qr + s\pi_2$. Then the conditional probability that she chooses action a_2 in the second round when she has chosen a_2 in the first round is $p^2_{12}|a_2 = e^{\lambda qr+\lambda s\pi_2}/(e^{\lambda qr} + e^{\lambda qr+\lambda s\pi_2}) = e^{\lambda s\pi_2}/(1 + e^{\lambda s\pi_2})$. Since each action is chosen with probability equal to $1/2$ in the first round, the unconditional probability Ep^2_{12} is

$$Ep^2_{12} = 1/2p^2_{12}|a_1 + 1/2p^2_{12}|a_2 = \left(\frac{e^{\lambda s\pi_2}}{1 + e^{\lambda s\pi_2}} + \frac{1}{1 + e^{\lambda s\pi_1}}\right)/2. \tag{4}$$

The probability that the same action is chosen in both rounds is

$$p^- = 1/2(1 - p^2_{12}|a_1) + 1/2p^2_{12}|a_2 = \left(\frac{e^{\lambda s\pi_1}}{1 + e^{\lambda s\pi_1}} + \frac{e^{\lambda s\pi_2}}{1 + e^{\lambda s\pi_2}}\right)/2. \tag{5}$$

The inspection of the equations (4) and (5) shows the following properties: First, from the equation (5) we can see that the probability p^- that the same action is played in the second round as in the first round is rising in the focus parameter λ, in the average size of the payoffs s, and in their product λs. If $\lambda s = 0$, then $p^- = 1/2$. As the product λs rises, the probability that the action chosen randomly in the first round is played again in the second round monotonically rises and converges to unity, i.e. $\lim_{\lambda s\to\infty} p^- = 1$. Notice that in our case, the agent's mixed strategy entropy E^t_i decreases as the probability that any action is chosen rises above $1/2$ (and the probability that the other action is chosen decreases below $1/2$). That means that the speed of learning between the first and the second round increases in λs.

Second, from equation (4) we can see that the unconditional probability Ep^2_{12} that the right action a_2 is chosen in the second round is increasing in π_2 and decreasing in π_1. In other words, the higher the difference in payoffs between the actions, the higher ex ante probability that the agent will choose the right action in the second round.

Third, from the same equation we can also see that the unconditional probability $Ep^2_{12} = p^1_{12} = 1/2$ in two cases: 1) when $\lambda s = 0$, i.e. when there is no learning at all and the agent chooses her actions independently in each round, each action with the same probability; 2) in the limit when λs is high ($\lim_{\lambda s\to\infty} Ep^2_{12} = 1/2$). In this case, the agent chooses in the second round the same action as in the first round (see the first point above), i.e. there is no further learning since the agent is locked in the previously randomly chosen action. The $Ep^2_{12} = 1/2$ because each action is chosen with this probability in the first round. Notice also that the agent's (with probability $1/2$ inefficient) action is locked, and the agent cannot change her mind: she chooses the previously chosen action in each round, i.e. the other action is never tried and the propensity to play it decreases to zero while the action randomly chosen in the first round is reinforced forever.

Fourth, $dEp^2_{12}/d(\lambda s) > 0$ for $\lambda s = 0$. This together with $Ep^2_{12} = p^1_{12} = 1/2$ for $\lambda s = 0$ and $\lim_{\lambda s\to\infty} Ep^2_{12} = 1/2$ implies that there is a level l such that $dEp^2_{12}/d(\lambda s) < 0$ for any $\lambda s > l$, i.e. the probability that the dominant action a_2 is chosen in the second round decreases in λs, i.e. it decreases with the speed of learning.

We can summarize it this way: If λs is too small, the agents learn too slowly. If the product is too high, the agents learns so fast that he may learn to play a dominated action. Obviously, this is no problem in most simulations—the modeler simply has to set the focus parameter λ properly: the higher s, the lower λ. This way the learning speed may be calibrated for instance to fit the convergence speed observed in an experiment. (However, it might then be difficult to interpret the parameters q and λ as behavioral.) However, there is one overlooked instance

where this is indeed a problem: in simulations in which the size of payoffs depend on the treatment, i.e. it changes within the simulation. Then the speed of the learning can change within the simulation and unexpected things can happen. We will provide an example of such a simulation in the following section.

3 Case study: voluntary provision of public goods

In this section, we will describe an agent-based computational model of the simplest version of the voluntary provision of public goods experiment. The experiment consists of T discrete rounds. There are $N > 2$ players, the same in all rounds. In each round, each player is given some endowment w. She can contribute part of the endowment to a public good and save the rest for herself. Agent i's payoff is $\pi_i^t = (w - c_i^t) + M \sum_{\forall k} c_k^t$ in the round t, where c_i^t is agent i's contribution to the public good in round t and $M \in (0, 1)$ is the payoff of the public good. Notice that the only dominant (and hence equilibrium) action of every agent (both in each round and in the whole finitely repeated game) is $c_i^t = 0$. However, in the typical situation $M > 1/N$, the socially optimal action is $c_i^t = w$.

The stylized facts on results of the experiment can be found in [2, 8, 9]. They show that most people follow neither their dominant, nor their socially optimal strategy, but contribute somewhere in-between. The typical average contribution is about one half in the first round and it decreases in the time, however not to the zero. Since the game theory is not predictive here and since we know that most agent change their actions in time, it seems plausible that they learn how to play the game. Hence the experiment is a natural candidate for an agent-based computational simulation trying to explain the agents' behavior.

We will use the version of the experiment taken from [13] for the simulation. Here, the return on the public good $M = 0.5$, the endowment $w = 40$ and the set of actions is limited to three actions: $a_1 : c_i^t = 40$, $a_2 : c_i^t = 20$, and $a_3 : c_i^t = 0$. The only treatment in the simulation (not in the experiment [13]) is the number of the agents, $N = 4, 7, 10, 20, 40$, and 100. Notice, that the structure of the game implies that an increase in the number of agents rises the size of payoffs for $c_i^t > 0$. The parameters of the reinforcement learning algorithm were casually calibrated in such a way that they follow the stylized fact with the typical number of agents, $N = 7$. The forgetting parameter $q = 0.9$, the focus parameter $\lambda = 0.005$, and initial propensities $R_{ij}^1 = 0$ for each agent i's each strategy j. The model was simulated for 21 rounds, which is more than enough for a comparison with data from any experiment. Each simulation was repeated one thousand times. The model was simulated and the resulting data were analyzed in R [11].

The results of the simulations are summarized in Table 1. In both its panels, the columns denote the number of agents in the game and the rows denote rounds. The right panel shows that the average entropy of agents' mixed strategies (averaged over the agents and the simulations). It can be clearly seen that the agents learn much faster when there are more agents, and hence the size of payoffs is higher. For instance, when $N = 4$, there is some entropy in the agents' mixed strategies even after twenty rounds, i.e. the agents are still learning even after twenty rounds. On the other hand, when $N = 20$, the learning process has stopped after 13 round, and when $N = 100$, the learning has stopped after only two rounds.

The left panel of Table 1 shows the average contributions to the public good in percents of the endowment w (averaged over the agents and the simulations). This can be seen a measure of how fast (and if at all) the agents learn to play their dominant free riding strategy a_3. The more agents learned to play the dominant strategy, the less they contribute, and hence the lower is the average contribution. In the first round, the expected value of the average percentage contribution is $1/2$, which is given by the initial mixed strategy $(1/3, 1/3, 1/3)$. The average contribution in later rounds is rising in N for $N \geq 7$. This is clearly an artifact of the learning algorithm since there is nothing in the agents' preferences or in the structure of the game that could cause it. The reason is that the learning is faster when the number N of agents in the game is higher. The faster the learning, the higher probability that the agents learn a wrong action, and hence contribute more. The extreme case happens when $N = 100$. Then the agents learns their actions almost instantly—they learn to play the action they randomly chose in the first round. The reason is simple. The expected total contribution of 100 agents is $100/3 \times 0 + 100/3 \times 20 + 100/3 \times 40 = 2000$; an agent's payoff is then about 1000 (we can neglect the agent's individual private savings here). The agent then sets the propensity of the randomly chosen action to $R_{ij}^2 = qR_{ij}^1 + \lambda\pi_i^1 \doteq 0.005 \times 1000 = 5$. The probability that the same action would be chosen in the second round is then about $e^5/(2 + e^5) \doteq 99$ %. The eventual expected value of the average contribution is then the same as the initial one: $1/2$. It is because each agent learned to play always the action she randomly chose in the first round, and each agent chose each action initially with probability equal to $1/3$. Now, one third agents play the pure action a_1, one third the pure action a_2, and one third the pure action a_3.

When the number of agents in the game is lower, the learning is slower, and the entropy of the agents' mixed strategies drops to zero later. The longer learning allows the agents to learn better action. However, not all agents

(a) average percentage contribution to the public good

$t \setminus N$	4	7	10	20	40	100
1	0.509	0.499	0.495	0.499	0.501	0.502
2	0.499	0.493	0.490	0.492	0.498	0.501
3	0.483	0.492	0.492	0.488	0.495	0.502
4	0.490	0.479	0.482	0.488	0.495	0.502
5	0.490	0.478	0.473	0.484	0.494	0.502
6	0.461	0.474	0.475	0.481	0.494	0.502
7	0.456	0.458	0.466	0.479	0.494	0.502
8	0.459	0.453	0.458	0.478	0.494	0.502
9	0.460	0.450	0.450	0.476	0.494	0.502
10	0.442	0.451	0.452	0.479	0.494	0.502
11	0.438	0.437	0.443	0.477	0.494	0.502
12	0.435	0.431	0.438	0.475	0.494	0.502
13	0.428	0.428	0.441	0.475	0.494	0.502
14	0.422	0.429	0.438	0.476	0.494	0.502
15	0.416	0.425	0.434	0.476	0.494	0.502
16	0.403	0.415	0.430	0.477	0.494	0.502
17	0.427	0.408	0.433	0.476	0.494	0.502
18	0.397	0.406	0.427	0.476	0.494	0.502
19	0.400	0.403	0.426	0.477	0.494	0.502
20	0.397	0.396	0.421	0.476	0.494	0.502
21	0.404	0.392	0.424	0.476	0.494	0.502

(b) average entropy of agents' mixed strategies

$t \setminus N$	4	7	10	20	40	100
1	1.000	1.000	1.000	1.000	1.000	1.000
2	0.964	0.924	0.865	0.610	0.199	0.002
3	0.926	0.838	0.728	0.344	0.045	0.000
4	0.887	0.752	0.598	0.185	0.009	0.000
5	0.842	0.665	0.476	0.096	0.002	0.000
6	0.802	0.582	0.383	0.050	0.001	0.000
7	0.757	0.511	0.306	0.028	0.000	0.000
8	0.712	0.448	0.249	0.016	0.000	0.000
9	0.671	0.388	0.200	0.008	0.000	0.000
10	0.633	0.337	0.160	0.005	0.000	0.000
11	0.597	0.296	0.127	0.003	0.000	0.000
12	0.563	0.268	0.102	0.001	0.000	0.000
13	0.533	0.240	0.083	0.001	0.000	0.000
14	0.508	0.218	0.069	0.000	0.000	0.000
15	0.485	0.192	0.057	0.000	0.000	0.000
16	0.461	0.174	0.049	0.000	0.000	0.000
17	0.435	0.157	0.041	0.000	0.000	0.000
18	0.419	0.145	0.035	0.000	0.000	0.000
19	0.403	0.135	0.030	0.000	0.000	0.000
20	0.388	0.125	0.027	0.000	0.000	0.000
21	0.374	0.116	0.025	0.000	0.000	0.000

Table 1 The left table shows the average contribution to the public goods in percents for a given number of agents (columns) in a given round (rows). The right table shows the average entropy of the agents' mixed strategies for a given number of agents (columns) in a given round (rows). Each value is averaged over one thousand simulations.

need to learn their dominant action. For instance, Table 1 shows that with $N = 20$, the agents' actions are locked after the 13th round but the agents still contribute in average 47.6 % of their endowment (instead of zero, as predicted by their dominant strategy).

4 Discussion

We have shown that the speed of reinforcement learning depends beside others on the size of payoffs. The higher payoffs, the faster learning. If the learning is too fast, the agents can learn to play the action they have randomly chosen in the first round. In such a case, the final distribution of the agents' pure strategies resembles the initial one agent's mixed strategy. When the modeler cannot control for this effect, unexpected results can happen. In the contribution to the public good simulation in the previous section, the result was that the average contribution rose with the number of agents in the game.

It might be interesting to compare the result with the replicator dynamics. Miller [10] showed for a similar version of the experiment that the more agents in the game, the slower the agents learn their dominant free riding actions through the replicator dynamics. He also showed that it is because the more agents imply the higher payoffs. However, there are two key differences between the replicator dynamics and reinforcement learning. First, in the replicator dynamics, the slow learning to play the right action is the result of the slow learning while in the reinforcement learning it is the result of too fast learning. Second, in the replicator dynamics, the agents eventually learn to play their dominant actions while in the reinforcement learning they can get stuck with suboptimal actions.

Although the problem discussed in this paper has been overlooked so far, there exists enhancements of the algorithm that can overcome it. The lock-in effect can possibly be alleviated by introduction of experimentation (I^t_{ij} in equation (2) is redefined so that $I^t_{ij} = 1$ when action j has been chosen, or $I^t_{ij} = \delta$ otherwise, where $\delta \in [0, 1]$) or by introduction of fictive payoffs to the non-chosen actions as in EWA (the term $I^t_{ij}\pi^t_i$ in equation (2) is redefined so that I^t_{ij} is as in experimentation and π^t_i is replaced by the payoff that the agent would obtain if she has chosen action j and the other agents have not changed their actions). However, neither of these techniques can solve the

problem completely if the payoffs are high enough since δ should be typically small.

The influence of the payoffs' size can be completely eliminated by substituting the gross payoffs determined by the game with some form of net payoffs, i.e. a transformation of the gross payoffs. There are many ways how to calculate the net payoffs. For instance, one can subtract an aspiration level from the gross payoffs, or calculate the fictive payoffs $\tilde{\pi}_{ij}^t$ and then subtract the $\max(0, \min_{\forall j}(\tilde{\pi}_{ij}^t)$ from the gross payoffs. Both these techniques can eliminate the variation in the speed of leaning if used properly. However, some modelers can object to it because it may be incompatible with learning from the feedback. For instance, in his seminal paper, Cross [5, p. 247] claims: "It should be stressed that the concept of opportunity cost has no place in this analysis either. The theory of opportunity cost is derived from the maximization hypothesis, and its introduction would be inconsistent with the action-taking orientation of this paper."

Finally, it should be mentioned that there is one more possibility, namely that the variation of the speed of learning with the payoff size and the resulting effects are not just an computational artifact but a property of human (and animal) learning. Indeed, one of the stylized facts of the contribution to public goods experiments is that the average contribution does rise with the number of human subjects in the game, see [2, 8, 9]. When the agent-based model of learning uses net payoffs (as e.g. in [2]), the fact must be explained by other-caring preferences. The variation in the speed of reinforcement learning then offers an alternative (and perhaps easier) explanation. Clearly, more research on how people truly learn is needed.

References

[1] Arifovic, J.—Ledyard, J.: Scaling Up Learning Models in Public Good Games, *Journal of Public Economic Theory* **6** (2004), 203–238.

[2] Arifovic, J.—Ledyard, J.: Individual evolutionary learning, other-regarding preferences, and the voluntary contributions mechanism, *Journal of Public Economic Theory* **96** (2012), 808–823.

[3] Bell, A. M.: Reinforcement Learning Rules in a Repeated Game, *Computational Economics* **18** (2001), 89–111.

[4] Brenner, T.: Agent Learning Representation: Advice on Modeling Economic Learning. In: *Handbook of Computational Economics*, vol. 2, (L. Tesfatsion and K. L. Judd, eds.), Elsevier, 2006, 831–880.

[5] Cross, J. G.: A stochastic learning model of economic behavior, *Quarterly Journal of Economics* **87** (1973), 239–266.

[6] Duffy, J.: Agent-Based Models and Human Subject Experiments. In: *Handbook of Computational Economics*, vol. 2, (L. Tesfatsion and K. L. Judd, eds.), Elsevier, 2006, 831–880.

[7] Fudenberg, D.—Levine, D. K.: *The Theory of Learning in Games*, MIT Press, 1998.

[8] Holt, Ch. A.: *Markets, Games, & Strategic Behavior*, Pearson Education, 2007.

[9] Ledyard, J. O.: Public Goods: A Survey of Experimental Research. In: *Handbook of Experimental Economics*, (J. H. Kagel and A. Roth, eds.), Princeton University Press, 1995.

[10] Miller, J. H.: Can evolutionary dynamics explain free riding in experiments?, *Economic Letters* **36** (1991), 9–15.

[11] R Development Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Wien, 2011, http://www.R-project.org/.

[12] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning: An Introduction*, 2nd ed. in progress, available at http://webdocs.cs.ualberta.ca/˜sutton/book/the-book.html, 2012.

[13] Šeneklová J.—Špalek, J.: Jsou ekonomové jiní? Ekonomický model versus realita, *Politická ekonomie* 1 (2009), 21–44.

[14] Tesfatsion, L.: Agent-Based Computational Economics: A Constructive Approach to Economic Theory. In: *Handbook of Computational Economics*, vol. 2, (L. Tesfatsion and K. L. Judd, eds.), Elsevier, 2006, 831–880.