Auton Agent Multi-Agent Syst (2011) 22:103–126
DOI 10.1007/s10458-010-9122-9

Human-inspired computational fairness

Steven de Jong · Karl Tuyls

Published online: 21 February 2010

Abstract In many common tasks for multi-agent systems, assuming individually rational agents leads to inferior solutions. Numerous researchers found that *fairness* needs to be considered in addition to individual reward, and proposed valuable computational models of fairness. In this paper, we argue that there are two opportunities for improvement. First, existing models are not specifically tailored to addressing a class of tasks named *social dilemmas*, even though such tasks are quite common in the context of multi-agent systems. Second, the models generally rely on the assumption that all agents will and can adhere to these models, which is not always the case. We therefore present a novel computational model, i.e., *human-inspired computational fairness*. Upon being confronted with social dilemmas, humans may apply a number of fully decentralized sanctioning mechanisms to ensure that optimal, fair solutions emerge, even though some participants may be deciding purely on the basis of individual reward. In this paper, we show how these human mechanisms may be computationally modelled, such that fair and optimal solutions emerge from agents being confronted with social dilemmas.

Keywords Multi-agent systems · Reinforcement learning · Fairness · Human-inspired mechanisms · Social dilemmas

1 Introduction

In the last few years, researchers have proposed various ways to address the limitations of purely self-interested agents [26]. Especially research in the area of computational social choice [10] devoted a great deal of attention to the development of computational models

S. de Jong (✉)
Computational Modelling Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
e-mail: drstevendejong@gmail.com

S. de Jong · K. Tuyls
Department of Knowledge Engineering, Maastricht University, PO Box 616, 6200 MD Maastricht,
The Netherlands

that include considerations such as *fairness* in addition to pure self-interest. We see two opportunities for improvement of the computational models of fairness proposed thus far.

First, current computational models are generally not explicitly tailored to addressing a class of tasks called *social dilemmas* [35]. These are tasks in which agents may choose between group reward (social cooperation), and individual reward (defection). The dilemma lies in the fact that cooperative joint strategies yield the highest reward for everyone, while individuals gain most by defecting against cooperators (i.e., the defective joint strategy constitutes a Nash equilibrium). Social dilemmas may occur frequently in practical applications of multi-agent systems. An example is multi-agent resource allocation, which is indeed a central task in multi-agent systems [9,20]. Even outside domains in which agents perform resource allocation explicitly, agents are using computational resources (e.g., bandwidth, memory or CPU cycles), which usually need to be shared [20]. In a typical scenario in the context of sharing computational resources, we have a large number of agents sharing a finite amount of these resources, with each agent potentially having specific preferences. If agents have to allocate resources to themselves and other agents, this essentially entails that they need to solve a social dilemma [53]. A model that is explicitly constructed to address social dilemmas may allow agents to find better solutions than a model that is not.

The second opportunity for improvement concerns the fact that existing computational models of fairness often rely on the assumption that all agents in the multi-agent system are willing to adhere to these models, and also capable of doing this. This assumption does not always hold. Most importantly, some agents may not be willing to behave according to a model that is external to them. Multi-agent systems are often open [1], i.e., agents that we did not design may enter the system. Such agents may be human or artificial, and they may be individually rational or even actively exploiting other agents. Computational models should be able to deal with such openness, i.e., it should not be necessary to assume that all agents will use the models. Moreover, even if we in fact assume that all agents are willing to adhere to the models, we need to take into account that they may not be capable to do so. For instance, many existing solution concepts and optimality criteria are computationally intractable or (partially) centralized (e.g., procedures aimed to maximize the collective utility functions known from welfare economics and computational social choice [9,10]), which means that they are not readily applicable to a large-scale decentralized multi-agent system.

We present a novel model, *human-inspired computational fairness*, which is inspired by the human way of dealing with social dilemmas. Humans show remarkable ability when they are confronted with such dilemmas [33]. They use a number of fully decentralized sanctioning mechanisms in order to enforce a desirable (i.e., fair and optimal) solution. In the presence of such mechanisms, striving for a fair and optimal solution actually becomes best even when a number of participants decide upon their actions based on their individual reward only. Interestingly, the decision whether or not to apply these mechanisms essentially entails a second-order social dilemma [54]. Researchers found numerous explanations why humans do apply such fairness mechanisms anyway. In previous work, we translated a number of these explanations to a computational equivalent.

In this paper, we unify our previous work by presenting a template model for human-inspired computational fairness, as well as two instantiations of the template model. Agents' behavior, resulting from our computational models of fairness, is intuitive for human observers. Moreover, the models are fully decentralized, computationally tractable, as well as quite robust with respect to agents trying to exploit the fairness of other agents. The models thus have a number of appealing properties that may not all be present in existing models.

The remainder of the paper is structured as follows. We start in Sect. 2 by outlining existing work related to the work presented in this paper. In Sect. 3, we take a closer look at social dilemmas in resource allocation and multi-agent systems. In Sect. 4, we briefly look at the possibilities and impossibilities of applying existing definitions and procedures for fairness in the context of social dilemmas. In Sect. 5, we present a template model for human-inspired computational fairness, based on a number of requirements that we also discuss, as well as a general approach for building a multi-agent system that addresses social dilemmas based on individual learning. In Sect. 6, we present two instantiations of our template model, based on models of observed human behavior, and in Sect. 7, we show how the resulting computational models allow agents that are learning individually to find desirable allocations of resources. In Sect. 8, we conclude the paper.

2 Related work

In this section, we discuss the work that this paper is based on. This work can be divided in three categories. After discussing these categories, we explicitly mention the contributions of the current paper.

2.1 Modelling human fairness

Computational models inspired by human fairness in social dilemmas are currently receiving a great deal of attention in behavioral literature. However, existing research is intended to be descriptive rather than computational. It generally follows a distinct two-step approach.

First, most research starts by performing carefully controlled experiments with human subjects. It turns out that humans are well able to optimize their individual reward, while taking into account the fairness of their actions. They respond to unfair interactions using a number of mechanisms. Most interesting for our work are two sanctioning mechanisms [40], i.e., (1) *altruistic punishment*, implying that someone who is perceived to act in an unfair way is treated with an immediate negative effect on his individual reward, at a smaller cost to the punisher [8,23,24], and (2) *withholding action*, implying that an unfair actor is somehow excluded from future interactions, even if interacting with this actor may lead to a positive reward [44]; an alternative name for this mechanism is volunteering [29]. Interestingly, these two mechanisms lead to a *second-order social dilemma*, as they are costly to the individual wanting to use them, without a directly perceivable reward. Researchers have therefore tried to find mechanisms explaining why humans behave as they do in the second-order social dilemma.

To validate proposed mechanisms, researchers proceed to the second step: proposed mechanisms are modelled in a computational context. This is generally done in one of three manners, i.e., (1) using a multi-agent learning approach, e.g., evolutionary algorithms [as in 43,44], or (2) using an evolutionary-game-theoretic analysis [as in 24,29,37], or (3) using statistical physics [as in 11]. Experiments are performed to determine the effect(s) of the proposed mechanisms on learned strategies in (selected) social dilemmas; if these learned strategies indeed become more fair or cooperative, this provides support for the proposed mechanisms. Many mechanisms (such as reputation [36,37], volunteering [29], priorities [18], and rewiring in social networks [44,55]) have been proposed and supported according to this procedure.

While models proposed in behavioral literature contain interesting and useful elements for the research at hand, the fact that they are intended to be descriptive rather than explicitly computational leads to certain abstractions. For instance, many models in the literature are applied to abstract social dilemmas with only two possible actions [e.g., 43,44]; also, many models work based on the imitation of successful strategies in the population [e.g., 44,55]. In realistic applications of multi-agent systems, there will usually be more than two actions to choose from, or even a continuum of actions [19]. Moreover, in multi-agent systems, individual learning, based only on agents' individual reward (e.g., reinforcement learning [48]), is much more common than learning by imitation. The novel models proposed in this paper allow agents to select actions from a continuous action space by learning individually.

2.2 Computational social choice

The research area of computational social choice [10] builds upon definitions of optimality and fairness given in welfare economics. A number of definitions and axioms are given in social choice theory [45] to describe what constitutes a desirable solution. Example definitions of desirable solutions include those that optimize utilitarian social welfare (i.e., average reward), or those that optimize egalitarian social welfare (i.e., minimal reward). Researchers propose computational models (e.g., negotiation [21]) which allow agents to find the most desirable solution according to a certain definition of fairness or optimality. A great deal of work is directed towards analysing the computational complexity of such models. Presently, definitions and procedures from computational social choice are finding their way into multi-agent systems [20], especially in multi-agent resource allocation [9]. We note that utilitarian social welfare is (still) most commonly used in multi-agent systems.

Although definitions and procedures proposed by computational social choice are very useful, especially due to the thorough analytical work surrounding them, there is still room for improvement regarding practical applicability, for two reasons that we already discussed in Sect. 1 and briefly restate here. First, there is generally no explicit attention for social dilemmas, even though social dilemmas occur regularly in typical tasks for multi-agent systems (especially resource allocation). Second, many procedures cannot be practically applied to large-scale, decentralized and open systems, for instance due to the assumption that all agents in the system will and can adhere to the proposed procedures. A pragmatic model, driving agents to a sufficient, intuitively appealing approximation of the optimal solution, is appropriate. In Sect. 4, we provide more details on this matter.

2.3 Human-inspired computational fairness

We were not the first to realize that it may be useful to incorporate elements of human decision-making in multi-agent learning. Verbeeck et al. [53] use the inequity-averse Homo Egualis model (see Sect. 6.1) as an inspiration for achieving a balance between optimality and fairness in multi-agent systems, more precisely, in coordination games. They focus on games with competition between players, where the overall performance is measured on a social level (e.g., performance is as good as that of the poorest-performing player, which corresponds to egalitarian social welfare). The usability of their approach is demonstrated in a practical task, i.e., load balancing, which they effectively map to the social dilemma named the tragedy of the commons [28].

We extended the work of Verbeeck et al. in two ways, i.e., (1) we aimed to use behavioral models, such as Homo Egualis (and also social networks), as literally as possible to improve alignment with (successful) human behavior, and (2) we aimed to address more social dilemmas. We here provide a short overview of our previously published work. First, we looked at a (relatively small) group of agents learning to play the Ultimatum Game and the Nash Bargaining Game [16], as well as the Public Goods Game [14], using inequity aversion [24]. Second, we wanted to increase the scale of the multi-agent systems investigated; we decided to do this using social networks, inspired by the work of Santos et al. [43,44]. In contrast to that work, in which agents learn by imitating their neighbors' strategies, we continued to follow an approach based on individual learning. We investigated large groups of agents learning to play the Ultimatum Game [19] and the Public Goods Game [15], based on a scale-free network topology [3]. Preliminary experiments revealed the structure of the network was not a strong influence on strategies agents converged to; this finding was in contrast with what was reported for agents that learn by means of imitation [52].

2.4 Contributions of the current paper

In this paper, we provide three contributions. First (Sect. 4), we look at the applicability of collective utility functions, as sometimes used in analytical approaches to fairness (e.g., welfare economics, computational social choice), to social dilemmas, and discuss why our proposed approach might be an interesting alternative. Second (Sect. 5), we discuss the common elements of the work done so far. We analyze the requirements for human-inspired computational fairness models and provide a template human-inspired computational fairness model that matches these requirements. The template model is based on useful aspects of human behavior in social dilemmas, i.e., those aspects that facilitate individually learning agents to find desirable solutions. We also give a general approach, based on individual learning, which allows agents to behave according to the template model (and instantiations of this model). As a third contribution, in Sect. 6, we give two instantiations of the template model, i.e., a computational model of inequity aversion [14,16] in Sect. 6.1, and a model based on social networks [15,19] in Sect. 6.2. Results obtained by these models are shown (in Sect. 7) for the Ultimatum Game [16] (Sect. 7.1) and the Public Goods Game [15] (Sect. 7.1), respectively.

3 Social dilemmas

Social dilemmas [35] are problems in which there is a conflict between group reward (cooperation) and individual reward (defection). The group of agents achieves the highest reward if every agent cooperates, but individual agents maximize their reward by defecting against cooperators. If every individual agent reasons in this manner, the result is that every agent defects, yielding a substantially lower reward than what can be be achieved if every agent cooperates. An early example of a social dilemma is the well-known Prisoners' Dilemma [2]. Models assuming individual rationality have shown limited value when they are applied to social dilemmas, as they predict that players will defect by focussing purely on their individual reward [26]. In reality, human players regularly cooperate [24,38]. Of the many types of social dilemmas, we focus on two that are relevant in the context of multi-agent systems and have already been modelled by means of abstract games, viz. the Ultimatum Game and the Public Goods Game. We explain these social dilemma games, as well as human behavior, below. Although the optimal strategies to play in each game may look trivial, they are in fact complicated to reach. We explain why this is the case. At the end of the section, we draw parallels between the games and typical tasks for multi-agent systems.

3.1 The Ultimatum Game

3.1.1 The game

The Ultimatum Game (UG) [27] is a bargaining game, played by two agents. The first agent proposes how to divide a (continuous) resource R between it and the second agent. If the second agent accepts the proposed division, the first obtains its demand and the second obtains the rest. If the second agent rejects the proposal, neither obtains anything. The individually rational solution (i.e., the Nash equilibrium) to the UG is for the first agent to offer the smallest positive amount to the other agent. After all, the other agent can then choose between receiving this amount by agreeing, or receiving nothing by rejecting. We note that some researchers do not consider the UG to be a social dilemma [e.g., 34]. However, Sigmund et al. [47] show that the analogy between offering high amounts in the UG and cooperation in a social dilemma (and similarly, offering low amounts and defection) can be shown with full mathematical rigor.

3.1.2 Human behavior

Humans usually do not choose the individually rational solution in the UG. Hardly any first player proposes offers that lead to large differences between the agents, and hardly any second player accepts such offers. Meta-studies of many experiments with humans [5,38] report that the average offer in the two-player UG is about 40%, with 16% of the offers being rejected by the other player. We replicated this finding in our own experiments [18]. Cross-cultural studies performed in 'primitive' hunter-gatherer cultures have shown that, although the average offer proposed and accepted differs substantially by culture, there is a general tendency to deviate from an individually rational solution [31].

3.1.3 Why the game is difficult

Although the UG has a clearly defined Nash equilibrium, playing according to this equilibrium will make most human players reject, yielding a payoff of 0 instead of the expected amount (nearly R). In contrast, a vast majority of human players will accept a 50–50 split—offering a 50–50 split to a human player is therefore much safer than offering the smallest possible amount. However, offering a 50–50 split is generally not the *optimal* strategy against a human player. A majority of human players accept a 50–50 split because in fact, e.g., a 30–70 split would also be accepted. In this case, offering a 30–70 split would be optimal. What the optimal strategy in the UG is thus depends completely on the acceptance threshold of the second player, and we might not know this threshold.

3.1.4 How the game maps to multi-agent systems

The dilemma present in the UG is also prominently present in applications that are meant to provide a service to humans. Examples include resource gathering as well as scheduling and planning [41]. Clearly, human customers must be satisfied with the given service, as they will otherwise decide to end their customer relationship with the service provider (potentially moving to a competitor that provides better service). What seems optimal from an individually rational perspective, may not be perceived as optimal (or fair) from the perspective of customers. Also, given the fact that the UG maps to other social dilemmas such as the Prisoners' Dilemma [47], a multi-agent system that can successfully find agreement in the UG (which basically implies balancing cooperation and defection to match the behavior of the other agents) may also be used in other applications that require agents to deal with social dilemmas.

3.2 The Public Goods Game

3.2.1 The game

In the Public Goods Game (PGG), which is typically played repeatedly by $n = 3, \ldots, 10$ players, every player can invest (part of) a private amount C in a common pool. All players simultaneously choose how much to invest. Then, everyone receives a share of the amount in the common pool, multiplied by a certain factor r $(1 < r < n)$ and divided equally among the players. To gain the most profit, everyone should cooperate by contributing their entire private amount C. However, every player can gain from solely not contributing (i.e., from defecting). Thus, the Nash equilibrium is for every player to defect by not contributing at all, leading to a group profit that is much lower than the optimal profit.

3.2.2 Human behavior

Typical human players start with a relatively cooperative strategy. Over time, they realize that more is gained by those that contributed less, which makes them lower their contribution until they are defective. This does not happen if they may perform altruistic punishment [23,54], i.e., if they are allowed to give up a small amount c (after every game) to decrease the gain of another player by a larger amount e. Humans often punish each other, and this punishment successfully enforces high contributions to the common pool.

3.2.3 Why the game is difficult

Although human players consistently apply punishment (successfully) if they are allowed, this is not a rational decision. Punishment is not a dominant strategy, since players may resort to second-order free-riding [39], i.e., contributing positive amounts while refusing to punish others that do not contribute. Many possible explanations exist of why humans punish anyway, but none of the existing explanations lead to stable cooperative solutions [14,17]. The PGG is a multi-player Prisoners' Dilemma with a continuous action space. Investing the entire amount C is optimal, but if everyone does so, we are individually much better off by refusing to invest. Motivating players to play the optimal strategy (or to apply punishment) is therefore not trivial.

3.2.4 How the game maps to multi-agent systems

We may argue that almost all applications of multi-agent systems have to deal with a dilemma similar to the PGG as they require the sharing of computational resources and the investment of effort, even though free-riding would be possible. A prime example here, apart from resource allocation (see Sect. 1), is load balancing [7,53].

4 Existing computational models of fairness

Computational models of fairness, as for instance investigated and formalized extensively in the field of computational social choice [10], provide a good starting point for studying and obtaining (computational) fairness in multi-agent systems. Computational social choice builds upon formal definitions and axioms concerning optimality and fairness, as provided by welfare economics, and aims to find procedures that allow agents to find an optimal (fair) solution, given a certain definition of optimality (fairness). In this section, we will restrict ourselves to one possible approach pursued in computational social choice, i.e., the definition of a *collective utility function* (CUF) with which optimality and fairness may be measured, and the subsequent development of a procedure that allows agents to maximize the value of this function. Numerous such procedures exist, e.g., based on negotiation [22,25] or (combinatorial) auctions [32,42].

A CUF $S : \mathbb{R}^n \rightarrow \mathbb{R}$ maps agents' utilities u_i (or alternatively, rewards), as given in a utility vector u, to one single real value $S(u)$, expressing social welfare. We may then devise a procedure allowing all agents to behave in a manner that maximizes $S(u)$, and thus to obtain a desirable solution. To select a suitable CUF, we may consider a number of axioms that we may or may not want to see reflected in the function; similarly, procedures aimed at optimizing a certain CUF have been extensively analyzed in the literature.

Under the assumption that all agents in the system will and can adhere to a procedure that optimizes the selected CUF, such a function may well be used to evaluate cooperative, desirable solutions to social dilemmas. One particularly interesting CUF in this respect is the *Nash product* (i.e., $S(u) = \Pi_i u_i$), which not only considers the efficiency of a certain solution, but also aims at reducing the inequality (and increasing the fairness) of the solution [9,20]. With this function, we are able to give the highest social welfare value to the most desired solution. As an example, we apply this function to a two-player two-strategy PGG (without punishment). Given a PGG with $r = 1.5$ and $C = 10$, the social welfare obtained if both agents defect is $S(u) = 0$. If one agent contributes 10 and the other 0, the common pool becomes $10 \cdot 1.5 = 15$, which is equally divided over the agents. The agents thus obtain profits of $7.5 - 10 = -2.5$ and $7.5 - 0 = 7.5$. The Nash product is usually only defined for non-negative utilities. To respect this requirement, we set all negative values to zero, i.e., the agents have a utility of 0 and 7.5, respectively. The Nash product then becomes $S(u) = 0 \cdot 7.5 = 0$. If both agents contribute 10, the social welfare is $S(u) = 5 \cdot 5 = 25$. Clearly, according to the Nash product, contributing is best here.

As has been said earlier, applying a measure of fairness, such as a CUF, only works well if we can assume that all agents are willing to adhere to a procedure that optimizes the measure in question, and are also capable of doing so. There are a number of reasons why an approach based on an explicit, formal measure of fairness may not work in the context of social dilemmas, and why a (less formal) human-inspired approach may work better. Clearly, our approach is not an intended replacement for the great deal of important analytical work that already exists; we only aim to present an intuitively appealing and pragmatic approach for obtaining fairness in multi-agent systems.

1. In case of open multi-agent systems, the assumption that all agents are willing to behave in a fair manner (e.g., to use a procedure that optimizes a CUF) does not hold. There may be agents that we do not control. These agents may not be interested in behaving fairly, or may even be trying to exploit agents that are. A human-inspired approach explicitly counters this problem: if we allow the part of the multi-agent system that we control to use mechanisms such as punishment or withholding action, based on their (expected)

utility, and set up an environment in which these mechanisms may affect every agent (including ones we do not control), we are able to force even individually rational agents to care for the reward of others, as failing to do so becomes costly.

2. In case of large-scale multi-agent systems, we cannot assume that the joint strategy leading to the optimal social welfare may be found by the agents. A procedure that establishes an optimal strategy for agents may be computationally intractable.[1] This implies that only an approximation of the optimal strategy is possible. With human-inspired mechanisms, agents have a decentralized, tractable manner of achieving intuitively appealing approximations of desired solutions in the social dilemma games investigated.[2]
3. Even under the assumption that all agents can and will optimize a certain measure of fairness (e.g., a CUF), there are many measures to choose from. Selecting a suitable CUF requires careful investigation.[3] The axiomatic foundations of CUFs are generally helpful in selecting a suitable function for the task at hand. However, the mapping of axioms to social dilemmas is not always transparent. For instance, in the prequel, we discussed that the Nash product is suitable to obtain a good solution in the PGG. The Nash product satisfies the axiom of scale-independence, which enforces the property of being independent from the way agents measure their individual utilities [9]. It is not immediately clear why this property would be useful in the PGG. In case of human-inspired mechanisms, there is also a great deal of choice (for instance, concerning the individual utility function employed by the agents that we do control); however, in this case, we can choose based on directly observed effects on human behavior in social dilemmas. Choices can therefore be done intuitively, based on extensive literature research.
4. Given a certain suitable optimality criterium (e.g., a CUF), we have not automatically found an optimal policy for agents. Complex procedures may be needed to allow agents to find a suitable policy [10]. In contrast, with punishment (and/or withholding action), there is a clear incentive to achieve a cooperative solution, even for simple learning agents that update their strategy only on the basis of individual reward. The motivation to cooperate is driven by the motivation to punish, and this motivation is coming from individual agents instead of an external factor (a CUF and/or associated procedures).

Thus, although existing work definitely provides a good starting point for achieving fairness in multi-agent systems, the inclusion of human-inspired mechanisms may address a number of issues that prevent learning agents from finding an optimal solution, while relaxing (some of) the formality associated with existing work.

[1] For instance, with two agents playing a PGG with two strategies, using CUFs, we already need to consider four different joint strategies, and therefore also four different CUFs (Nash products). With n agents and m strategies, we would need to consider n^m Nash products. In a continuous strategy space (which we use in our social dilemmas), m is actually infinite.

[2] We note that we do not imply the proposed approach will work for *all* social dilemmas. For instance, in the Travelers' Dilemma [4], there is no possibility to punish, and human behavior can also not be sufficiently explained by models such as inequity aversion, which we use extensively in our work. However, the approach does work in two social dilemma games that model typical interactions in the context of multi-agent systems.

[3] Even in the two-strategy PGG, we could select the wrong function, or the wrong parameters. For instance, using the k-rank dictator function [9], both agents contributing C would be best for $k = 1$, whereas single-sided defection would be best for $k = 2$.

5 The foundations of human-inspired computational fairness

In this section, we first outline three requirements that need to be met by computational models of human-inspired fairness. Then, we provide a template model that builds upon these requirements [13]. Finally, we present a general approach for individual learning.

5.1 The three requirements

First, the models should be rooted in a game-theoretic background. Game theory [6,26] provides us with a well-established, well-defined manner to describe interactions between multiple parties, where the individual reward of each party may be formalized using a *utility function*. As such, it is a good basis for (learning in) multi-agent systems [46]. Second, the models should be computationally applicable, i.e., in a setting of multi-agent systems addressing social dilemmas. Many existing solution concepts (e.g., the Nash equilibrium, or CUFs), as well as many existing models of human fairness (e.g., the Homo Egualis utility function, see Sect. 6.1), actually do not meet this requirement directly, for instance due to tractability issues or insufficiently smooth utility functions. Third, the models should enable adaptive agents to mimic human fairness mechanisms. Two human mechanisms to maintain fairness are of interest to us, viz. *altruistic punishment* and *withholding action*.[4] In order to transfer these mechanisms to our computational models, the models should allow agents to answer three questions:

1. “Do I consider the given reward distribution to be fair to me?”
2. “Am I willing to pay in order to punish a certain other agent?”
3. “Do I want to interact with a certain other agent?”

5.2 The template model

The requirements and questions presented here directly map to the three elements of our template model for computational human-inspired fairness. In Fig. 1, we provide a graphical depiction of this template model.

Here, white boxes are part of the template, and gray boxes represent elements that need to be added. From left to right, we first encounter a *utility function*.[5] Progressing one step to the right, we see that, given their utility function, agents may perform three *measurements*, i.e., from top to bottom in the figure, first, they may measure the fairness of an interaction, second, they may determine whether punishment would be desirable, and third, they may determine whether withholding action would be desirable. In the right-most column of the figure, we see the *consequences* of the three measurements. From top to bottom, once again, we observe three consequences. First, after having measured the fairness of an interaction, agents ‘know’ whether this interaction was fair to them (i.e., they may agree with the interaction), or whether they have been exploited. Second, if according to their utility function, they

[4] We acknowledge that these may not be the only, or the most important, human mechanisms; we discuss many more mechanisms in previous work [17]. However, both mechanisms have been demonstrated to be quite effective when they are applied in a computational context [24,29,44], which explains why we focus on these mechanisms.

[5] We note that in our work, a utility function maps agents’ perceivable rewards to a higher-order quantity called utility; agents learn based on their utility. In decision theory, the term ‘utility’ is generally associated with what we call ‘reward’ here.

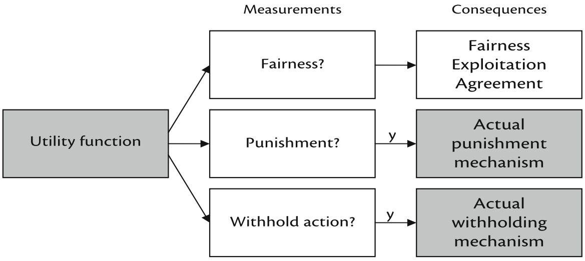

Fig. 1 The template model for computational human-inspired fairness

would benefit from punishing, we may provide agents with an actual punishment mechanism, which specifies how punishment takes place. Third and last, a similar mechanism may be introduced for withholding action.

As can be seen from the figure, computational models that are based on the template model need to provide three elements (i.e., gray boxes). Here, we discuss the white boxes in the figure, i.e., elements already present in the template itself. First, we look at measuring fairness. As in game theory, reinforcement learning [48,50], and welfare economics, our models will be centered around the concept of a utility function. We measure whether agents feel treated fairly by looking at their utility function u_i. As a threshold between a fair and an unfair interaction, we use the utility value that the agent obtains from not participating in the interaction, which we denote as u_i^0. This is defined as follows.

Definition 1 For each agent i, u_i^0 denotes the utility value that agent i experiences by refraining from participation in an interaction, while the other agents (potentially) do participate.

In the social dilemmas investigated in this paper, refraining from participation gives an agent a reward $r_i = 0$. Given the agent's utility function (and the rewards of the other agents, which we assume are known, e.g., in our social dilemma games, also 0 if one agent refrains from participation), we can calculate u_i^0.[6]

Comparing their utility value to u_i^0 helps agents to answer the question: "Do I consider the given reward distribution to be fair to me?". If agent i has $u_i \geq u_i^0$, then he *agrees* with the outcome of the interaction. If i has $u_i < u_i^0$, the agent feels *exploited* and will not agree. Given the notion of a fairness utility function and u_i^0, we may define a fair interaction.

Definition 2 An interaction between n agents is fair with an error margin of $\epsilon \in [0, 1]$ iff no more than ϵn agents are exploited, i.e., $|\{i : u_i < u_i^0\}| \leq \epsilon n$.

Clearly, the actual utility function to be used has to be selected with care, depending on the requirements of the task at hand, especially since the utility function also determines how we calculate u_i^0. A great deal of our (previous) work was concerned with actually finding and applying suitable fairness utility functions.

Once a suitable fairness utility function has been found, it may also be used to allow agents to answer the question: "Am I willing to pay in order to punish a certain other agent?". Since a fairness utility function may be based not only on individual reward, but also (how this compares to) the reward of others, it may be well possible that an agent i's utility u_i increases

[6] Obviously, should an agent decide to change its utility function (i.e., with the objective to cheat), there is little we can do to prevent this, but this is not the focus of the paper.

if i spends c to reduce another agent's reward by $e > c$. The same goes for withholding action; if i expects or measures that interacting with another agent gives it a fairness utility $u_i < u_i^0$, it is better off by not participating at all, as this gives i a utility of precisely u_i^0.

5.3 Individual learning

Our general approach, aimed at showing that our computational models allow agents to find good solutions in social dilemmas, is based on reinforcement learning [48], more specifically continuous action learning automata (CALA) [49].[7] CALA maintain a Gaussian distribution from which actions are pulled. They require feedback on *two* actions, being the action corresponding to the mean μ_i of the Gaussian distribution, and the action corresponding to a sample x_i, taken from this distribution. With n learning automata, every automaton i receives feedback with respect to the joint actions, respectively $\beta_i(\mu)$ and $\beta_i(x)$, with $\mu = (\mu_1, \ldots, \mu_n)$ and $x = (x_1, \ldots, x_n)$. In turn, this feedback is used to update the probability distribution's μ_i and σ_i. The update formula for CALA can be written as follows.

$$\begin{aligned}\mu_i &= \mu_i + \lambda \frac{\beta_i(x) - \beta_i(\mu)}{\Phi(\sigma_i)} \frac{x_i - \mu_i}{\Phi(\sigma_i)} \\ \sigma_i &= \sigma_i + \lambda \frac{\beta_i(x) - \beta_i(\mu)}{\Phi(\sigma_i)} \left[\left(\frac{x_i - \mu_i}{\Phi(\sigma_i)} \right)^2 - 1 \right] - \lambda K (\sigma_i - \sigma_L)\end{aligned} \tag{1}$$

In this equation, λ represents the learning rate; K represents a large constant driving down σ. The variance σ_i is kept above a threshold σ_L to keep calculations tractable even in case of convergence.[8] This is implemented using the function:

$$\Phi(\sigma_i) = \max(\sigma_i, \sigma_L) \tag{2}$$

Using this update formula, CALA rather quickly converge to a (local) optimum. Theoretically, they will also converge if multiple CALA are learning optimal joint actions [49]. However, we observe that we need a small modification to the update formula in this case. To explain why, we introduce an example. Assume two CALA are learning from each other in the UG (with an amount of 10 to bargain about). One has $\mu_1 = 5$ and $\sigma_1 = \sigma_L$, the other has $\mu_2 = 6$ and $\sigma_2 = 1$. Their joint actions currently are, e.g., $\mu = (5, 6)$ and $x = (5.00001, 4.5)$. In the UG, we have two possible roles for each agent (i.e., proposer and responder). For both roles, the strategy denotes the amount the agent wants to walk away with (for a motivation of this abstraction, we refer to [19]). Thus, the joint action $\mu = (5, 6)$ implies that agent 1 keeps 5 (and offers 5 to agent 2), while agent 2 wants to obtain at least 6, so it rejects an offer of 5. Similarly, the joint action $x = (5.00001, 4.5)$ implies that agent 1 wishes to keep 5.00001 and thus offers 4.99999 to agent 2, which accepts because it wants to obtain at least 4.5. Thus, from the perspective of the first automaton, its μ_1-action is not acceptable for the other automaton, yielding a reward of 0. Its x_1-action is accepted, yielding a reward of 4. If we use these values in the formula of Eq. 1, we obtain a new μ_1-value of 8×10^7 (i.e., for the

[7] Many researchers have pursued a more analytical approach and used evolutionary game theory to study attractors produced by certain mechanisms [e.g., 37,47], e.g., by means of replicator dynamics [50]. Our approach, i.e., using learning automata, has been shown to yield equivalent results in the case of a small, discrete set of strategies [30]. A similar equivalence in the continuous case is currently being investigated [51].

[8] In all experiments, we used the following settings after some initial experiments: $\lambda = 0.02$, $K = 1$ and $\sigma_L = 10^{-7}$. The precise settings for λ and σ_L are not a decisive influence on the outcomes, although other values may lead to slower convergence. The value of K is quite important; with lower values, the standard deviation never decreases sufficiently, whereas with higher values, we may get premature convergence.

first automaton). This value is far beyond the allowed boundaries. To counter this undesired adaptation, we impose the following limitation:

$$\left| \frac{\beta_i(\mu) - \beta_i(x)}{\Phi(\sigma_i)} \right| \leq 1 \tag{3}$$

This prevents the CALA from adapting too drastically in cases such as the example. Convergence is not affected [16].

6 Two computational models of human-inspired fairness

In this section, we discuss two computational models of human-inspired fairness, based on the template model presented in Sect. 5. In the section following this one, we will present a selection of experiments and results.

6.1 A computational model of inequity aversion

The first computational model discussed here is based on a model of human behavior called inequity aversion [24]. In this paper, we apply this model to the UG only. We discuss the utility function used, as well as the manner in which altruistic punishment is implemented. We note that withholding action, which is part of the template model, is not further implemented in this computational model; it is in the second computational model.

6.1.1 The utility function

Human behavior in the UG (and many other interactions), i.e., providing only non-zero offers and never accepting any overly low offers, is successfully explained by the *inequity aversion* model, as developed by Fehr et al. [17,24]. This model is centered on a utility function called *Homo Egualis* [26], based on the rewards $r = (r_1, \ldots, r_n)$ obtained by each of the n agents:

$$u_i(r) = r_i - \frac{\alpha_i}{n-1} \sum_j \max\left\{r_j - r_i, 0\right\} - \frac{\beta_i}{n-1} \sum_j \max\left\{r_i - r_j, 0\right\} \tag{4}$$

Here, $u_i(r)$ is the utility of agent $i \in \{1, 2, \ldots, n\}$. This utility is calculated based on agent i's own reward r_i and two inequity-averse terms related to considerations on how this reward compares to the rewards r_j of other agents j. Every agent i experiences a negative influence on its utility for other agents j that have a higher reward (weighed by a parameter α_i) as well as other agents that have a lower reward (weighed by a parameter β_i). The two resulting terms are subtracted from the utility of agent i.[9] Thus, given its own reward r_i, agent i obtains a maximum utility $u_i(r)$ if $\forall j : r_j = r_i$. For resource allocations that contain a great deal of inequity, agents may experience a utility below u_i^0. In that case, they are better off rejecting the proposed allocation, as this, by definition, yields a utility of u_i^0 (which is 0 in the UG).

[9] Research with human subjects provides strong evidence that humans care more about inequity when doing worse than when doing better in society [24]. 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 reward in order to reduce inequity, since simply reducing their reward (without giving it to someone else) already increases their utility value.

In our experiments, we allow any agent that experiences a utility below $u_i^0 = 0$ to reject the allocation. Thus, we use $\epsilon = 0$ here (Def. 2).

Since the inequity aversion model is already based on a utility function, it provides us with a good opportunity to build a computational model according to our template. Before we can actually do this, we need to ensure that the utility function is computationally applicable. A typical problem with the Homo Egualis utility function is that rather large areas of the function are not useful [16]. More precisely, if two agents currently have incompatible strategies in the UG, they will receive a utility of 0, regardless of how incompatible the strategies actually are. In this computational model, we address this problem by introducing a *driving force* (in the next subsection, we discuss a second computational model, which uses a different approach). We base this force on the knowledge that, if both μ_i and x_i yield a utility of 0, the *lowest* action was nonetheless the best one in the UG (after all, we get a more probable agreement if the offerer keeps less to himself, as well as if the responder accepts a lower amount). Therefore, we set

$$u_i = \max\left(u_i, \mu_i - x_i\right), \tag{5}$$

essentially driving the automaton's μ_i downward [16].

6.1.2 Implementing punishment

In our experiments, we look at UGs with only two players, but also at games with more players. In these games, every player keeps a portion of the resource at hand; the last player receives what is left, if anything. Everyone then calculates their utility, and if one of the players have a utility below u_i^0 (which is 0 in the UG), this player is better off rejecting the resource allocation (i.e., to punish), since this gives everyone a utility of u_i^0. The game is played repeatedly, allowing agents to learn an optimal strategy.

6.1.3 Experimental setup

For every experiment, we start with an amount of 100 being equally divided over the n agents (i.e., $\mu_i = 100/n$ for all agents, with $\sigma_i = 0.1\mu_i$). Agents are then allowed to play 2,000 UGs together, and after each game, their strategy is updated according to the CALA update scheme.

6.2 A computational model based on social networks

Recent research indicates that interactions between humans are typically structured in complex social networks [3]. In simulations with agents that learn by imitating their neighbors' strategies, such networks have been shown to have a decisive impact on the outcomes of social dilemma interactions [43,55], especially when agents are allowed to change the structure of the network as a result of being exploited [44]. Apart from focussing on learning by imitation instead of individual learning, existing work is typically limited to social dilemmas with a low, discrete number of strategies per agent (i.e., usually 2), one of which is then labeled 'defective' and the other 'cooperative'. In case of the PGG for example, this means we are essentially considering a multi-player Prisoners' Dilemma [2].

In our work, we look at agents that learn individually, and we do not limit agents to discrete strategies, as our resources are assumed to be continuous in nature. In this paper, we restrict ourselves to reporting our results for the PGG (arguably the more difficult of the two games under consideration here; results concerning the UG may be found in [19]). Instead of having all agents interact at the same time, we organize agents on the basis of a *scale-free network* [3]. In such a network, many nodes have connections to only few nodes, whereas only few nodes have connections to many nodes. In every learning iteration, we randomly pick one agent, and play a PGG between this agent and a random neighbor. This approach allows us to use many more agents than in the previous model (e.g., $10K$ instead of 100).

Once again, our computational model builds upon the template model. In contrast to the first model, as presented in the previous section, the second model includes all three elements of the template model, i.e., a utility function, altruistic punishment, as well as withholding action.

6.2.1 The utility function

In comparison to the utility function used in the first model, the one used here is much simpler. We already derived that, if we apply inequity aversion to the PGG, agents will punish another agent whenever this agent contributes less than them [14]. We may use this derivation as an inspiration for our utility function here (which is based on only two agents, as agents play pairwise games in their social network), i.e.,

$$u_i(r) = r_i - r_j. \tag{6}$$

Thus, if agent j contributes less than agent i (leading to a higher reward for j), this is perceived as defective and unfair. Clearly, in the absence of any agents willing to contribute anything, no agent will perceive unfair behavior, which is undesirable in the PGG (as we need contributions to make profit). We therefore introduce a small percentage of *fixed strategy* (FS) agents, which provide a good example by always performing the most contributive strategy possible, i.e., C. We investigate the influence of these agents.

6.2.2 Implementing punishment

Even if all agents are willing to punish (i.e., because $u_i < 0$), they should not always do so. The main problem is that, in a continuous strategy space, many learning algorithms (e.g., CALA) optimize by performing a great deal of local search.[10] To solve this problem, we propose the mechanism of *probabilistic punishment*, i.e., the probability that an agent i punishes an agent j should depend on the actual strategies μ_i and μ_j and the resulting rewards r_i and r_j. Punishment is more often performed for higher differences between these rewards. We may derive that the punishment probability should be set to

$$p_i^p > -\frac{1}{e}(1 - 0.5r)u_i, \tag{7}$$

with r denoting the factor with which the common pool is multiplied, and e denoting the effect of punishment on the reward of the agent being punished [14], as explained in Sect. 3. We may also derive that, as long as $e > (1 - 0.5r)\,C$, we may set $p_i^p = -\frac{1}{C}u_i$. In our case,

[10] Imagine agent j with $\mu_j = 2$, playing against agent i with $\mu_i = 8$. Agent j may want to try $\mu_j = 3$. If agent i punishes j in *both* cases, the essential idea underlying punishment, i.e., a reversal of the inverse relation between contribution and reward, fails to work: $\mu_j = 2$ gives a higher reward than $\mu_j = 3$, because both strategies are punished.

we use $C = 10$, $r = 1.5$, and $e = 3$, so we may indeed set p_i^p to the value indicated.[11] We note that probabilistic punishment in this second model fulfills the same role as the driving force in the first model, i.e., avoiding useless feedback to the CALA.

6.2.3 Implementing withholding action

The mechanism of *rewiring* in the agents' social network provides an opportunity for agents to refrain from interacting with a certain other agent again, i.e., essentially, to withhold action. In order to avoid playing with a certain undesirable neighbor j, agent i may decide to break the connection between him and j and create a link to a random neighbor of j [44].[12] For rewiring, we use the assumption, taken from the template model, that agents want to disconnect themselves from (relative) defectors, as these give them a negative utility (i.e., $u_i < u_i^0$). The probability that agent i unwires from agent j, is calculated as:

$$p_i^r = -\frac{1}{C} u_i. \tag{8}$$

Even if agents determine that they want to unwire because of this probability, they may still not be allowed to, if this breaks the last link for one of them. If unwiring does take place, agent i creates a link to a random neighbor of j.

In our experiments, we use withholding action only in combination with punishment. This is in contrast with related work [44], which reports that rewiring works well without the need for punishment. However, as mentioned before, this related work uses an approach based on imitation, where agents imitate the strategies of their neighbors based on accumulated rewards, rather than an approach based on individual learning. If cooperators can get rid of defectors (by unwiring), the cooperators can obtain a very good payoff, whereas defectors cannot, as there is no-one to exploit. Given learning by imitation, defectors will therefore learn to cooperate. In contrast, using individual learning, withholding action by itself will not give any results, as agents do not receive an explicit signal that they are overly defective. The point of the withholding mechanism is to enhance the effects of punishment. Essentially, as also observed by Santos et al. [44], rewiring leads to a network in which (relative) cooperators are strongly connected, mostly to other relative cooperators, whereas relative defectors lose all but a few (or even one) links. This means that the majority of games will be played between cooperators, strengthening their tendency to cooperate. A minority of games will be played between a cooperator and a defector, where the cooperator is strong and the defector weak (loosely connected). Thus, because of rewiring, in almost every game the defector participates in, he will be punished, and will therefore learn to cooperate.

6.2.4 Experimental setup

In our experiments, we use three types of agents. In addition to the fixed-strategy (FS) agents discussed above (i.e., agents that always contribute 10 and punish those who contribute less), we introduce dynamic-strategy (DS) agents with two different initial settings. DSr agents are

[11] For instance, with the settings above, given $\mu_i = 5$, and $\mu_j = 3$, we obtain rewards of $r_i = \frac{1}{2} \cdot 1.5 \cdot (5 + 3) - 5 = 1$ and (similarly) $r_j = 3$, making the utilities $u_i = 1 - 3 = -2$ and $u_j = 2$. Thus, the punishment probabilities are $p_i^p = -\frac{1}{10} \cdot -2 = 0.2$ and $p_j^p = -0.2$.

[12] Note that we may also choose to allow an agent i to create a new connection to specific other agents instead of only random neighbors of their neighbor j. However, this allows (relative) defectors to identify (relative) cooperators quickly, with which they may then connect themselves in an attempt to exploit. Preliminary experiments have shown that this behavior may seriously impair the emergence of agreement and cooperation.

initially individually rational; they do not contribute at all. DSh agents are human-inspired in the sense that they are initially rather cooperative; they contribute $X \sim N(7, 1)$. In our experiments, we run 6,000n games, with n the number of agents. Simulations are repeated 50 times. Every experiment determines the effect of rewiring by comparing a population that may not rewire, with a population that may. We analyze outcomes by drawing box plots of four interesting quantities, i.e., (1) the number of games every agent needs to play (on average) in order to converge to a certain strategy, (2) the average strategy the population converges to (remember that a rational population would converge to a strategy of 0), (3) the performance of the population, i.e., the percentage of pairwise games in which both agents have a similar converged strategy (this is ϵ in Definition 2), and finally, (4) the structure of the network of interaction, which may have changed due to rewiring.[13]

7 Experiments and results

Although both models discussed above may be successfully applied to both social dilemma games discussed in the paper, we show results for only one game (a different game per model). More results may be found in [14,16,17,19].

7.1 Inequity aversion in the Ultimatum Game

7.1.1 Results

We give a small selection of results here. A more extensive overview of results, for up to 100 agents, is given in earlier work [16].

1. In the first experiment discussed here, we use two agents and set $\alpha = 0.6$, $\beta = 0.7$ for both agents' Homo Egualis utility function. Analysis [24] reveals that the first agents should then continue to offer 50 of the resource at hand to the second agent, which should be accepted. This indeed always happens.
2. In the second experiment discussed here, we again use two agents and set $\alpha = 0.6$, $\beta = 0.3$ for both agents. Results of one particular run are illustrated in Fig. 2, where we see that the second agent punishes whenever its utility is below 0, and thus obtains 27 out of 100 instead of 0, as predicted [24].
3. In the third experiment discussed here, we use three agents and set $\alpha = 0.6$, $\beta = 0.3$ for all three. Analysis [13,16] reveals that all agents should obtain at least 15.8. A typical learning curve is given in Fig. 3. We see that the first two agents learn to reduce the third agent's reward to 15.8, which gives this agent a utility of nearly 0. Whenever the first two agents reduce the third agent's reward to an amount below 15.8, the third agent refuses the proposed allocation (we see five such moments in the utility graph of Fig. 3).

7.1.2 Discussion

Instead of the rational allocations obtained by individually rational agents (i.e., keeping everything to themselves), or the possibly overly altruistic allocations obtained by applying certain social welfare functions (e.g., with egalitarian social welfare, only a 50-50 split would

[13] Details about the four quantities, and how they are derived, may be found in [19], in which we follow a similar approach for the UG instead of the PGG.

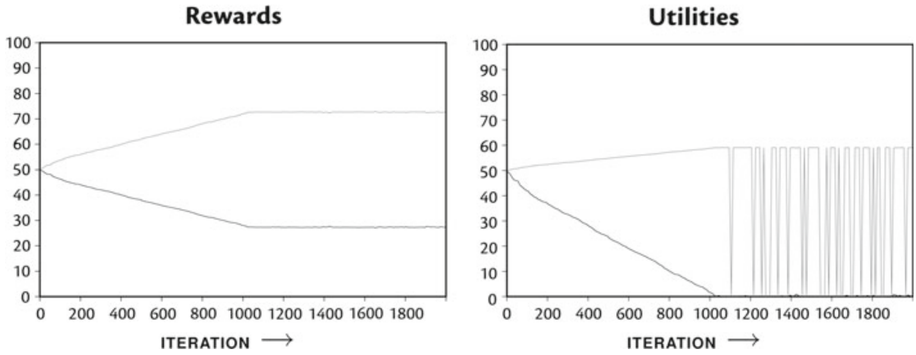

Fig. 2 Learning to play the 2-agent Ultimatum Game with inequity aversion

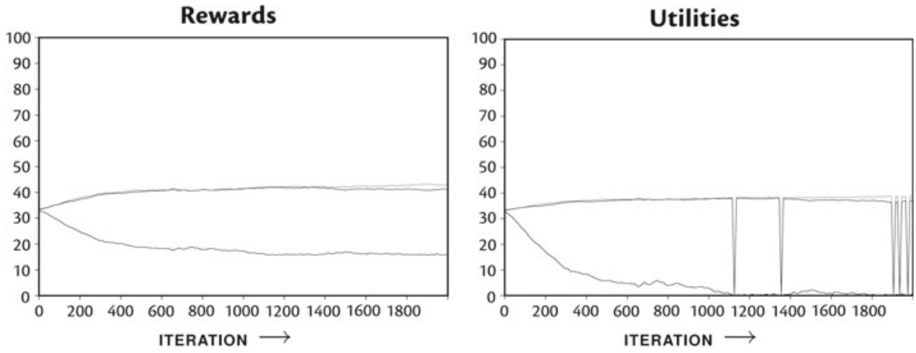

Fig. 3 Learning to play the 3-agent Ultimatum Game with inequity aversion

be optimal), an approach based on the inequity aversion model and (altruistic) punishment allows agents to find allocations that are more similar to human allocations. Thus, when playing UGs against humans, these agents are able to maximize their profit (provided we estimate the values for α and β these humans have, which is possible [24,12]), without needing to be overly fair. Essentially, inequity-averse agents are just fair enough.

7.2 Social networks in the Public Goods Game

Our experiments focus on varying two parameters. First, we investigate the influence of the population size on the outcome, using a fixed proportion of 33% FS agents, 33% DSr agents and 33% DSh agents. Second, we investigate the influence of the FS agents on the outcome, by varying the proportion of FS agents, while keeping the other two types in equal proportion. We present results for up to 500 agents playing the PGG. In [19], we present results for up to 10, 000 agents playing the UG. Such results are omitted here, since they would be highly similar to results for smaller populations, while taking much longer to simulate.

7.2.1 Results: population size

Experiments were performed with populations of 50, 100, 200, and 500 agents. Results are shown in Fig. 4. We compare a setup with a static population structure (top) to a setup with a

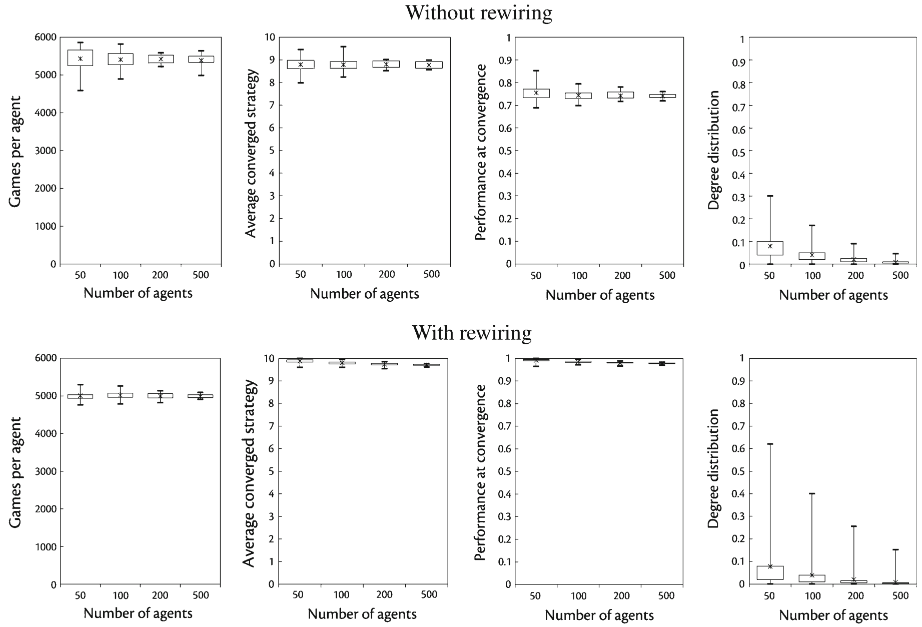

Fig. 4 Influence of population size on learning in the Public Goods Game

dynamic structure (bottom). Concerning the *number of games per agent* until convergence, we see that introducing the option to rewire allows agents to reach convergence with approximately 10% less games, e.g., 5,000 instead of 5,500. This number is not strongly influenced by the size of the population. It is interesting to note that earlier work [e.g., 43,44] reports a significantly higher number of games required per agent (e.g., 10^5), even though our learning task is more difficult. Concerning the *average learned strategy*, we observe a good result in a static population structure (i.e., around 9, where 10 would be desired), and an even better result in a dynamic structure. The *performance* shows similar characteristics; without rewiring, there is still quite some disagreement on strategies, as only around 75% of the neighbors have a common strategy, whereas with rewiring, there is hardly any disagreement. Looking at the *network structure*, we may observe that rewiring increases the preferential attachment already present in the scale-free network; very few agents become rather densely connected (e.g., with 50 agents, we observe one agent being connected to more than 30 others), whereas most agents stay sparsely connected or even lose some connections. The performance increase caused by rewiring is even more surprising if we study the number of times agents actually rewired; this turns out to be quite low (say, 1,000, i.e., it happens after less than 1% of the games played). In general, we may state that the size of the population is not a strong influence on the quantities of interest. In all cases, a static population reaches a performance of around 75%, whereas a population that may use rewiring achieves nearly 100%.

7.3 Results: percentage of fixed-strategy agents

Once again, we perform experiments in populations of 50, 100, 200, and 500 agents. We vary the proportion of FS agents between 0 and 100%, by adapting the probability that a

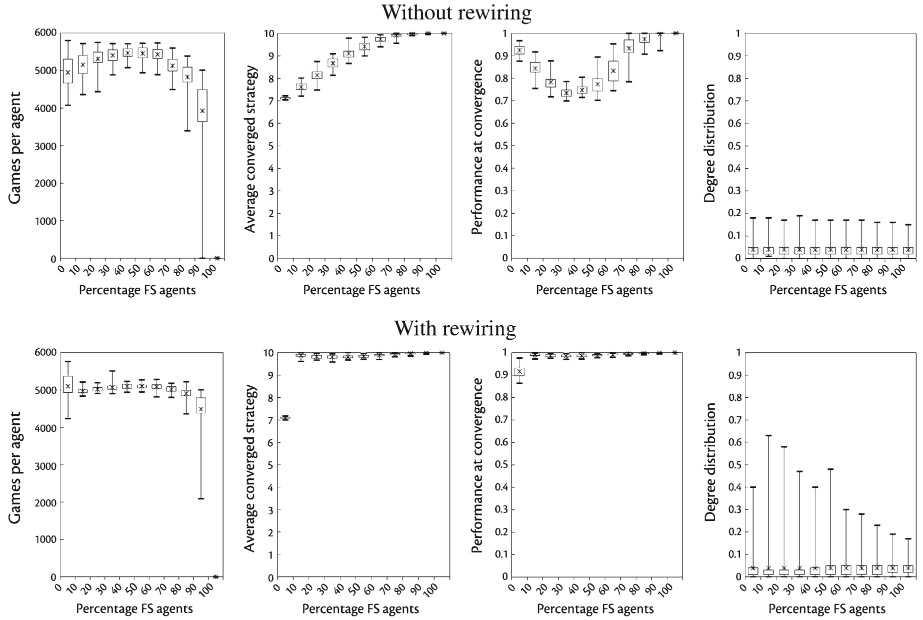

Fig. 5 Influence of percentage FS agents on learning in the Public Goods Game

newly generated agent is an FS agent. This implies that the actual number of FS agents in the population varies over individual experiments of 6,000 learning iterations. For the sake of brevity, we only report results of our experiments with a population of 100 agents. The results for other population sizes are highly similar to those reported here. The results for 100 agents are shown in Fig. 5. We compare a setup with a static population structure (top) to a setup with a dynamic structure (bottom).

When we look at the *number of games per agent* until convergence is reached, we see that populations with a low or high percentage of FS agents converge slightly more quickly than populations with a percentage in between, e.g., 40%. Introducing rewiring reduces the number of games needed by approximately 10%. When we look at the *average converged strategy*, we may observe that (1) a population without any FS agents converges to the most cooperative strategy present, i.e., the 7 of the DSh agents; (2) adding FS agents to the population allows the DS agents to learn an even more cooperative strategy, with much better results for a population that is allowed to rewire. Once again, the *performance* measures reflect the quality of the average learned strategy. Interestingly, with an increasing proportion of FS agents, the performance of a static population initially decreases. If we look at the average learned strategy, we can see why this is; the averages reported reflect that the DS agents learn to contribute 7 at least until there are around 40% of FS agents. With more FS agents, the DS agents learn to contribute more than 7, making them more compatible with the FS agents. When we allow agents to rewire, a low percentage of FS agents is already sufficient to achieve full cooperation. Finally, looking at the *network structure* resulting from rewiring, as compared to the static network, we see an interesting phenomenon: with a low percentage of FS agents, a few agents show a drastic increase in their number of connections to other agents. For instance, with 10% FS agents, there is a single agent that is connected to 63 of the 100 agents. Clearly, if this single agent is an FS agent, connecting to it is useful, as it allows a DS agent to quickly learn the desired strategy. Once again, the number of times agents actually rewire, is low.

7.3.1 Summary of results

The population of agents is (almost) insensitive to an increase in population size. Allowing agents to rewire (i.e., restructure the network) significantly enhances their abilities to find desirable, cooperative solutions. One of the most important results, in our view, is that a population without any FS agents at all converges to the most cooperative strategy initially present (as long as this strategy is played by a sufficient number of agents). This result confirms results found earlier [44] by making agents learn by imitation in social dilemmas with a discrete strategy set, and extends these results to agents learning individually in a continuous strategy space.

7.3.2 Discussion

In this subsection, we demonstrated how the human-inspired mechanisms of punishment and withholding action allow agents to reach satisfactory outcomes in the PGG, without running into scalability or complexity issues, and even in the presence of agents that are (initially) not willing to care for fairness. With an approach that is not based on punishment and/or withholding action, achieving a similarly satisfactory outcome would require either complex agents (e.g., agents that negotiate or use norms), or a centralized approach (e.g., a single agent determines what is best for all agents, and everyone complies to this).

8 Conclusion

In this paper, we argue that many applications of multi-agent systems may benefit from the inclusion of human-inspired fairness. Typically, agents in multi-agent systems are designed according to the principles of classical game theory, i.e., agents are assumed to have full information, and to act in an individually rational manner based on this information [26]. Such assumptions are not realistic and not desirable in many cases. Recent publications [9,20,26,53] stress the importance of addressing this problem. Researchers propose to include the concept of social welfare, as known from welfare economics, in multi-agent systems. Although including social welfare indeed enhances agents' abilities to find allocations of resources that balance individual rewards and fairness, we argue that we may further improve these abilities by including mechanisms inspired by humans.

Humans show remarkable ability when they are confronted with a class of problems called social dilemmas, in which there is a conflict between optimizing individual reward and optimizing the overall reward of the collective [26]. As a result, in social dilemmas, fairness is important—individuals need to find a strategy that provides a fair balance between individual reward and collective reward. In some social dilemmas, the fairest action is to be cooperative (e.g., to invest all one's money in the Public Goods Game), and in other social dilemmas, what is fair depends on the actions of the opponent (e.g., to match an opponent's request in the Ultimatum Game). Among the (numerous) mechanisms humans apply in order to establish desirable outcomes, are altruistic punishment and withholding action. These mechanisms are successfully applied by humans even though doing so is not individually rational; essentially, the decision whether or not to execute these mechanisms entails a second-order social dilemma.

Tasks containing social dilemmas are also prominently present in many applications of multi-agent systems, most notably because resource allocation is a common task for these systems [9,20]. We discuss in this paper how agents may be motivated to follow the human example, i.e., to enforce desirable solutions by altruistic punishment or by withholding action. To this end, we first provide the foundations of human-inspired fairness, i.e., requirements for models based on human-inspired fairness, a template model, and a general approach based on the individual learning of a utility function. The algorithm with which the agents are learning (i.e., CALA), has been shown to converge to equilibrium points [49]. Thus, if a group of learning agents, driven by this algorithm and proposed computational models, learns a certain desired strategy, we may conclude that the computational models facilitate the establishment of this desired strategy.

Exemplifying the usability of the template model and the learning approach, we then present two different computational models and apply them to two prominent social dilemmas, i.e., the Ultimatum Game (UG) and the Public Goods Game (PGG). In both games, we do not restrict ourselves to only a discrete set of strategies, which is often done in the literature. We use individually learning agents, following our general approach, to demonstrate the efficacy of our computational models. We show that both models allow agents to learn desired solutions in the two social dilemma games under study, which is a novel and interesting result, for two reasons. First and foremost, existing work generally does not manage to establish decentralized mechanisms that allow agents to find and maintain desired solutions to social dilemmas [especially the PGG; see, e.g., 29,54]. Second, existing work generally reports requiring a great deal more learning iterations (i.e., repeated games) in order to establish convergence, even in dilemmas with only a discrete set of strategies [e.g., 43,44].

In conclusion, this paper argues that human-inspired fairness models may need to be integrated in many multi-agent systems. We proposed a manner in which this task may be executed. The resulting multi-agent systems are shown to find solutions to difficult problems that thus far were difficult to address.

Acknowledgments

We thank the anonymous reviewers for their constructive criticism.

References

1. Aldewereld, H. (2007). *Autonomy vs. conformity: An institutional perspective on norms and protocols*. PhD thesis, Universiteit Utrecht.
2. Axelrod, R. (1984). *The evolution of cooperation*. New York: Basic Books.
3. Barabasi, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*, 509–512.
4. Basu, K. (1994). The traveler's dilemma: Paradoxes of rationality in game theory. *American Economic Review, 84*(2), 391–395.
5. Bearden, J. N. (2001). Ultimatum bargaining experiments: The state of the art. *SSRN eLibrary*.
6. Binmore, K. G. (1991). *Fun and games: A text on game theory*. Lexington: D.C. Heath.
7. Bourke, T. (2001). *Server load balancing*. Sebastopol: O'Reilly Media Inc.
8. Boyd, R., Gintis, H., Bowles, S., & Richerson, P. J. (2003). The evolution of altruistic punishment. *Proceedings of the National Academy of Science USA, 100*, 3531–3535.
9. Chevaleyre, Y., Dunne, P., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodriguez-Aguilar, J., & Sousa, P. (2006). Issues in multiagent resource allocation. *Informatica, 30*, 3–31.
10. Chevaleyre, Y., Endriss, U., Lang, J., & Maudet, N. (2007). 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), Vol. 4362 of LNCS* (pp. 51–69). Berlin: Springer.

11. Dall'Asta, L., Baronchelli, A., Barrat, A., & Loreto, V. (2006). Agreement dynamics on small-world networks. *Europhysics Letters, 73*(6), 969–975.
12. Dannenberg, A., Riechmann, T., Sturm, B., & Vogt, C. (2007). Inequity aversion and individual behavior in public good games: An experimental investigation. *SSRN eLibrary*.
13. de Jong, S. (2009). *Fairness in multi-agent systems*. PhD thesis, Maastricht University.
14. de Jong, S., & Tuyls, K. (2008). *Learning to cooperate in public-goods interactions 2008*. Presented at the EUMAS'08 Workshop, Bath, UK, December 18–19.
15. de Jong, S., & Tuyls, K. (2009). Learning to cooperate in a continuous tragedy of the commons. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2009)* (pp. 1185–1186).
16. de Jong, S., Tuyls, K., & Verbeeck, K. (2008a). Artificial agents learning human fairness. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'08)* (pp. 863–870).
17. de Jong, S., Tuyls, K., & Verbeeck, K. (2008b). Fairness in multi-agent systems. *Knowledge Engineering Review, 23*(2), 153–180.
18. de Jong, S., Tuyls, K., Verbeeck, K., & Roos, N. (2008). Priority awareness: Towards a computational model of human fairness for multi-agent systems. *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning, 4865*, 117–128.
19. de Jong, S., Uyttendaele, S., & Tuyls, K. (2008). Learning to reach agreement in a continuous ultimatum game. *Journal of Artificial Intelligence Research, 33*, 551–574.
20. Endriss, U. (2008). Fair division. *Tutorial at the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).*
21. Endriss, U., Maudet, N., Sadri, F., & Toni, F. (2003). On optimal outcomes of negotiations over resources. In: *AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 177–184). New York: ACM. ISBN 1-58113-683-8. doi:10.1145/860575.860604.
22. Endriss, U., Maudet, N., Sadri, F., & Toni, F. (2006). Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research, 25*, 315–348.
23. Fehr, E., & Gaechter, S. (2002). Altruistic punishment in humans. *Nature, 415*, 137–140.
24. Fehr, E., & Schmidt, K. (1999). A theory of fairness, competition and cooperation. *Quarterly Journal of Economics, 114*, 817–868.
25. Gerding, E., van Bragt, D., & Poutré, J. L. (2003). Multi-issue negotiation processes by evolutionary simulation: Validation and social extensions. *Computational Economics, 22*, 39–63.
26. Gintis, H. (2001). *Game theory evolving: A problem-centered introduction to modeling strategic interaction*. Princeton: Princeton University Press.
27. Gueth, W., Schmittberger, R., & Schwarze, B. (1982). An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior and Organization, 3*(4), 367–388.
28. Hardin, G. (1968). The tragedy of the commons. *Science, 162*, 1243–1248.
29. Hauert, C., Monte, S. D., Hofbauer, J., & Sigmund, K. (2002). Volunteering as red queen mechanism for cooperation in public goods games. *Science, 296*, 1129–1132.
30. Hennes, D. (2008). *Multi-agent learning in stochastic games—Piecewise and state-coupled replicator dynamics*. Master's thesis, Universiteit Maastricht.
31. Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., & Gintis, H. (2004). *Foundations of human sociality: Economic experiments and ethnographic evidence from fifteen small-scale societies*. Oxford: Oxford University Press.
32. Kalagnanam, J., & Parkes, D. C. (2004). Auctions, bidding and exchange design. In D. Simchi-Levi, S. D. Wu, & M. Shen (Eds.), *Handbook of quantitative supply chain analysis: Modeling in the e-business era, Int. Series in operations research and management science, Chapter 5* (pp. 1–84). Dordrecht: Kluwer.
33. Kollock, P. (1998). Social dilemmas: The anatomy of cooperation. *Annual Review of Sociology, 24*, 183–214.
34. Larrick, R., & Blount, S. (1997). The claiming effect: Why players are more generous in social dilemmas than in ultimatum games. *Journal of Personality and Social Psychology, 72*(4), 810–825.
35. Messick, D. M., & Brewer, M. B. (1983). Solving social dilemmas: A review. *Review of Personality and Social Psychology, 4*, 11–44.
36. Milinski, M., Semmann, D., & Krambeck, H. J. (2002). Reputation helps solve the tragedy of the commons. *Nature, 415*, 424–426.
37. Nowak, M. A., Page, K. M., & Sigmund, K. (2000). Fairness versus reason in the ultimatum game. *Science, 289*, 1773–1775.

38. Oosterbeek, H., Sloof, R., & van de Kuilen, G. (2004). Cultural differences in ultimatum game experiments: Evidence from a meta-analysis. *Experimental Economics, 7*, 171–188.
39. Panchanathan, K., & Boyd, R. (2004). Indirect reciprocity can stabilize cooperation without the second-order free rider problem. *Nature, 432*, 499–502.
40. Rockenbach, B., & Milinski, M. (2006). The efficient interaction of indirect reciprocity and costly punishment. *Nature, 444*(7120), 718–723. ISSN 0028-0836.
41. Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach (2nd ed.)*. Englewood Cliffs: Prentice-Hall.
42. Sandholm, T. (2006). Optimal winner determination algorithms. In P. Cramton, Y. Shoham, & R. Steinberg (Eds.), *Combinatorial auctions, Chapter 14*. MIT Press.
43. Santos, F. C., & Pacheco, J. M. (2005). Scale-free networks provide a unifying framework for the emergence of cooperation. *Physical Review Letters, 95*, 98–104.
44. Santos, F. C., Pacheco, J. M., & Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS Computational Biology, 2*(10), 1284–1291.
45. Sen, A. K. (1970). *Collective choice and social welfare*. San Francisco: Holden Day.
46. Shoham, Y., Powers, R., & Grenager, T. (2007). If multi-agent learning is the answer, what is the question? *Artificial Intelligence, 171*(7), 365–377.
47. Sigmund, K., Hauert, C., & Nowak, M. A. (2001). Reward and punishment. *Proceedings of the National Academy of Sciences, 98*(19), 10757–10762.
48. Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press. A Bradford Book.
49. Thathachar, M. A. L., & Sastry, P. S. (2004). *Networks of learning automata: Techniques for online stochastic optimization*. Dordrecht: Kluwer.
50. Tuyls, K., & Nowé, A. (2005). Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review, 20*, 63–90.
51. Tuyls, K., & Westra, R. (2009). Replicator dynamics in discrete and continuous strategy spaces. In *Accepted in multi-agent systems: Simulation and applications* (accepted).
52. Uyttendaele, S. (2008). *Fairness and agreement in complex networks*. Master's thesis, MICC, Maastricht University.
53. Verbeeck, K., Nowé, A., Parent, J., & Tuyls, K. (2007). Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Journal of Autonomous Agents and Multi-Agent Systems, 14*, 239–269.
54. Yamagishi, T. (1986). The provision of a sanctioning system as a public good. *Journal of Personality and Social Psychology, 51*(1), 110–116.
55. Zimmermann, M. G., & Eguíluz, V. M. (2005). Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma with adaptive local interactions. *Physical Review E, 72*(5), 056118. doi:10.1103/PhysRevE.72.056118.