Artificial agents learning human fairness

Steven de Jong
MICC, Maastricht University,
The Netherlands
steven.dejong@micc.unimaas.nl

Karl Tuyls
Faculty of Industrial Design,
Eindhoven Technical
University, The Netherlands
ktuyls@gmail.com

Katja Verbeeck
Katholieke Hogeschool St.
Lieven, Gent, Belgium
katja.verbeeck@kahosl.be

ABSTRACT

Recent advances in technology allow multi-agent systems to be deployed in cooperation with or as a service for humans. Typically, those systems are designed assuming individually rational agents, according to the principles of classical game theory. However, research in the field of behavioral economics has shown that humans are not purely self-interested: they strongly care about fairness. Therefore, multi-agent systems that fail to take fairness into account, may not be sufficiently aligned with human expectations and may not reach intended goals. In this paper, we present a computational model for achieving fairness in adaptive multi-agent systems. The model uses a combination of Continuous Action Learning Automata and the Homo Egualis utility function. The novel contribution of our work is that this function is used in an explicit, computational manner. We show that results obtained by agents using this model are compatible with experimental and analytical results on human fairness, obtained in the field of behavioral economics.

Categories and Subject Descriptors

I.2.6 [**Learning**]; I.2.11 [**Distributed Artificial Intelligence**]; J.4 [**Social and Behavioral Sciences**]

General Terms

Algorithms, Design, Human Factors

Keywords

Fairness, Homo Egualis, Reinforcement Learning

1. INTRODUCTION

Modeling agents for a multi-agent system requires a thorough understanding of the type and form of interactions with the environment and other agents in the system, including any humans. Since many multi-agent systems are designed to interact with humans or to operate on behalf of them, for instance in bargaining [12, 36], resource distribution [10] and aircraft deicing [24], agents' behavior should often be aligned with human expectations. Otherwise, agents may fail to reach their goals.

Usually, multi-agent systems are designed according to the principles of a standard game-theoretical model, i.e., assuming individual rationality. However, recently, this strong assumption has been relaxed in various ways, for instance by including well-known concepts such as bounded rationality [42] and social welfare [7, 8]. Research in the field of behavioral economics shows us that humans are not purely rational and self-interested; their decisions are often based on considerations about others [4, 17, 18]. Therefore, multi-agent systems using only standard game-theoretical principles risk being insufficiently aligned with human expectations and may not obtain satisfactory payoffs. Prime examples known from (evolutionary) game theory include games such as the Ultimatum Game [17], in which purely rational players usually obtain a very low payoff, and games such as the Public Goods Game [17, 41] or the Traveler's Dilemma [2], in which humans can actually obtain a higher payoff by failing to find the rational solution, i.e., the Nash equilibrium. More generally speaking, fairness may be important in any problem domain in which the allocation of limited resources plays an important role [7], as in the examples mentioned above.

Thus, designers of a variety of multi-agent systems should take the human conception of fairness into account. If the motivations behind human fairness are sufficiently understood and modeled, the same motivations can be transferred to multi-agent systems. More precisely, *descriptive* models of human fairness may be used as a basis for *prescriptive* or *computational* models, used to control agents in multi-agent systems in a way that guarantees alignment with human expectations. This interesting track of research ties in with the descriptive agenda formulated by Shoham [40] and the objectives of evolutionary game theory [18, 44].

In this paper, we show that it is possible for multi-agent systems to explicitly represent and utilize human fairness. We use a descriptive model of human fairness called Homo Egualis [17] and introduce this model into an adaptive multi-agent system driven by Continuous Action Learning Automata. In contrast to earlier work [46], in which agents were inspired by the Homo Egualis model to obtain a fair distribution of limited resources, we use the model in a direct, computational manner, to obtain the best possible alignment with human behavior. We study the concrete behavior of our computational model in two game settings (more precisely, the Ultimatum and Nash Bargaining Game, extended for more players), both of which represent common bargaining situations. We then determine whether we can find and maintain solutions as calculated by behavioral economists – i.e., fair solutions that tie in with human behavior.

In the remainder of this paper, we first discuss work in the area of descriptive models of human fairness. Then, we look at computational or prescriptive modeling of fairness, first outlining existing work in this area, then discussing the games we are looking at in more detail, and finally presenting our own methodology. The paper continues with a set of experiments, after which we discuss results elaborately and conclude.

Cite as: Artificial agents learning human fairness, Steven de Jong, Karl Tuyls and Katja Verbeeck, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, MüllerandParsons(eds.),May,12-16.,2008,Estoril,Portugal,pp.863-870.

2. MODELING HUMAN FAIRNESS

Already in the 1950's people started investigating fairness, for instance in the Nash Bargaining Game [27]. Recently, research in behavioral economics and evolutionary game theory has examined human behavior in various games, such as the Ultimatum Game and the Public Goods Game (e.g., [3, 17]). In comparison to the fair outcomes reached by human players, standard game-theoretical models predict a very selfish (and suboptimal) outcome in these games. The current state of the art describes and models three main motivations for human fairness.

Inequity aversion. In [17], this is defined as follows: *"Inequity aversion means that people resist inequitable outcomes; i.e., they are willing to give up some material payoff to move in the direction of more equitable outcomes"*. To model inequity aversion, an extension of the classical game theoretic actor is introduced, named Homo Egualis [17, 18]. Homo Egualis agents are driven by the following utility function:

$$u_i = x_i - \frac{\alpha_i}{n-1} \sum_{x_j > x_i} (x_j - x_i) - \frac{\beta_i}{n-1} \sum_{x_i > x_j} (x_i - x_j) \quad (1)$$

Here, u_i is the utility of agent $i \in \{1, 2, \ldots, n\}$. This utility is calculated based on agent i's own payoff, x_i, and two terms related to considerations on how this payoff compares to the payoffs x_j of other agents j: every agent i experiences a negative influence on its utility for other agents j that have a higher payoff as well as other agents that have a lower payoff. Thus, given its own payoff x_i, agent i obtains a maximum utility u_i if $\forall j : x_j = x_i$.

Research with human subjects provides strong evidence that humans care more about inequity when doing worse than when doing better in society [17]. Thus, in general, $\alpha_i > \beta_i$ is chosen. Moreover, the β_i-parameter must be in the interval $[0, 1]$: for $\beta_i < 0$, agents would be striving for inequity, and for $\beta_i > 1$, they would be willing to "burn" some of their payoff in order to reduce inequity, since simply reducing their payoff (without giving it to someone else) already increases their utility value.

The Homo Egualis utility function has been shown to adequately describe human behavior in various games, including the Ultimatum Game [17] and the Public Goods Game [9]. However, it should be noted that there are also experiments in which human behavior is not adequately captured by a utility model that is exclusively based on inequity aversion and material interest [6]. Subjects may also be motivated by additional information they may have about each other, and by reciprocity: they become less cooperative in the presence of defectors and sometimes punish unfair behavior. This leads to two other models, viz. priority awareness and reciprocal fairness, which will be outlined below.

Priority awareness. In [11], the relation between priorities and fairness is studied. Experiments with human subjects show that priorities matter strongly. For instance, priority mail is more expensive than regular mail and should therefore be delivered sooner. To examine the human response in such situations, an additional parameter is introduced in the two-player Ultimatum Game, denoting the fact that one of the players is substantially more wealthy than the other one – i.e., one player has a higher priority in receiving the money at stake. It turns out that humans tend to give less money to more wealthy opponents and accept less money from poor opponents, and the other way around.This behavior is modeled in a descriptive model called *priority awareness*.

Reciprocal fairness. The most important limitation of the inequity-averse and priority-aware models is that they do not explicitly explain how fair behavior evolves with repeated interactions between agents [17]. For instance, a group of people repeatedly playing the same game may start by playing in an individually rational manner, but for some reason may end up playing in a fair, cooperative manner. Reciprocal fairness models aim at providing an answer to the questions why and how this happens. The main idea is that humans cooperate because of direct and indirect reciprocity – here, direct means that a person is nice to someone else because he expects something in return from this other person, and indirect means that an agent is nice to someone else because he expects to obtain something from a third person. It turns out that the opposite, i.e., punishing someone who is nasty, has an even greater effect on cooperation [41]. However, being nasty may be costly, and thus, it would be individually rational to punish when we are sure to encounter the object of punishment again. Once again, humans do not select the individually rational solution: even in one-shot interactions, they consistently apply punishment if this is allowed. Since this is clearly not of direct benefit to the punisher, this phenomenon is referred to as altruistic punishment (see, e.g., [15, 16, 47]). Interestingly, the question thus seems to shift from 'why do people cooperate?' to 'why do people perform costly punishment?'. Various explanations have been analyzed from the perspective of evolutionary game theory [18]. For instance, many researchers argue that altruistic punishment only pays off when the reputation of the players somehow becomes known to everyone [14, 25]. There are also alternative explanations such as volunteering [20, 21], fair intentions [13] or the topology of the network of interaction [38].

Although reciprocal fairness and priority awareness are interesting descriptive models, our current work focuses on constructing a computational model based on inequity aversion, since this model can already explain many aspects of human behavior in bargaining situations, our main topic of interest [9, 17].

3. COMPUTATIONAL FAIRNESS

In this section, we first discuss related work in computational modeling of fairness. Then, we describe the games under study and analyze their rational and fair solutions. Finally, we outline the methodology for the design of our learning agents.

3.1 Related work

Here we discuss some contributions to prescriptive modeling of human fairness. Many of these contributions were originally intended to be descriptive, but were immediately verified in adaptive agent systems and are thus also computational.

Cooperation in multi-agent games. Various researchers study fairness using multi-agent games and claim that fairness (or, alternatively, altruistic punishment) is achieved using internal agent mechanisms such as reputation. To support this claim, the behavior of agents driven by such systems is analyzed, mostly from the perspective of evolutionary game theory [18]. In many papers, it is shown that reputation can indeed increase cooperation [13, 14, 29, 32]. In addition to studies being performed on internal mechanisms of agents, there are also studies focusing on external factors that may lead to fairness. Most notably, researchers argue that humans do not interact on a random basis, as traditionally assumed by population dynamics; instead, human interactions, like many other natural phenomena, seem to be organized in scale-free or small-world networks [38]. Moreover, humans are able to adjust their social ties: in case they interact with a person they turn out not to like, they may refuse to interact with this person again [37]. Indeed, both ideas increase cooperation in adaptive multi-agent systems.

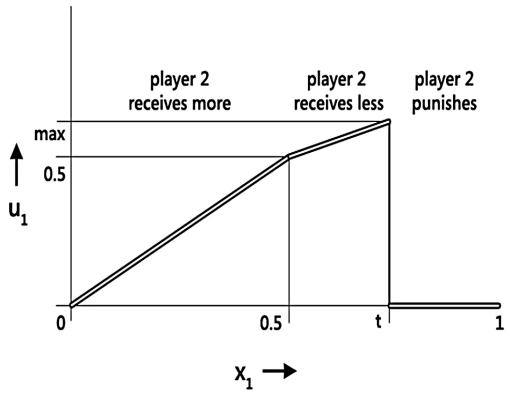

Figure 1: Homo Egualis in the two-agent Ultimatum Game. We illustrate the functional mapping between the payoff agent 1 keeps to himself (x_1) and the utility experienced by this agent (u_1). Agent 2 can reject in case of a negative utility, i.e., if the payoff agent 1 keeps to himself exceeds a threshold $t = \frac{\alpha_2}{1+2\alpha_2}R$. In this case, both agents receive 0.

Mechanism design. Economy-based collaboration mechanisms are very popular in multi-agent research. Mechanism design [33] studies the art of designing the rules of a game such that a specific outcome is achieved. As in the research outlined in this paper, mechanism design assumes that players' individually rational actions may not lead to a desired global outcome. Thus, designers set up a structure in which each player has an incentive to behave as intended. For a comprehensive overview, see [23].

From a computational point of view, some specific issues arise in mechanism design. To start with, unlike with game theoretic assumptions, software agents do not possess unbounded computational power to calculate equilibrium strategies. Theory focuses on centralized mechanisms, but the infrastructure might be unable to compute the outcome because the problem might be intractable. Furthermore, communication between the agents is not necessarily cost- or error-free and the system might be dynamic, with agents entering or leaving the system over time. Current state-of-the-art research in computational mechanism design addresses one or more of these added computational issues [33].

Computational social choice. Another area in which fairness is extensively studied in a computational manner is that of computational social choice (see [7, 8] for a comprehensive overview). This area encompasses many interesting problems at the interface of social choice theory and computer science, for instance fair division in resource allocation [7]. In this case, fairness conditions and mechanisms relate to the well-being of society as a whole. This well-being can be measured in various ways, such as utilitarian social welfare (i.e., maximized average payoff), egalitarian social welfare (i.e., maximized minimal payoff), or Pareto-optimality. We argue that another measure for the well-being of society as a whole should be introduced, i.e., a definition of fairness that is backed up by numerous well-documented experiments with human subjects.

Institutional and social norms. As a final contribution in this section, we mention research in the area of norms and institutions [34, 45, 1]. There are interesting parallels between this research and the research described in the previous sections. More precisely, norms can be used as an environment-driven coordination mechanism, and are an alternative to classical, agent-centered coordination. Especially in open multi-agent systems, i.e., systems in which agents may be heterogeneous or designed by different parties, one cannot assume that all agents pursue the same goal or have the same internal procedures. In fact, the goal of certain agents may be to disrupt the system or to exploit the other agents in the system. In situations such as these, norms may help, since they allow agents to make predictions about others and to direct their own actions toward desirable behavior. Enforcing norms is a complex problem, since norms are usually represented in formalisms that have a declarative nature, but should be translated to an operational implementation [45].

In natural societies, norms and associated punishments emerge over time, either spontaneously or deliberately. Societies use social constraints (norms) to regulate relations among their members, such as customs, traditions, regulations or laws. In addition to being institutional (i.e., enforced by the environment), norms may also be appointed between individual agents; in this case, they are referred to as social norms. Fairness may be reflected in both types of norms. Studying the emergence of social norms in agent systems is recognized as an important research track, since it may improve coordination and functioning of the agent system [39].

3.2 Game analysis

In this work, we aim at computationally obtaining fair solutions in two abstract games, modelling common bargaining situations. More precisely, the games under study are the Ultimatum Game and Nash Bargaining Game, extended for more players. Fair solutions in our case are solutions that, according to research, are generally considered good (or fair) by humans. We will now present a brief analysis of these two games.

Ultimatum Game. The Ultimatum Game [19] is a simple bargaining game, played by two agents. The first agent proposes how to divide a (rather small, e.g., \$10) reward R with the second agent. If the second agent accepts this division, the first gets his demanded payoff and the second gets the rest. If however the second agent rejects, neither gets anything. The game is played only once, and it is assumed that the agents have not previously communicated, i.e., they did not have the opportunity to negotiate with or learn from eachother. The individually rational solution (i.e., the Nash equilibrium) to the Ultimatum Game is for the first agent to leave the smallest positive payoff to the other agent. After all, the other agent can then choose between receiving this payoff by agreeing, or receiving nothing by rejecting. Clearly, a small positive payoff is rationally preferable over no payoff at all.

However, research with human subjects indicates that humans usually do not choose the individually rational solution. Hardly any first agent proposes offers that lead to large differences in payoff between the agents, and hardly any second agent accepts such proposals. In [31], many available experiments with humans are analyzed. It is indicated that the average proposal in the two-agent Ultimatum Game is about 40%, with 16% of the proposals being rejected by the other agent. Our own experiments confirm this [11]. Cross-cultural studies of small cultures have shown that these numbers are not universal. However, independent of culture, the individually rational solution is hardly ever observed [22].

Using the Homo Egualis utility function with two agents, [17] calculates that the optimal payoff for agent 1 depends on two factors, viz. β_1 and α_2. More precisely, in the two-agent game, we have $n = 2$ and $x_2 = R - x_1$. Thus, the Egualis function can be rewritten for both agents as:

$$u_i = x_i - \alpha_i \max(R - 2x_i, 0) - \beta_i \max(2x_i - R, 0) \quad (2)$$

If $\beta_1 > 0.5$, agent 1's utility u_1 will decrease with values of $x_1 > 0.5R$, since $2x_1 - R > 0$. This implies that agent 1 will give

$0.5R$ to agent 2 if $\beta_1 > 0.5$. If $\beta_1 < 0.5$, agent 1's utility is not decreased by increasing his payoff x_1. The agent would like to keep everything to himself. However, he must ensure agent 2 receives a payoff that is not rejected. Agent 2 will reject iff $x_2 - \alpha_2 (R - 2x_2) \leq 0$. Equivalently, we obtain:

$$x_2 \geq \frac{\alpha_2}{1 + 2\alpha_2} \cdot R \rightarrow agent\ 2\ accepts \tag{3}$$

Note that $\lim_{\alpha_2 \rightarrow \infty} = 0.5R$. Thus, agent 2 can expect to obtain at most half of the total reward. For additional clarity, the functional mapping between x_1 and u_1 is illustrated in Figure 1. From this figure, it is clear that the utility function for agent 1 is not continuous: there is a discontinuity immediately after the maximum.

Multi-agent Ultimatum Game. Usually, the Ultimatum Game is played with only two agents. As we are interested in a multi-agent perspective, we also analyze the role of inequity aversion in Ultimatum Games with more than two agents. There are various extensions of the Ultimatum Game to more agents, e.g., introducing proposer competition or responder competition. We propose a different extension. More precisely, we define a game in which $n - 1$ agents one by one take a portion of the reward R. The last agent, n, receives what is left. In this case, we can calculate that the worst-performing agent i will not reject as long as:

$$x_i \geq \frac{\alpha_i}{\alpha_i n + n - 1} \cdot R \tag{4}$$

Moreover, given that $\forall j : i \neq j \rightarrow x_i < x_j$, and assuming that $\forall j : \alpha_j > \beta_i$, we can calculate in a straightforward manner that the utility value u_i of the worst-performing agent i will always be the lowest one. Thus, if agent i does not reject, neither will any other agent. For instance, with three agents and $\alpha_3 = 0.6$, we obtain that the last agent (which can be assumed to perform worst of all in the Ultimatum Game) needs to obtain at least $0.1578R$ in order to accept the deal at hand. As long as the other agents obtain more, they will accept any deal.

Nash Bargaining Game. The Nash Bargaining Game [28] is traditionally played by two agents, but can easily be extended to more agents. In this game, all agents simultaneously determine how much payoff x_i they are going to claim from a common reward R. If $\sum_i x_i > R$, everyone receives 0. Otherwise, everyone receives what they have asked for. Note that payoffs may not sum up to R, i.e., a Pareto-optimal solution is not guaranteed. The game has many Nash equilibria, including one where all agents request the whole R. The common human solution to this game is an even split [30, 35]. Inequity aversion may increase the ability of agents to find such a fair solution. Thus, we give agents an additional action, i.e., even if the payoff distribution was successful, agents may compare their payoff with that of others. Then, if their payoff is too small, they may reject, once again leading to all agents obtaining a payoff of 0. To decide whether their payoff is satisfactory, agents use the Homo Egualis utility function, as in the Ultimatum Game. Thus, we can perform the same analysis as in the Ultimatum Game and obtain that any solution for which every agent obtains at least $\frac{\alpha_i}{\alpha_i n + n - 1} \cdot R$ is not rejected. For example, with $n = 2$ and $\alpha_1 = \alpha_2 = 0.6$, every agent should obtain at least $0.27R$.

3.3 Methodology

In our approach, we aim at simple, learning agents that are sufficiently modelled after humans. To this end, we use a combination of the Homo Egualis utility function (in short, Egualis) and Continuous Action Learning Automata (CALA). As has been mentioned above, Egualis provides the necessary connection between the artificial agents and the human way of thinking in games such as the Ultimatum Game. CALA facilitate the learning process, based on repeated interactions with a specific environment (e.g., a game). We will now briefly discuss the components of our agents.

Learning Automata. Originally, learning automata were developed for learning optimal policies in single-state problems with discrete action spaces [26]. An automaton is assumed to be situated in an environment, in which it executes a certain action x from its non-infinite set of possible actions $\mathbb{A}$. This action x is observed by the environment and leads to a feedback $\beta(x)$ to the automaton. The automaton uses this feedback to update the probability that action x will be chosen again. Thus, a learning automaton is a simple reinforcement learner. With multiple (i.e., n) learning automata, every automaton i receives feedback $\beta_i(\bar{x})$, resulting from the joint action $\bar{x} = (x_1, \ldots, x_n)$, but is not informed about the actions of the other automata. Nonetheless, with certain update schemes, learning automata have been shown to converge to an equilibrium point, e.g., a Nash equilibrium.

Continuous Action Learning Automata. CALA [43] are learning automata developed for problems with continuous action spaces. CALA are essentially function optimizers; for every action a from their continuous, one-dimensional action space $\mathbb{A}$, they receive a feedback $\beta(x)$ – the goal is to optimize this feedback. CALA have a proven convergence to (local) optima, given that the feedback function $\beta(x)$ is sufficiently smooth. The advantage of CALA over other reinforcement techniques, is that it is not necessary to discretize continuous action spaces; actions are simply real numbers. Moreover, they are much less complicated to implement and analyze than various other multi-agent reinforcement techniques for continuous action spaces [44].

Essentially, CALA maintain a Gaussian distribution from which actions are pulled. In contrast to standard learning automata, CALA require feedback on *two* actions, being the action corresponding to the mean μ of the Gaussian distribution, and the action corresponding to a sample x, taken from this distribution. These actions lead to a feedback $\beta(\mu)$ and $\beta(x)$, respectively, and in turn, this feedback is used to update the probability distribution's μ and σ. More precisely, the update formula for CALA can be written as:

$$\mu = \mu + \lambda \frac{\beta(x) - \beta(\mu)}{\Phi(\sigma)} \frac{x - \mu}{\Phi(\sigma)} \tag{5}$$

$$\sigma = \sigma + \lambda \frac{\beta(x) - \beta(\mu)}{\Phi(\sigma)} \left[\left(\frac{x - \mu}{\Phi(\sigma)} \right)^2 - 1 \right] - \lambda K (\sigma - \sigma_L) \tag{6}$$

In this equation, λ represents the learning rate, set to 0.05 in our case; K represents a large constant driving down σ, which in our case is set to 0.1. The variance σ is kept above a threshold σ_L (set to 10^{-5} in our case), to keep calculations tractable even in case of (near-)convergence. This is implemented using the function:

$$\Phi(\sigma) = \max(\sigma, \sigma_L) \tag{7}$$

The intuition behind the update formula is quite straightforward. First, if the signs of $\beta(x) - \beta(\mu)$ and $x - \mu$ match, μ is increased, otherwise it is decreased. This makes sense, given a sufficiently smooth feedback function: for instance, if $x > \mu$ but $\beta(x) < \beta(\mu)$, we can expect that the optimum is located below the current μ. Second, the variance is adapted depending on how far x is from μ. The term $(\frac{x-\mu}{\Phi(\sigma)})^2 - 1$ becomes positive iff x is more than a standard deviation away from μ. In this case, if x is a better action than μ, σ is increased to make the automaton more explorative. Otherwise, σ is decreased to decrease the probability that

the automaton will select x again. If x is not more than a standard deviation away from μ, this behavior is reversed: a 'bad' action x close to μ indicates that the automaton might need to explore more, whereas a 'good' action x close to μ indicates that the optimum might be near. Using this update function, CALA rather quickly converge to a (local) optimum. With multiple (e.g., n) learning automata, every automaton i receives feedback with respect to the joint actions, respectively $\beta_i(\bar{\mu})$ and $\beta_i(\bar{x})$. In this case, there still is convergence to a (local) optimum [43].

Homo Egualis. As mentioned above, we aim at creating agents that can learn 'human' behavior. The Homo Egualis utility function is a satisfactory model of human behavior in various games, including the games we are letting our agents play, i.e., the Ultimatum Game and the Nash Bargaining Game. Therefore, we use this utility function in our learning agents. More precisely, we use a four-step process. First, every agent i is equiped with a Continuous Action Learning Automaton. This automaton selects its actions μ_i and x_i, indicating how much payoff the agent requests. Second, the environment evaluates the joint actions $\bar{\mu}$ and $\bar{x}$ and gives feedback $\beta_i(\bar{\mu})$ and $\beta_i(\bar{x})$, using the rules of the game at hand. In the Ultimatum Game, every agent receives what it has asked for, unless there is not enough reward remaining due to the actions of preceding agents. In this case, the agent receives what is remaining. In the Nash Bargaining Game, everyone receives what they have asked for, unless the sum of their requests exceeds R. In that case, everyone receives 0. Third, the environment's feedback is mapped to utility values $u_i(\bar{\mu})$ and $u_i(\bar{x})$, using the Egualis function, possibly including punishment (i.e., if any agent experiences a negative utility, all utilities are set to 0). Finally, the utility values are reported to the learning automata, which subsequently update their strategies. Note that the n-player Ultimatum Game requires $n-1$ automata, whereas the n-player Nash Bargaining Game requires n automata. In the Ultimatum Game, the last agent's behavior is static: he simply rejects if his utility drops below 0. In the Nash Bargaining Game, all agents are the same.

We use the same parameters for the Homo Egualis function (i.e., α_i and β_i) for all agents participating. This makes the analysis and verification of outcomes easier, especially with many agents. Results obtained by giving each agent i private α_i- and β_i-values will be highly similar to our results, but calculating an expected or optimal solution to compare these results with, is more difficult and requires various constraints on the parameters.

Extensions to the learning rule. CALA have a proven convergence to a local optimum in the case of smooth and continuous feedback functions [43]. However, as is clearly visible in Figure 1, the feedback function we use (i.e., Egualis) displays a discontinuity: maximum feedback is obtained at a certain value, after which the feedback immediately drops to 0 (if punishment is possible). This leads to two problems, both of which need to be addressed without affecting the convergence of CALA.

The first problem arises when the automaton is near the optimum, and either its x-action or its μ-action is slightly too high. As can be seen from Figure 1, one of the actions will then receive (almost) optimal feedback, whereas the other action receives a feedback of 0. Due to the CALA update function, the μ of the underlying Gaussian will therefore shift drastically (e.g., we observed values of -10^6). As this is a highly undesirable effect, we chose to limit the terms of the update function. More precisely, we limit the term $\beta(x) - \beta(\mu)$ to the interval $[-\Phi(\sigma), \Phi(\sigma)]$. In essence, this addition has the same effect as a variable learning rate, which is not uncommon in literature (e.g., [5]). In normal cases, i.e., when the automaton is not near the discontinuity, the limit is hardly, if ever, exceeded. Near the discontinuity, it prevents drastic shifts. This addition to the learning rule therefore does not affect convergence.

The second problem arises when both the μ-action and the x-action of the automaton yield a feedback of 0 – i.e., the automaton receives no useful feedback at all. In this case, due to the CALA update function, the underlying Gaussian's μ and σ are not changed. Therefore, in the next learning round, there is a high probability that the automaton again receives a feedback of 0 for both actions. In other words, if this happens, the automaton is very likely to get stuck. We address this issue by including the knowledge that, if both μ and x yield a feedback of 0, the lowest action was nonetheless the best one. Therefore, we set $\beta(x) = \max(\beta(x), \mu - x)$, essentially driving the automaton's μ downward. Once again, in normal cases, the update function remains unchanged. In cases where the automaton receives no useful feedback, it can still update the parameters of the underlying Gaussian.

4. EXPERIMENTS AND RESULTS

We performed a set of experiments, which are summarized in Table 1. The agents used CALA for learning and the Homo Egualis utility function was applied to the feedback from the environment. The CALA parameters were set as outlined above. In addition, the agents started from an initial solution of equal sharing, i.e., μ for all n agents' CALA was set to $R \cdot \frac{1}{n}$ ($\sigma = 0.1\mu$). In the experiments, we used $R = 100$. All experiments lasted for 10000 rounds, were run 1000 times, and the Egualis parameters were set to $\alpha = 0.6$ and $\beta = 0.3$.[1] The number of agents varied between 2, 3, 4, 10 and 100, as denoted under 'Agents'. Whether or not punishment could be used by the agents is indicated in the 'Pun' column (i.e., with punishment enabled, agents could reject a solution for which they obtained a negative utility). The analytically determined solution, i.e., the solution resulting from playing optimally, which depends on the aforementioned columns, is indicated in the 'Solution' column (either the exact solution or the conditions that a solution must satisfy, if any). Whether or not the extended learning rule was used in the CALA, is indicated under 'Ext. LR.'.

Next, we show experimental results (average payoff and standard deviation; the values are separated per agent by a '/'). In every case, we also measure how many times a valid solution was found and subsequently maintained over the full 10000 rounds (results are displayed under 'Maint.'). Finally, we indicate whether the experiment can be considered a success; more precisely, we consider the experiment to be successful (+) if a valid solution was found and maintained in all experimental runs. An experiment is a failure (-) if a solution was not found and/or maintained in any run. Otherwise, i.e., a solution was found but not always maintained, an experiment was neither a success, nor a failure (○).

5. DISCUSSION

In this section, we discuss our results more elaborately. Moreover, we assess whether starting with a different initial solution than an equal split makes a difference. Due to the architecture used, it is not possible to start with a truly random initial solution; if any of the agents already rejects the initial solution, there is no information on which the learning process can be based. Thus, the CALA remain in this invalid initial solution, or, with the learning rule extensions, they all learn to request a payoff of 0. Therefore, we generate random, but valid initial solutions to determine how these affect the

[1] In the 'Game' column, we indicate experiments where different settings were used (1: β=0.7; 2: 200 experimental runs and λ and σ_L lowered by a factor 10).

Games with 2 agents (a)

Game	Agents	Pun.	Solution	Ext. LR.	Average	St. Dev.	Maint.	Result
UG[1]	2	no	50.0/50.0	no	50.1/49.9	0.2/0.2	100%	+
UG	2	no	100.0/0.0	no	100.0/0.0	0.0/0.0	100%	+
UG	2	yes	72.7/27.2	no	72.3/27.7	5.5/5.5	100%	+
NBG	2	yes/no	all$\geq$27.2	no	46.5/46.6	2.9/2.7	0%	-
NBG	2	yes/no	all$\geq$27.2	yes	48.2/48.2	2.4/2.4	100%	+

Games with 3 to 10 agents (b)

Game	Agents	Pun.	Solution	Ext. LR.	Average	St. Dev.	Maint.	Result
UG	3	yes	all$\geq$15.8	yes	41.0/41.0/18.0	1.6/1.5/1.7	100%	+
UG	4	yes	all$\geq$11.1	yes	29.0/29.0/29.0/13.0	1.5/1.5/1.5/1.6	100%	+
UG	10	yes	all$\geq$4.0	yes	10.5/10.5/.../6.7	1.1/1.1/.../2.0	100%	+
NBG	3	yes	all$\geq$15.8	yes	33.2/33.1/33.3	1.7/1.7/1.7	100%	+
NBG	4	yes	all$\geq$11.1	yes	24.5/24.5/24.5/24.5	1.6/1.6/1.6/1.6	100%	+
NBG	10	yes	all$\geq$4.0	yes	9.8/9.8/.../9.8	1.2/1.2/.../1.2	100%	+
NBG	3	no	any	yes	33.2/33.1/33.1	1.9/1.9/1.9	93%	○
NBG	4	no	any	yes	25.0/25.0/25.0/25.0	1.1/1.1/1.1/1.1	93%	○
NBG	10	no	any	yes	10.0/10.0/.../10.0	0.9/0.9/.../0.9	100%	+

Games with 100 agents (c)

Game	Agents	Pun.	Solution	Ext. LR.	Average	St. Dev.	Maint.	Result
UG[2]	100	yes	all$\geq$0.4	yes	0.98/0.98/.../2.2	0.3/0.4/.../1.7	100%	+
NBG[2]	100	yes	all$\geq$0.4	yes	0.96/0.92/.../1.0	0.3/0.4/.../0.4	100%	+
NBG[2]	100	no	any	yes	0.92/0.96/.../0.9	0.3/0.3/.../0.3	100%	+

Table 1: Results of our experiments in the Ultimatum Game (UG) and Nash Bargaining Game (NBG).

learning process. Since generating such a solution essentially entails solving a constraint satisfaction problem, we did this only for 2, 3, 4 and 10 agents.

Two-agent Ultimatum Game. In the two-agent Ultimatum Game, we use only one learning automaton; the last agent's behavior is static. Results are summarized in Table 1(a). In the first experiment, we set $\beta = 0.7$ and $\alpha = 0.6$. The first setting theoretically ensures that agent 1 gives 50% to agent 2, even in the absence of punishment. Initially, we therefore disable the punishment option. The automaton maintains to offer 50%, without any enforcement (i.e., punishment); with punishment, exactly the same happens (and punishment is never needed). When a different starting point is chosen than the 50-50 split, the automaton also converges to this solution. In the second experiment, we use $\beta = 0.3$ and $\alpha = 0.6$ and disable the punishment option. In the absence of punishment, the first agent can simply take the whole reward for himself, as predicted also by [17]. In the third experiment, we therefore use the same settings, but with punishment enabled, i.e., if the second agent obtains a utility value below 0, he rejects, leading to a payoff of 0 for both agents. With $\beta = 0.3$ and $\alpha = 0.6$, [17] predicts a payoff fraction of $\frac{0.6}{1+2\times0.6} \approx 0.27$ being given to the second agent. The learning process turns out to be robust with respect to the parameters used. As long as the initial setting is a valid solution, the same final solution is found. Thus, we see that our agents are capable of learning to play the two-player Ultimatum Game in a 'human' way.

Two-agent Nash Bargaining Game. In the two-player case, we need two learning agents and therefore also two CALA. Whenever the joint action of the CALA results in a summed payoff higher than R, both agents receive 0. Whenever the summed payoff is at most R, the Egualis function is applied to determine whether each agent considers their respective payoff to be fair. If not, they can choose to give both themselves and the other agent a payoff of 0. Results are summarized in Table 1(a). In the first experiment, we use $\beta = 0.3$ and $\alpha = 0.6$, enabling punishment. Clearly, any solution yielding a payoff of at least 27 for both agents is acceptable using the Egualis function. As can be seen in Table 1, the CALA do not learn a solution now. Therefore, in the second experiment, we introduce the extended learning rule, as outlined in Section 3.3. Typical results obtained using this extended rule are displayed in Figure 2. This time, the CALA find and maintain the correct solution; note that the solution is nearly Pareto-optimal as well as close to a 50-50 split, as predicted in literature. We observed that punishment was never used by the agents, even though it was possible. Choosing a valid starting point different than the equal split has no significant effect on outcomes. Thus, with the extended learning rule, a 'human' solution to the two-player Nash Bargaining Game can be learned.

Multi-agent Ultimatum Game. As has been outlined before, in the Ultimatum Game, agents take turns in taking some of the reward R for themselves. The last agent in the row gets what is left. Results are summarized in Table 1(b). As with the Nash Bargaining Game, using the standard settings of CALA for the multi-agent Ultimatum Game turns out to lead to invalid solutions. For this reason, we introduce the extended learning rule. In this case, the CALA can indeed learn to obtain and maintain a valid solution. Typical results for a three-player game are shown in Figure 3. We see that the last agent's utility is quickly decreased to a low positive value by keeping approximately 16 for this agent. The other two agents obtain an equal split of the remaining 84. Note that the first agent could have exploited the other agents; he could have obtained approximately 64 without the other two agents rejecting.

However, since all agents are learning simultaneously, both agent 1 and agent 2 are increasing their payoffs at the same time; at a certain point, they thus have reduced agent 3's utility value to 0. Then, if any agent wishes to increase his payoff, agent 3 will reject. Thus, agent 1 cannot exploit agent 2 unless agent 2 willingly lowers his payoff, which simply will not happen.[2] When we use a valid initial solution different from an equal split, we see that agent 1 may obtain a higher payoff than agent 2, if his payoff was already higher in the initial solution. However, with a set of 1000 randomly generated valid initial solutions, we see that the difference is small (i.e., agent 1 obtains 42 instead of 41 – the standard deviation increases from 1.6 to 4). Results generalize well over an increasing number of agents; with 4 and 10 agents, a valid solution in which the last agent is 'exploited' is found and maintained every time, with the other agents achieving an equal split. Again, choosing random valid initial solutions instead of an equal split does not affect results in a noticable manner. With 100 agents, the solution is successfully maintained in only 81% of the experimental runs with standard settings for the CALA's parameters. Since agents now each have to obtain a much smaller portion of the reward R, especially the learning rate could be lowered to increase convergence. Indeed, with a lower learning rate and a lower σ_L (i.e., ten times lower), every experimental run is a success (see Table 1(c)). Note that, in case of success, the last agent receives a rather high payoff, due to the fact that the other 99 agents can only approximate the optimal payoff of 1. Thus, we can conclude that a multi-agent Ultimatum Game poses no difficulties for our agent architecture; a 'human' solution can always be found.

Multi-agent Nash Bargaining Game. As with the Ultimatum Game, we scale up our problem to include more agents. Results for this experiment are summarized in Table 1(b). We immediately start with CALA that include the extended learning rule. With 3, 4 and 10 agents and punishment possible, a valid solution is always found and maintained. This solution is always very close to a Pareto-optimal equal split, as can be seen in Table 1. The same happens in case we use a random valid initial solution instead of an equal split. With 100 agents, a valid solution is often found, but not maintained in about half of the cases. Once again, this is caused by the fact that we did not adapt the learning rate of the CALA. Lowering the learning rate and σ_L with a factor 10, we can achieve success in every experiment (see Table 1(c)). Thus, a multi-agent Nash Bargaining Game can indeed be played in a 'human' way by our agent architecture.

Multi-agent Nash Bargaining Game without punishment. Since we saw that in the two-agent case, the Nash Bargaining Game was played without any agent punishing, we assessed the effects of disabling the punishment option in this game. Results are summarized in the last three rows of Table 1(b). Interestingly, the game can be often solved if agents do not have the possibility to punish, both in initial equal-split solutions as in valid random initial solutions. However, when we add the possibility to punish, solutions are easier to be found and maintained because agents are slightly more conservative (i.e., less greedy). It is quite easy to see why this happens: due to the rules of the game, an overly greedy agent is still punished, if not by other agents, then by the environment. Therefore, regardless of the initial solution, an agent increasing his own payoff too much is immediately given negative feedback. As a result, valid solutions are found and maintained only slightly less often with punishment disabled than with punishment enabled. Moreover, it is interesting to note that solutions are on average closer to a Pareto-optimal solution. Once again, with 100 agents, lowering the learning rate and σ_L with a factor 10 increases the number of experiments that were finished successfully (see Table 1(c)). Thus, we see that the possibility to punish is not really necessary for CALA to learn 'human' solutions for the Nash Bargaining Game, but it does increase agents' ability to learn such solutions.

[2]Research with humans has shown that only a minority of human subjects actually exploits the other player(s). For instance, in [11], we saw that people tend to give away 50% even if the stakes are very high. Thus, the fact that the first agent does not exploit only makes it more 'human'.

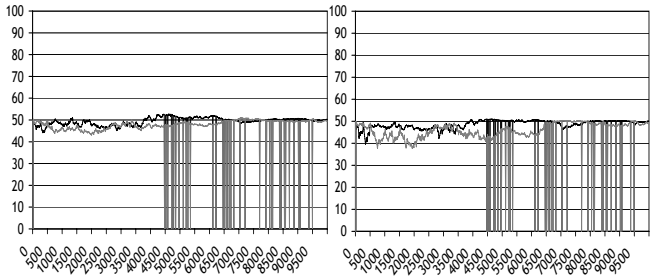

Figure 2: Two agents playing the Nash Bargaining Game; payoffs (left) and utilities (right) evolve over time.

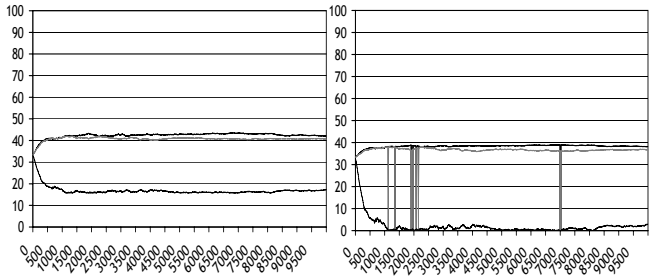

Figure 3: Three agents playing the Ultimatum Game; payoffs (left) and utilities (right) evolve over time.

6. CONCLUSION

In this paper, we presented our work in the area of human-inspired fairness in adaptive multi-agent systems. In essence, there are two distinct reasons for incorporating fairness. First, multi-agent systems often perform tasks for humans, or even interact with them. Since research shows that humans are not individually rational, the classical, individually rational agent model may not be sufficient to obtain alignment with human expectations. Second, there are multiple examples of multi-agent interactions in which following an individually rational strategy actually leads to bad results. Strategies that consider concepts such as social welfare or fairness have been shown to perform better.

We presented a straightforward architecture which enables the inclusion of a descriptive model of human fairness, i.e., the inequity-averse Homo Egualis utility function, into an adaptive multi-agent system, driven by Continuous Action Learning Automata. Homo Egualis was used in an explicit, computational manner, in contrast to earlier work [46], in which the agents' architecture is only inspired by Homo Egualis. The resulting system has been used to learn solutions for the Ultimatum Game and the Nash Bargaining Game, both of which are abstract models of actual bargaining interactions. From our experiments, we can draw two conclusions, viz. (1) using this adaptive agent system, we can find and maintain valid solutions to both games under study, even with many agents learning together; (2) the solutions found by the agents conform to

solutions found using an existing descriptive model, which in turn adequately conforms to solutions found using human subjects. The proposed methodology therefore presents a possibility to integrate explicitly human-inspired fairness in adaptive multi-agent systems.

In future work, we wish to apply our findings in actual applications of multi-agent systems in which fairness is important. More precisely, we wish to look at fairness in scheduling of aircraft deicing [24]. Moreover, the Homo Egualis model (and thus, our adaptive agents) should be extended by including notions such as bargaining powers, priorities and reputation, which have been shown to be important for humans [11].

7. REFERENCES

[1] H. Aldewereld. *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*. PhD thesis, Universiteit Utrecht, 2007.

[2] K. Basu. The Traveler's Dilemma. *Scientific American*, Volume 296, Number 6:68–73, 2007.

[3] K. Binmore. *Natural Justice*. Oxford University Press, 2005.

[4] S. Bowles, R. Boyd, E. Fehr, and H. Gintis. Homo reciprocans: A Research Initiative on the Origins, Dimensions, and Policy Implications of Reciprocal Fairness. *Advances in Complex Systems*, 4:1–30, 1997.

[5] M. H. Bowling and M. M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[6] G. Charness and M. Rabin. Understanding Social Preferences with Simple Tests. *Quarterly Journal of Economics*, 117:817–869, 2002.

[7] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar, and P. Sousa. Issues in Multiagent Resource Allocation. *Informatica*, 30:3–31, 2006.

[8] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A Short Introduction to Computational Social Choice. In *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*, volume 4362 of *LNCS*, pages 51–69. Springer-Verlag, 2007.

[9] A. Dannenberg, T. Riechmann, B. Sturm, and C. Vogt. Inequity Aversion and Individual Behavior in Public Good Games: An Experimental Investigation. *SSRN eLibrary*, 2007.

[10] S. de Jong, K. Tuyls, and I. Sprinkhuizen-Kuyper. Robust and Scalable Coordination of Potential-Field Driven Agents. In *Proceedings of IAWTIC/CIMCA 2006, Sydney*, 2006.

[11] S. de Jong, K. Tuyls, K. Verbeeck, and N. Roos. Priority awareness: towards a computational model of human fairness for multi-agent systems. *Adaptive Agents and Multi-Agent Systems III - Lecture Notes in Artificial Intelligence*, 4865, 2008.

[12] I. Erev and A. E. Roth. Predicting how people play games with unique, mixed strategy equilibria. *American Economic Review*, 88:848–881, 1998.

[13] A. Falk and U. Fischbacher. A theory of reciprocity. *Games and Economic Behavior*, 54:293–315, 2006.

[14] E. Fehr. Don't lose your reputation. *Nature*, 432:499–500, 2004.

[15] E. Fehr and S. Gaechter. Fairness and Retaliation: The Economics of Reciprocity. *Journal of Economic Perspectives*, 14:159–181, 2000.

[16] E. Fehr and S. Gaechter. Altruistic punishment in humans. *Nature*, 415:137–140, 2002.

[17] E. Fehr and K. Schmidt. A Theory of Fairness, Competition and Cooperation. *Quarterly Journal of Economics*, 114:817–868, 1999.

[18] H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2001.

[19] W. Gueth, R. Schmittberger, and B. Schwarze. An Experimental Analysis of Ultimatum Bargaining. *Journal of Economic Behavior and Organization*, 3 (4):367–388, 1982.

[20] C. Hauert, S. D. Monte, J. Hofbauer, and K. Sigmund. Volunteering as red queen mechanism for cooperation in public goods games. *Science*, 296:1129–1132, 2002.

[21] C. Hauert, A. Traulsen, H. Brandt, M. Nowak, and K. Sigmund. Via freedom to coercion: the emergence of costly punishment. *Science*, 316:1905–1907, 2007.

[22] J. Henrich, R. Boyd, S. Bowles, C. Camerer, E. Fehr, and H. Gintis. *Foundations of Human Sociality: Economic Experiments and Ethnographic Evidence from Fifteen Small-Scale Societies*. Oxford University Press, 2004.

[23] M. Jackson. Mechanism Theory. *Humanities and Social Sciences*, October:228–277, 2000.

[24] X. Mao, A. ter Mors, N. Roos, and C. Witteveen. Agent-Based Scheduling for Aircraft Deicing. In P.-Y. Schobbens, W. Vanhoof, and G. Schwanen, editors, *Proceedings of the 18th Belgium - Netherlands Conference on Artificial Intelligence*, pages 229–236. BNVKI, October 2006.

[25] M. Milinski, D. Semmann, and H. Krambeck. Reputation helps solve the tragedy of the commons. *Nature*, 415:424–426, 2002.

[26] K. Narendra and M. Thathachar. *Learning Automata: An introduction*. Prentice-Hall International, 1989.

[27] J. Nash. Equilibrium Points in N-person Games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.

[28] J. Nash. The Bargaining Problem. *Econometrica*, 18:155–162, 1950.

[29] M. Nowak, K. Page, and K. Sigmund. Fairness versus reason in the Ultimatum Game. *Science*, 289:1773–1775, 2000.

[30] R. Nydegger and H. Owen. Two-person bargaining, an experimental test of the Nash axioms. *International Journal of Game Theory*, 3:239–250, 1974.

[31] H. Oosterbeek, R. Sloof, and G. van de Kuilen. Cultural Differences in Ultimatum Game Experiments: Evidence from a Meta-Analysis. *Experimental Economics*, 7:171–188, 2004.

[32] K. Panchanathan and R. Boyd. Indirect reciprocity can stabilize cooperation without the second-order free rider problem. *Nature*, 432:499–502, 2004.

[33] D. C. Parkes. Computational Mechanism Design. In *Lecture notes of Tutorials at 10th Conf. on Theoretical Aspectsof Rationality and Knowledge (TARK-05)*. Institute of Mathematical Sciences, University of Singapore, 2008.

[34] J. A. Rodriguez-Aguilar. *On the design and construction of agent-mediated electronic institutions*. PhD thesis, Monografies de l'Institut d'Investigació en Intelligència Artificial, 2003.

[35] A. Roth and M. Malouf. Game Theoretic Models and the Role of Information in Bargaining. *Psychological Review*, 86:574–594, 1979.

[36] S. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach*. Prentice Hall, 2 edition, 1995.

[37] F. Santos, J. Pacheco, and T. Lenaerts. Cooperation Prevails When Individuals Adjust Their Social Ties. *PLoS Comput. Biol.*, 2(10):1284–1291, 2006.

[38] F. Santos, J. Pacheco, and T. Lenaerts. Evolutionary Dynamics of Social Dilemmas in Structured Heterogeneous Populations. *Proc. Natl. Acad. Sci. USA*, 103:3490–3494, 2006.

[39] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1507–1512, 2007.

[40] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.

[41] K. Sigmund, C. Hauert, and M. Nowak. Reward and punishment. *Proceedings of the National Academy of Sciences*, 98(19):10757–10762, 2001.

[42] H. Simon. *Models of Man*. John Wiley, 1957.

[43] M. Thathachar and P. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, 2004.

[44] K. Tuyls and A. Nowe. Evolutionary Game Theory and Multi-Agent Reinforcement Learning. *The Knowledge Engineering Review*, 20:63–90, 2005.

[45] J. Vazquez-Salceda, H. Aldewereld, and F. Dignum. Norms in multiagent systems: from theory to practice. *Comput. Syst. Sci. Eng.*, 20(4), 2005.

[46] K. Verbeeck, A. Nowé, J. Parent, and K. Tuyls. Exploring Selfish Reinforcement Learning in Repeated Games with Stochastic Rewards. *Journal of Autonomous Agents and Multi-Agent Systems*, 14:239–269, 2007.

[47] T. Yamagishi. The provision of a sanctioning system as a public good. *J. Person. and Soc. Psych.*, 51(1):110–116, 1986.