

**PUNISHMENT MECHANISMS AND THEIR EFFECT ON
COOPERATION - A SIMULATION STUDY**

Mike Daniel Farjam

Master thesis Artificial Intelligence

Radboud University Nijmegen

15.06.2013

Supervisors:

M. Faillo, University of Trento, Italy

W.F.G. Haselager, Radboud University Nijmegen, The Netherlands

I.G. Sprinkhuizen-Kuyper, Radboud University Nijmegen, The Netherlands

Acknowledgements

The research presented in this thesis has been done at the Radboud Universiteit Nijmegen (The Netherlands) and the Università degli studi di Trento (Italy). Thanks to a scholarship of the Erasmus Mundus program I was able to study at both universities and join a 1-month Italian intensive course in Venice. I learned to value Europe, its cultures and the differences between them during that time.

I want to thank Marco (Faillo) for his openness towards my background in A.I. and the methodology of agent-based modeling. He provided me with literature and the necessary background knowledge needed to understand Public Goods Games and some broader concepts in Economics. In fact he made me so interested in this field that I will begin a PhD in Economics this year.

Pim (Haselager) and Ida (Sprinkhuizen-Kuyper) have been helping me for more than 3 years to do my research on resource sharing. Except from the things I learned in their classes, I could come to their office whenever I wanted, discuss my ideas and they helped me making my first steps in the scientific world. Pim, thanks to his background in philosophy, taught me a lot about good writing and structuring arguments. Ida, thanks to her background in the exact sciences, showed me where I was not precise enough and drew my attention to the details that I – out of enthusiasm – too often forget. Although I am German, I highly value their informal way of supervision.

I do not take the freedom I had (during actually my whole education) for granted and I realize that this meant extra work for my supervisors and educational staff in Nijmegen. For me, the freedom and support I got from the Radboud University is (above all others) its biggest quality.

Abstract

Punishment Mechanisms and their Effect on Cooperation - A Simulation Study

In social dilemmas punishment costs resources, not just from the one who is punished but often also from the punisher and society. Reciprocity on the other side is known to lead to cooperation without the costs of punishment. The question at hand is whether punishment besides its costs brings advantages and how its negative side-effects can be reduced to a minimum in an environment populated by reciprocal agents. Various punishment mechanisms have been studied in the economic literature such as unrestricted punishment, legitimate punishment, cooperative punishment, and the hired gun mechanism. All these mechanisms are implemented in a simulation where agents can share resources and may decide to punish other agents when they do not share. Through evolutionary learning agents adapt their sharing/punishing policy. Despite the costs of punishment, legitimate punishment compared to no-punishment increased performance when the availability of resources was low. When the availability was high, performance was better in no-punishment conditions with indirect reciprocity. Furthermore the hired gun mechanism worked only as good as other punishment mechanisms when the availability of resources was high. Legitimate punishment leads to a higher performance than unrestricted punishment. Summarized, this paper shows that punishment – given the right environment – can play a facilitating role for cooperation even if the cooperating system already adopted reciprocity.

Table of Contents

Preface	5
Journal paper (submitted to JASSS)	
Introduction	6
Punishment mechanisms	7
The model.....	9
Results	11
Discussion	15
Additional content	
Different agents	17
Additional experimental variables	18
Extra results of the final simulation	19
References	21
Appendix	23

1. Preface

The main part of this thesis (sections 2-6) was submitted to the Journal of Artificial Societies and Social Simulation (JASSS) on 20.06.2013. As the readership of this journal is familiar with the methodology of agent-based modeling (ABM) section 2 does not contain any introduction to ABM. I expect that the average reader of my thesis is not familiar with this methodology, as ABM is a fairly new approach of modeling and not (yet) widely used. The following paragraphs give a very short overview of ABM.

ABM is an approach that is often used when the object of study is a large system. Many systems – such as ecosystems, economies, societies – are the emerging product of individuals (the agents) that interact with each other. I will use “the economy” as an example to show how ABM differs in its way of modeling from other methodologies.

When we read in the news about “the economy” we hear something about goods that are produced, unemployment, consume, prices, etc. and everything is somehow related to “the economy”. From an economic point of view the economy is a large system with many actors (economic agents) interacting and depending on each other. “The economy” is thus an emerging product of the interactions of individuals. The dominant models to predict what will happen in the economy are of two kinds (Farmer and Foley 2009): Either they are statistical models, built to fit data from the past, or these models are built on agents interacting in a perfect world and taking rational choices (Dynamic stochastic general equilibrium, DSGE). Both models were not able to predict the financial crisis since 2007/2008 and are heavily criticized since then.

In ABM the individual/agent is the central unit that is modeled. Every pattern that we can observe in the larger system (e.g. the economy) is the result of the interaction of the agents. In classical economical models the macroeconomic phenomena like growth and prices are modeled directly on a macro level. In ABM they are the product of local interaction of agents on the micro level in ABM and emerge in a bottom-up way. ABMs make four assumptions about agents (Macy and Willer 2002):

1. Agents autonomously decide what they do.
2. What agents can and will do depends on what other agents do.
3. Agents’ decisions are based on simple rules.
4. Agents are adaptive.

Mae and Mac (2010) and Farmer and Foley (2009) argue that ABMs are better in predicting economic activity and crisis in the future since they are – opposite to DSGE models – not assuming a perfect world and rationality. Data from previous years, which statistical models try to fit, may not contain the information needed to predict future events. Furthermore, statistical models per definition leave out individuals, while individuals and their interaction lie at the heart of ABM and what the economy is all about: interaction and individuals.

Besides economics ABM has shown to be a very broadly applicable method in diverse fields such as e.g. sociology (Squazzoni 2012), biology (Wooldridge and Jennings 1995), chemistry (Troisi, Wong and Ratner 2004) and occupational psychology (Hughes et al. 2012). For Artificial Intelligence (AI) ABM can be a bridge to transfer theoretical knowledge from

the social sciences, via formalization and implementation in an ABM, to artificial intelligences that have to solve problems autonomously. At the end of section 6 I will give an example for this.

2. Introduction

From an evolutionary point of view cooperation is a double-edged sword. On the one hand, it can bring an evolutionary advantage to a group, since some tasks can only be achieved through cooperation or at least in a more efficient way through the contribution of the group. On the other hand, not investing in cooperation, but enjoying its resources seems to be the most efficient choice from an egoistic point of view. This is the typical free-rider problem that characterizes social dilemmas. Since selection in evolution takes place on the level of the individual, cooperators should be replaced by free-riders, putting cooperation to an end. So why is there cooperation?

It is trivial to see that the classic mechanism of evolution does not directly select for generosity. However, indirect mechanisms were proposed how cooperating leads not just to an advantage for the group but also to the individual. Henrich (2004) catches this in a formula in which the chance for altruistic behavior positively correlates with the chance of others being cooperative. Nowak (2006) lists five rules under which cooperation can evolve. Each of these rules itself is sufficient to lead to cooperation in an evolutionary system. Only humans seem to have used all five during evolution. One of these is ‘indirect reciprocity’ (IR). IR – more specifically ‘downstream reciprocity’ (Nowak and Roch 2007) – implies that by cooperating an individual increases the chance that someone else will cooperate with it. IR will lead to cooperation if the chance of knowing how often someone has shared before is bigger than the ratio of ‘costs of sharing’/‘benefit of sharing’.

Social dilemmas have been widely studied by social psychologists (Dawes and Messick 2000; Messick and Brewer 1983), game theorists (Rapoport and Chammah 1965; Axelrod 1984), and political scientists (Ostrom 1990). Experimental economists have studied the emergence of cooperation with special reference to the problem of market failures in the provision of public goods (Ledyard 1995; Chaudhuri 2011). In recent years behavioral economists have explained cooperation in social dilemmas using the concepts of peer punishment and reciprocity (Fehr and Gächter 2000; Fehr and Gächter 2002). Fehr and Gächter let human subjects play a Public Goods Game where cooperation leads to a high payoff for all if everyone cooperates, but the single player earns the maximum payoff when all the others cooperate and he/she free-rides. Free-riding is hence the unique Nash equilibrium of the game. Fehr and Gächter found that cooperation decreased during the game, but when players had the possibility to punish free-riders cooperation stabilized.

In classical Public Goods Games strangers play with each other, knowing nothing about the others’ history. Under such conditions reciprocity mechanisms (like IR) cannot work, but punishment proved to be effective in keeping cooperation going.

We hypothesize that punishment is more than just a stopgap to achieve cooperation, and that it can play a facilitating role for maximizing the efficiency of cooperation, even when it is not explicitly needed to keep it going. To test this hypothesis we implement an agent-

based simulation in which we compare agent systems that can punish and use IR, with agent systems that only use IR. In all simulations IR will be sufficient for cooperation.

The study of punishment mechanisms and reciprocity through simulation is not new. Among the most recent contributions, Jaffe and Zaballa (2010) compare a specific punishment mechanism (cooperative punishment) with the punishment mechanism used by Fehr and Gächter and find that their mechanism works better. However, it is not clear whether the increase in performance is stable with different settings of parameters in the simulation and how their mechanism performs compared to more sophisticated mechanisms proposed since Fehr and Gächter. Ye et al. (2011) found that if a group has the possibility to show appreciation for altruistic behavior, appreciation and altruistic behavior will become dominant in the group leading to cooperation. To the best of our knowledge our study is the first comparing many punishment mechanisms in one simulation. It is also the first analyzing the extra advantage punishment can give to a system of agents that are already sharing through the mechanism of reciprocity.

Humans seem to have a feeling for when punishment is necessary. Obviously it is important to not punish too often, as punishment comes with a cost and accepting a little bit of free-riding may be acceptable, but if one does not punish enough free-riding becomes dominant. In our simulation agents will have their tendency to free-ride (not cooperate) and their tendency to punish encoded in their genes. As in other Public Goods Games agents try to maximize their own earnings. Through mutation and selection agents learn when free-riding is for their own good and when punishment is necessary for the public good.

The remaining part of the paper is organized as follows: in section 2 we present four different punishment mechanisms: Unrestricted punishment, legitimate punishment, cooperative punishment, and the hired gun mechanism. In section 3 we describe the implementation of the simulation and all punishment mechanisms. The results are presented in section 4. Section 5 concludes.

3. Punishment mechanisms

Fehr and Gächter (2000; 2002) studied a form of peer punishment that is often defined as Unrestricted Punishment (UP). In their setting everyone can punish everyone else. This means that it is also possible for free-riders to punish cooperators. This phenomenon is called antisocial punishment and it is undesirable when punishment is meant to be a means to enforce cooperation (Herrmann, Thöni and Gächter 2010). Faillo, Grieco and Zarri (2013) propose a different punishment mechanism that they call Legitimate Punishment (LP). In LP an agent can only punish another agent if they are better cooperators than the agent they want to punish. This mechanism prevents antisocial punishment. Faillo et al. found that LP compared to UP saves resources to the group ensuring higher levels of cooperation among human players of a Public Goods Game.

The Hired Gun Mechanism (HGM), as proposed by Andreoni and Gee (2011), restricts the possibility to punish to prevent antisocial punishment. In contrast to LP in HGM punishment is not carried out by peers, but by an external agent - the 'hired gun' - who is in charge of punishing low contributors. In particular in Andreoni and Gee the hired gun always

punished the agent that has contributed the least. Hence agents have an incentive to provide at least the second lowest level of contribution.

The final mechanism that we consider is based on an agent based simulation from Jaffe and Zaballa (2010), called Co-operative Punishment (CP). In CP no restrictions are made on who may punish who, instead the costs of punishing are not paid by the individual that punishes but by the entire group. Punishment becomes thereby a less altruistic action. Jaffe and Zaballa found that CP was a much stronger stabilizer of cooperation than altruistic punishment.

We go one step further than CP in our simulation. In UP, LP, and HGM punishment implies a cost both for the punisher and for the punished. The resources subtracted to the punished thus vanish. In many social situations this is not true. If we get a traffic ticket, we pay this amount to the government. We do not burn the money. Nevertheless punishment is still costly, as the society has to pay the police officer. In the punishment mechanism that we call ‘Zero Loss Punishment’ (ZLP) the costs for punishment the punisher has to pay are paid by all agents, as in CP, and the cost for the punished will be reallocated to all agents. From an agent’s point of view punishment will cost the ‘cost to punish’/‘number of agents’ but will pay back ‘loss of punished’/‘number of agents’. This means that punishment will lead to a small increase of resources for the punishing agent if the costs to punish are less than the energy the punished loses (as it is in our simulation and in almost all Public Goods Games). This implies that punishment in this context is different from “altruistic” and completely disinterested punishment activity observed in UP and LP. Although in ZLP punishment is not costly from an agent’s point of view, it is costly from a global point of view, as the cost to punish still vanishes.

In figure 1 the four punishment mechanisms are classified according to two criteria: the presence of restrictions on punishment activity (restricted or unrestricted) and the presence of a net cost attached to the punishment activity (altruistic or not altruistic).

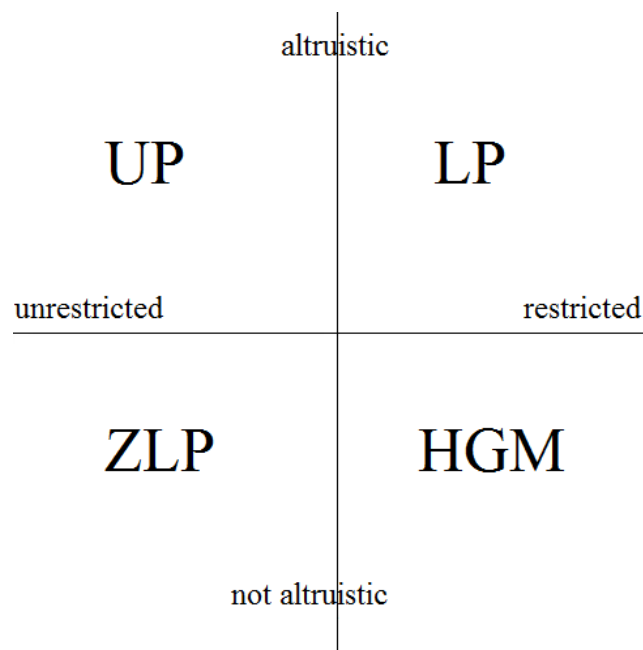


Figure 1: Placement of punishment mechanisms w.r.t. altruism and restriction involved

In our simulation we assume that also cooperation is costly. Agents must invest an extra amount x to cooperate/share y resources. This assumption of the model is based on the transaction cost theory by Williamson (1981) and captures the fact that moving resources from one actor to the other consumes resources (the ‘transaction costs’). If the transaction costs needed to cooperate with another agent are higher than the social synergy achieved, agents should not cooperate.

4. The model

Figure 2 shows a screenshot of the simulation, programmed with the open source software Breve 3D (Klein 2002), using the programming language Steve. No special libraries have been used for implementation. In the appendix we show details of the implementation. The following sections will describe the simulation on a conceptual level.

The white cubes in figure 2 represent resources. At the initialization “availabilityOfResources” resources are placed randomly in the environment. During the simulation a constant supply of resources $\text{availabilityOfResources}/25$ is put into the simulation at random positions. Resources move toward the agent that is closest by and as soon as they reach the agent they are ‘eaten’ (the object is destroyed) and the agent’s energy increases by 50. Because of the random positioning of resources, areas of the simulation differ in the amount of resources available. Without sharing of resources energy will be heterogeneous among agents.

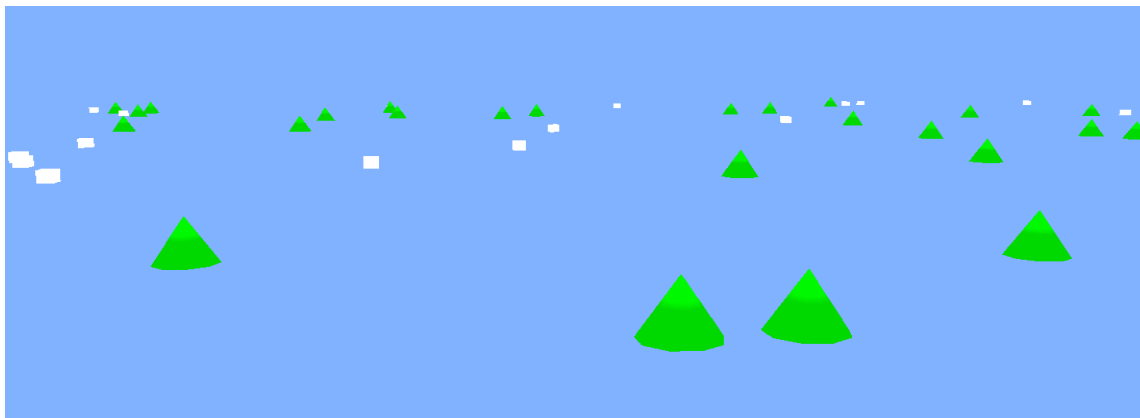


Figure 2: Screenshot, showing an excerpt of the environment in which agents interact (green cones are agents, white cubes resources)

In the screenshot the green cones represent agents that are placed randomly in a quadratic, flat area. Within a neighborhood agents can punish or share with other agents. The neighborhood size is chosen such that on average every agent has five other agents to interact with (the average degree of the agent network is 5). This number is comparable to the group-size in most studies done on Public Goods Games. Throughout a simulation an agent stays at its initial position, to avoid any effect of a certain kind of random movement of agents on cooperation as described by Smaldino and Schank (2012). Every iteration agents do three things: (1) Decide if (and if yes, who) to punish, (2) decide to share and (3) consume energy.

The way in which punishment works depends on the punishment mechanism and is described in the following paragraphs.

Every agent is initialized with two fixed parameters in its genome: toleranceS and toleranceP (always positive doubles). Whenever an agent shares with another, this will increase its reputation by the amount it has shared. A sharing action only effects the reputation for 200 iterations, so that the value of the reputation only gives information about the recent sharing history. An agent decides to share once per iteration with the poorest neighbor if that neighbor has toleranceS energy less. Whether this decision really leads to an action depends on a chance that is equal to 'others reputation' / 'own reputation' (if this value exceeds 1 it is rounded to 1). The reputation of an agent is initially 1 to prevent division by 0. Whenever an agent has the highest reputation it can be sure that others will share with it if the agent is the poorest in the neighborhood. If its reputation is low hardly anyone shares with it. This is the implementation of 'downstream reciprocity' as introduced by Nowak and Roch (2007). An agent punishes the richest neighbor once per iteration if that neighbor has toleranceP more energy.

In simulations where UP is the punishment mechanism an agent has to pay an incentive of one energy point to punish another agent (distract five energy points from the other). In UP punishment will thus consume six energy points in total. In simulations where LP is used only agents with a higher reputation can punish agents with a lower reputation. With ZLP the costs of the incentive for punishing is payed by all agents collectively and the energy that the agent who is punished loses is re-distributed to all agents. In HGM agents cannot punish each other. Instead nine 'hired guns' are evenly distributed in and observe a part of the area. Together they observe the entire field. Every gun observes about six agents. This number is similar to the group size used by Andreoni and Gee (2011). Furthermore it is the same number of agents that are in the neighborhood of agents in the other conditions. Once every 10 iterations a gun punishes the agent in its neighborhood with the lowest reputation. Cooperation is implemented by sharing. In all punishment mechanisms an agent has to invest six energy points in order to share five with another. Sharing thus leads to a loss of one energy point to the system of agents. A detailed description of an agent's and all other object's behavior in the simulation can be found in the appendix.

Each Agent starts with 100 energy and consumes energy per iteration. If an agent has 6 energy it cannot punish nor share and its energy consumption is 0. Hence an agent's energy cannot drop below 0. The consumption of an agent grows quadratic with the energy an agent has (Figure 3). If the total energy is distributed among few agents, total consumption is much higher than in the case in which total energy is distributed among a larger number of agents. The extreme cases are those in which one agent has all the energy (maximum disparity) and the case in which energy is evenly distributed among all the agents (minimum disparity). This assumption of the model is based on the literature of economic inequality. It has been found that in societies where disparity is high economic growth phases are more likely to end than in societies where disparity is low (Berg, Ostry and Zettelmeyer 2012). Furthermore high disparity is linked to high crime rates (Fajnzylber, Lederman and Loayza 2002) and bad health (Sapolsky 2005) in societies. In our model efficiency is therefore at its maximum when disparity is minimal, i.e. when agents share. This makes our interaction system similar to a typical social dilemma in which the single agent has the incentive to collect the maximum

amount of energy for itself, but the highest level of energy for the society is reached when all the agents share their energy.

The evolution of the population is implemented as follows. The fitness of an agent is equal to its energy. Every 50 iterations the agent with the worst fitness takes over slightly mutated values for toleranceS and toleranceP of the fittest agent. Energy and reputation are not changed. Every agent faces the trade-off between keeping energy high to have a high fitness and sharing to avoid punishment. Sharing brings also an indirect advantage since it decreases the disparity in the system, thereby the total energy consumption, and hence there will be more energy in the future that the agents can benefit from. Note that all agents keep their initial position and that the neighborhoods do not change during the simulation.

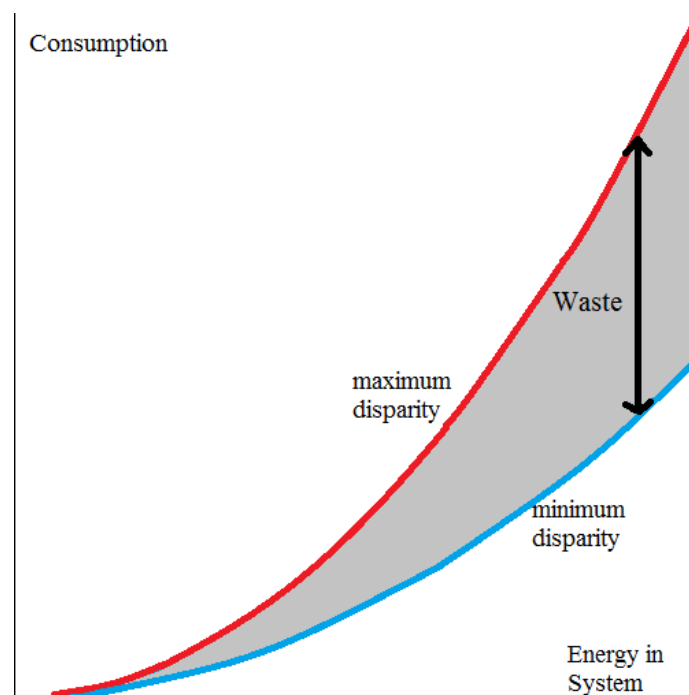


Figure 3: Showing how total energy consumption of agents per iteration increases quadratically when total energy in the simulation increases. Red, when one agent has all energy, blue, when all agents have equal energy

All simulations end after 50,000 iterations and per simulation there are 50 agents. The amount of resources set into the simulations varies and is implemented via a variable called availabilityOfResources. Possible initial values are 50 (low), 100 (mid) and 200 (high) with an addition of resources of respectively 2, 4 and 8 per iteration. AvailabilityOfResources is in some way similar to the ‘marginal per capita return’ (MPCR) in standard Public Goods Games. Figure 3 shows that with growing energy in the system the difference between the blue and the red function increases. When the availabilityOfResources increases agents will lose more and more resources when their disparity is high. Cooperation thus becomes more important. This reminds of the MPCR: When being high rewarding cooperation more than when it is low.

Per simulation only one punishment mechanism is used: no punishment and only sharing (NP), UP, LP, HGM and ZLP.

Punishment mechanism and availability of resources are treated as the independent variables. The dependent variable will be the performance of the system of agents (operationalized as the average energy of all agents during the last 10,000 iterations). Furthermore we will look at the change of agent behavior during simulations and interactions between the independent variables.

5. Results

Per possible combination of the two independent variables 30 simulations were performed, leading to 3 (availabilityOfResources) x 5 (punishment mechanism) x 30 = 450 simulations. Within the groups based on availabilityOfResources and punishment mechanism the average energy level during simulations was distributed close to normal (kurtosis and skewness always between -1.2 and 1.0). Hence an ANOVA was used to analyze the effect of the independent variables on average energy of agents. For a better understanding table 1 gives an overview of terms that we use in this section and their operationalization.

Table 1: Operationalization of variables. Every variable includes the last 10,000 iterations

<i>average energy</i>	average energy of all agents during simulation
<i>disparity</i>	average absolute deviation of energy during simulation
<i>sharing actions</i>	average number of sharing actions per iteration during simulation
<i>(antisocial) punishments</i>	average number of punishments per iteration during simulation
<i>coefficient of variation</i>	disparity / average energy

Figure 4 (left) shows the average energy of all simulations during the last 10,000 iterations. When the availabilityOfResources was ‘low’ unrestricted punishment (UP), legitimate punishment (LP) and zero loss punishment (ZLP) were performing better than no punishment (NP). This difference was statistically significant only for NP vs. ZLP ($p = 0.029$), not for UP ($p = 0.135$) and LP ($p = 0.085$). The hired gun mechanism (HGM) was performing the worst when availabilityOfResources is ‘low’ and the difference NP vs. HGM was significant ($p = 0.042$). The right punishment mechanism (ZLP) thus can increase the performance of agents that have to share resources when the availability of resources is low. But the wrong mechanism (HGM) can lead to a decrease.

Figure 4 (right) shows that the UP, LP and ZLP were the conditions where most sharing took place when availabilityOfResources was ‘low’. It seems that when the availability of resources is low indirect reciprocity as used in NP is not strong enough to enforce cooperation in a system. Avoiding punishment can be an extra motivator for agents to serve to public good instead of egoism.

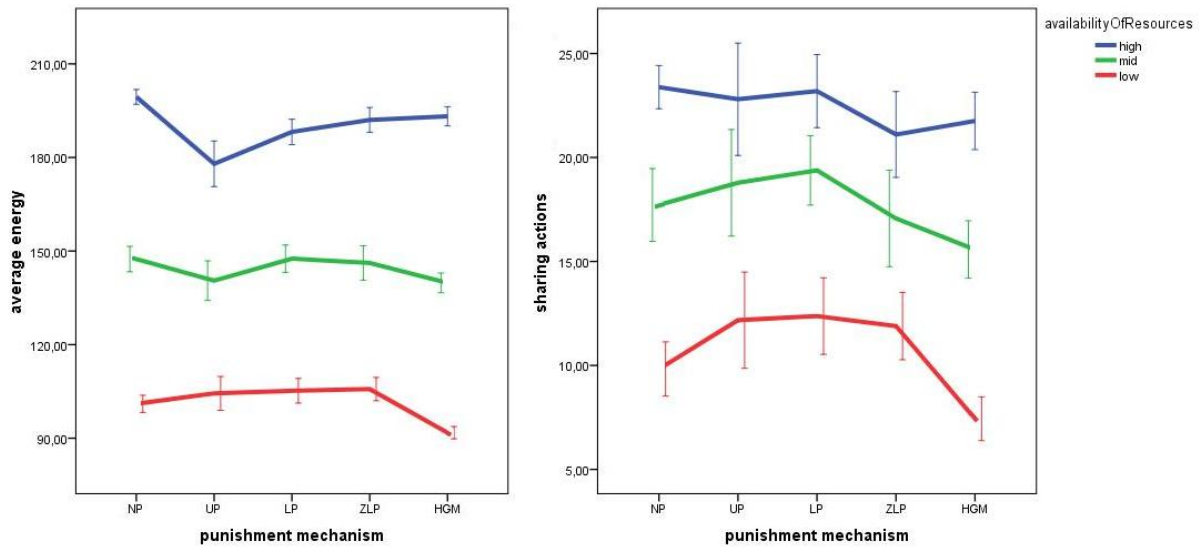


Figure 4: Average energy (left) and sharing actions (right) in various simulations. Bars indicate the 95 % confidence interval

Things change when the availability of resources is ‘high’. Here NP is performing the best of all mechanisms. Differences are significant for NP vs. UP ($p < 0.001$), LP ($p = 0.002$) and ZLP ($p = 0.026$). For HGM the difference was only marginally significant ($p = 0.081$). It seems that when there are many resources available to a system of agents punishment wastes resources and is a bad ingredient for efficient cooperation. Opposite to the simulations where availabilityOfResources was low figure 4 (right) shows that the differences in performance of mechanisms is not associated with the amount of sharing actions during the simulations when availabilityOfResources is ‘high’ or ‘mid’.

When availabilityOfResources was ‘mid’ none of the differences in figure 4 (left) between punishment conditions and NP were statistically significant. Furthermore no punishment mechanism differed from any other mechanism significantly.

Furthermore we can observe in figure 4 (left) that HGM becomes more effective compared to other mechanisms that include punishment if the availabilityOfResources increases. It seems that periodically punishment by an external agent has the best effect when the availability of resources is high. Otherwise punishment should be conducted by the agents themselves.

We can observe that LP is always performing better than UP. Figure 5 (left) shows how many punishments were performed per iteration during the simulations when availabilityOfResources was ‘high’. Note that there are much less punishment actions than there are sharing actions (figure 4, right). Since in NP punishment was not possible it is not included in the figure. Figure 5 (right) shows how much of this punishment was antisocial punishment (antisocial punishment is only possible for punishment mechanisms UP and ZLP). In the simulations with UP more than half of all punishment was antisocial punishment. For the figure data from the simulations where availabilityOfResources was ‘high’ is used but the ratio of punishment/antisocial punishment is similar for ‘low’ and ‘mid’. Antisocial punishment decreases the fitness of cooperators and, since in our model punishment was meant to facilitate cooperation, is a misuse of punishment. In the LP simulations the system

had to use much less punishments to keep cooperation going. Interestingly the ratio between punishment actions in general and antisocial punishment is almost exactly the same in ZLP. Nevertheless, as discussed in the previous sections, ZLP is performing as well as LP, sometimes even better. The performance of ZLP may increase significantly when it avoids antisocial punishment by incorporating legitimate punishment.

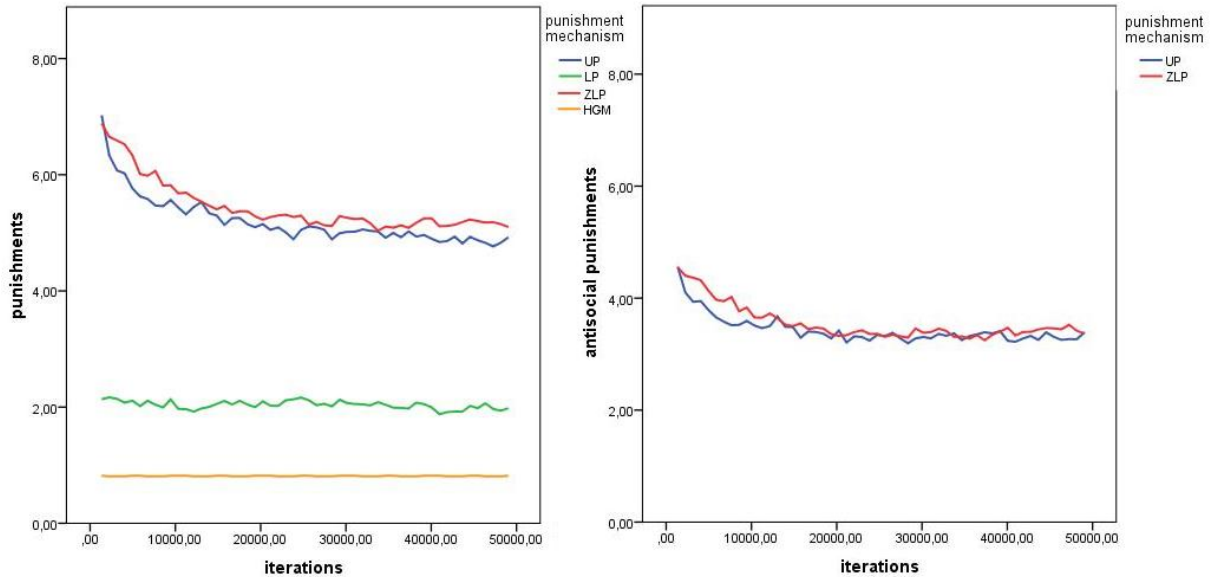


Figure 5: Development of punishment actions (left) and antisocial punishments (right) during simulations. For reasons of clarity every data point represents the average of 500 iterations.

It is also noticeable that after the first half of the simulation the amount of punishment actions stabilized (figure 5, left). Punishing others (though it was altruistic punishment in case of UP and LP) must have brought an (indirect) evolutionary advantage to the individual to remain encoded in the agents' genes.

Figure 6 (left) shows that average energy and the disparity (variation of energy level of agents) during a simulation are negatively correlated. In figure 6 (right) we see that ZLP was always the mechanism leading to the lowest disparity. In the figure we use the coefficient of variation (definition in table 1) to correct for different average energy in the conditions. As we already saw in figure 4 (left) ZLP was not always the best performing mechanism in terms of average energy of agents. Keeping the disparity low among agents seemed to be more effective when the availability of resources was low, than when it was high. When availabilityOfResources was 'high' it seems that NP and HGM are better in balancing the costs of sharing and a high disparity within the system.

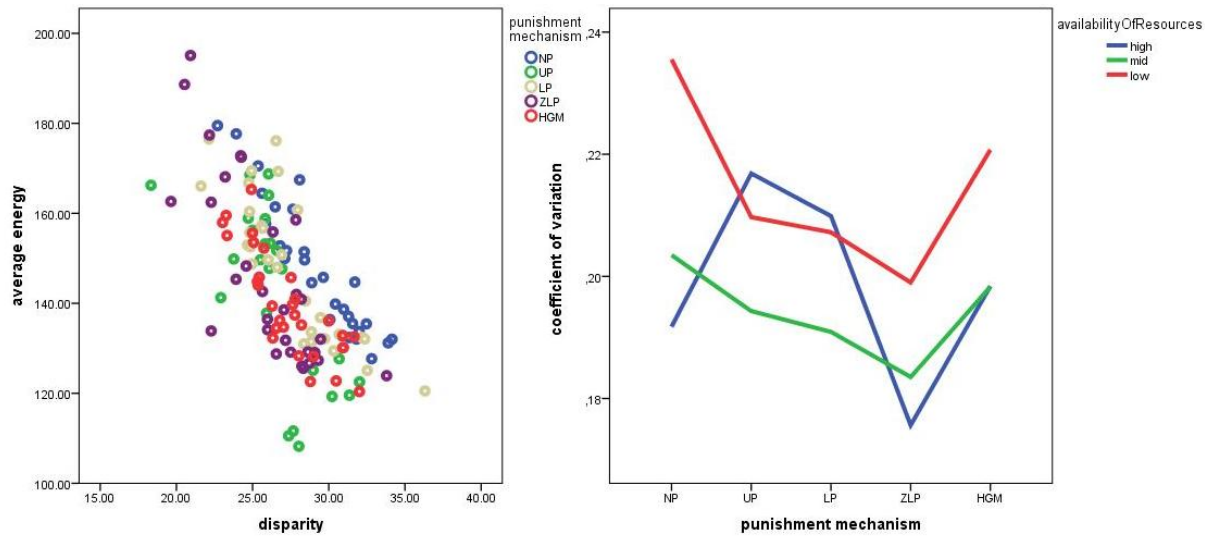


Figure 6: Left, Every dot represents the average of one simulation. Disparity and average energy are negatively correlated (availabilityOfResources = 'mid'). Right, Coefficient of variation of agent's energy during the last 10,000 iterations.

6. Discussion

With the help of an agent based simulation we showed that punishment can be a facilitator for effective cooperation. Punishment is not just a stopgap for cooperation when agents lack information about each other but can bring an additional coordinative advantage. In our simulation selection took place on the agent level. Since punishment was constant during evolution (figure 5) we showed that (altruistic) punishment is rational not just on a group level, but also for individuals, even when it is not explicitly needed to keep cooperation going. However, when which punishment mechanism works best and if it can play a facilitating role for maximizing the efficiency of cooperation depends on the environment in which agents interact.

Analysis of the simulations has shown that punishment mechanisms are most powerful as a facilitator of the effectiveness of cooperation if the availability of resources is low. Because of the energy consume function of agents this corresponds to Public Goods Games where the marginal per capita return (MPCR) is low. The MPCR works as a motivator for cooperation in Public Goods Games and the lower the return the lower the willingness of subjects to cooperate (Kim and Walker 1984). It seems that mechanisms like indirect reciprocity, giving all responsibility for cooperation to the individual, works not strong enough as a facilitator to ensure cooperation in these situations. It seems to be necessary and effective to give the group through punishment the possibility to steer cooperation when MPCR is low.

The results have also shown that the negative effects of antisocial and counter punishment can be effectively cut back through legitimate punishment. LP was always performing better than unrestricted punishment and this is confirming the results of Faillo et al. (2012). Although zero loss punishment was generally the punishment mechanism performing best, we expect ZLP's performance will increase when legitimate punishment is combined with it.

The results for the hired gun mechanism have to be interpreted with care. The effectiveness of HGM may depend on variables like punishment frequency and group size. By fine tuning them performance of HGM may increase significantly. However we saw that HGM can outperform other punishment mechanism even without fine-tuning when the availability of resources was high.

Research in Public Goods Games has focused on the extent to which punishment and reciprocity lead to cooperation, but cooperation should not be a goal in itself. In the simulations presented cooperation also consumes resources, leading to the necessity for agents to find a balance between benefits of cooperating and not cooperating. This is a fact ignored within the framework of Public Goods Games and our results show that it can be a crucial aspect when judging the effectiveness of resource management.

In the presented simulations all agents were choosing the action that was best for them. Despite this egocentric point of view, agents did neither stop punishing nor sharing, although this did not bring a direct advantage to them. We confirm the findings of Ye et al. (2011) that through the incorporation of reciprocity in our evolutionary model the first- (why should we share?) and second-order social dilemmas (why should we altruistically punish?) resolve. Contrary to Ye et al. ‘altruistic’ behavior was not rewarded directly by the group but indirectly through a higher chance that others will cooperate. Furthermore we did not make any assumptions about the types of agents that could exist and their behavior, they just evolved themselves.

The environments investigated in the simulations only differ in one parameter: the availability of resources. Many more parameters such as a changing availability of resources during simulation, or a changing number of agents are possible and should be taken into account in order to increase the external validity of our results. Furthermore all agents were equal in their energy consumption and only differed with respect to tolerance_S and tolerance_P. Humans are far more diverse and research is needed to understand how well various punishment mechanisms can deal with this diversity.

Because of the simulation methodology and the evolutionary learning algorithm underlying, this research has a strong computational flavor. It is not often fully realized that social simulation cannot only help to validate theories in economics and social science, but the findings can also be used to create or improve artificial social intelligence. The results point towards solutions for problems in e.g. decentralized power grids or wherever software agents have to autonomously share resources. In order to keep e.g. a power grid stable very quick decisions have to be made about when energy producers are allowed to feed electricity into the grid or when e.g. extra energy has to be bought from foreign countries. Instead of centrally steering this network, one could decide to give agents (energy producers) local control about the energy grid. This would avoid exploding computational complexity of decisions in such networks and increase the speed and flexibility with which the system can react. The objectives would be similar to those in the simulation: Maximize own energy feed-in, while sharing and punishing others on rights to feed energy. Given the increasing size, flexibility, and demands on such systems, we assume that more social intelligence is needed for agents within these networks.

7. Additional content

The paper presented in the previous sections is based on the final version of the simulation, after more than 6 months of programming, testing and analysis. The sections 7.1 and 7.2 I present some of the intermediate results that are not mentioned in the paper for two reasons. Either they were simply reproduction of previous well-known phenomena or by discussing them the paper would have exceeded the page limit of the journal it was submitted to. Nevertheless I believe that some of them point towards interesting directions.

The simulation and its parameters have changed during the 6 months it has been developed and the results in this chapter are not based on the most recent version of the simulation (used for the paper). Comparing results of this chapter with those presented in the paper may lead to frictions. The results in this chapter therefore have to be interpreted with care and have to be seen as what they are: Results from a pilot study, done with not fully tested software.

For the pilot study 480 simulations were performed, 40 per condition based on 3 values for availability of resources (low, mid, high) and 4 punishment mechanism (NP, UP, LP, HGM).

7.1 DIFFERENT AGENTS

In the early stage zero loss punishment was not yet implemented. This chapter knows therefore only four punishment mechanisms: No punishment (NP), unrestricted punishment (UP), legitimate punishment (LP), and the hired gun mechanism (HGM). From a conceptual point of view agents in the early versions of the simulation are different in two ways. I will discuss these differences and their effects on the behavior of the agents in this section.

The first difference is that indirect reciprocity was not implemented in the early model. Agents shared whenever another agent's energy level was toleranceS points below their own. Opposite to the final version of the simulation, there was no chance (based on agents' reputation) involved. Second, the function through which the fittest agent was determined was different in the pilot study. To avoid that sharing behavior among agents died out, I chose to let the fitness of agents not just depend on their energy level, but also on their reputation. The fitness function was $= 1/(\text{Rank according to energy level} * \text{Rank according to reputation})$. As in the final simulation, agents with the worst fitness took over the strategy of those with the highest fitness.

Figure 7 shows how the sharing and punishing developed during the simulation under these two main differences. ToleranceS(-haring) and toleranceP(-unish) show the average difference that was accepted by agents before they started to punish or share with other agents. Although the fitness function was able to slow down the increasing tolerance, cooperation died out on the long run. ToleranceS and toleranceP simply became too high to lead to sharing or punishment.

In my simulation all punishment mechanisms were not able to keep cooperation going on the long run. However, in the conditions in which agents could punish figure 7 (left) shows that the willingness of agents to share decreased not so quickly when punishment was present. In UP and LP punishment was – as discussed in the previous chapter – altruistic. Figure 7

(right) shows that agents kept punishing for a longer time when punishment was restricted and antisocial punishment was - as in LP - not possible.

As soon as I implemented indirect reciprocity in the simulation it was not necessary to let agents' fitness depend on the reputation to keep cooperation going. Hence I choose – as discussed in the paper - to simplify the function and make agents' fitness equal to their energy.

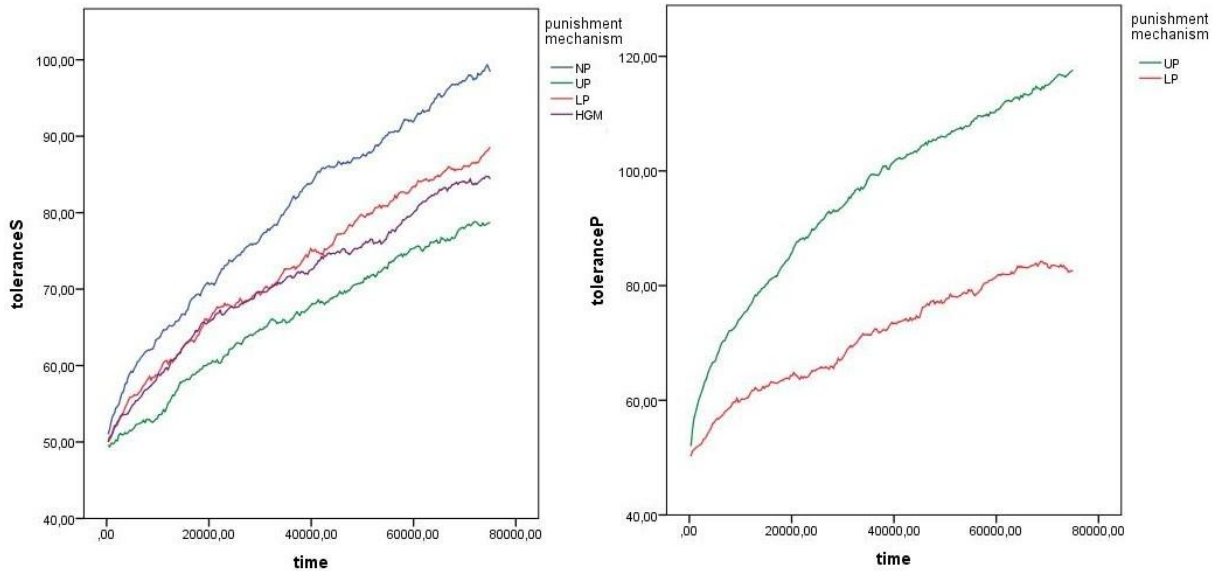


Figure 7: Development of average tolerance S (left) and tolerance P (right) during the pilot-study. The right figure does not include NP and HGM since tolerance P was not leading to action under these conditions

7.2 ADDITIONAL EXPERIMENTAL VARIABLES

Besides the main differences discussed in the section 7.1 I also experimented with two other variables not mentioned in the paper. The first is called “season difference”. Possible values were 0 (no) and 0.3 (yes). A value of 0.3 represents environments in which the availability of resources changed by 30 % every 1000 iterations in a winter/summer style. During winter only 85% of the resources suggested by the variable “availabilityOfResources” were placed into the environment and in summer it there were 115%. A value of 0 stands for environments in which resource supply is constant. I hoped to find that some punishment mechanisms can deal better with fluctuating availability than others. Figure 8 (left) shows that there was no big difference between simulations in which there were seasons and those where there was none. Figure 8 only shows the values when availability of resources was ‘mid’, but the picture is the same for ‘low’ and ‘high’.

The second variable was the chance with which an agent's decision to punish really lead to punishment. In the final simulation an agent's decision to punish led certainly to the punishment action, but in the early versions we also had simulations where the chance was only 0.5 for punishment to happen. The intuition was that the threat of punishment may be so strong that wasting energy through actual punishment is not always necessary. Figure 8 (right) shows that this is indeed true in the model. Agents in conditions with punishment were performing significantly better when the chance was 0.5. However, figure 8 (right) also shows

that there was no interaction between the variable ‘chance’ and ‘punishment mechanism’. We therefore decided to keep the model as simple as possible in the final simulation and set chance to 1.

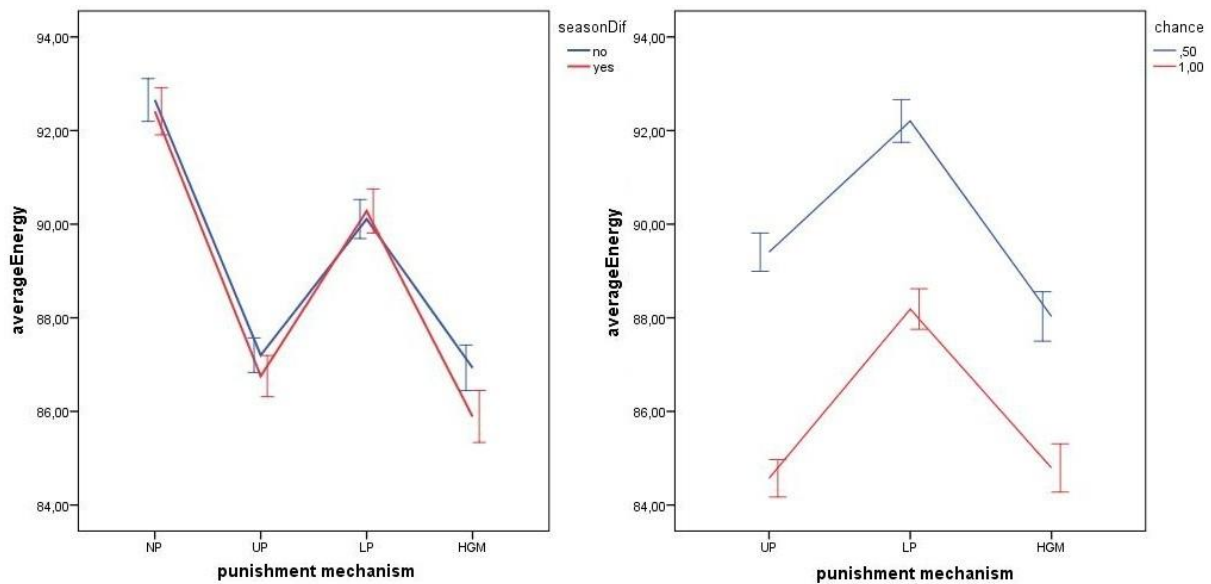


Figure 8: Difference in average energy during the pilot study w.r.t. the groups based on punishment mechanism and season difference (left)/ punishment chance (right). Bars indicate the 95 % confidence interval

7.3 EXTRA RESULTS OF THE FINAL SIMULATION

Figure 9 is based on data from the final simulation discussed in sections 2-6 and was not included for reasons of space limits of the journal. They show the average toleranceS and toleranceP per iteration (averaged over availabilityOfResources).

Generally we see in figure 9 (left) that agents become more cooperative during time even without punishment. This is very different from the early model in figure 7 (left) where toleranceS was increasing during the simulation and agents were thus less cooperative in the course of the simulation. The difference can be explained through the presence of indirect reciprocity in the final version of the simulation.

Values for toleranceS (figure 9, left) drop much quicker when agents can punish each other (UP, LP, ZLP) than when punishment is not possible (NP). In the simulations punishment was thus not necessary for cooperation but was increasing the speed with which cooperation was learned. HGM seems to be an exception. Maybe guns did not punish enough or too much in this condition. As discussed in section 6 HGM may become more effective when the amount of guns and their punishment frequency are chosen carefully.

Figure 9 (right) shows that toleranceP was lower in LP than in other punishment mechanisms allowing for antisocial punishment (UP and ZLP). Agents in simulations with LP were thus more sensible to disparity than with other punishment mechanisms. We can only speculate about the reasons, but one explanation could be that when counter/antisocial punishment is not possible (as in LP) it is not so dangerous for a group of agents to punish quicker or more rigorously. The chance that punishment gets hold of the agent that is lowering the group’s performance is bigger in LP and therefore rigorously punishing is more

effective than with UP and ZLP. However agents were not strategically thinking, but simply react to their own energy and those of others in the way their genes force them to. In LP a low toleranceP was the most effective for agents and hence this low toleranceP gene became dominant in the population.

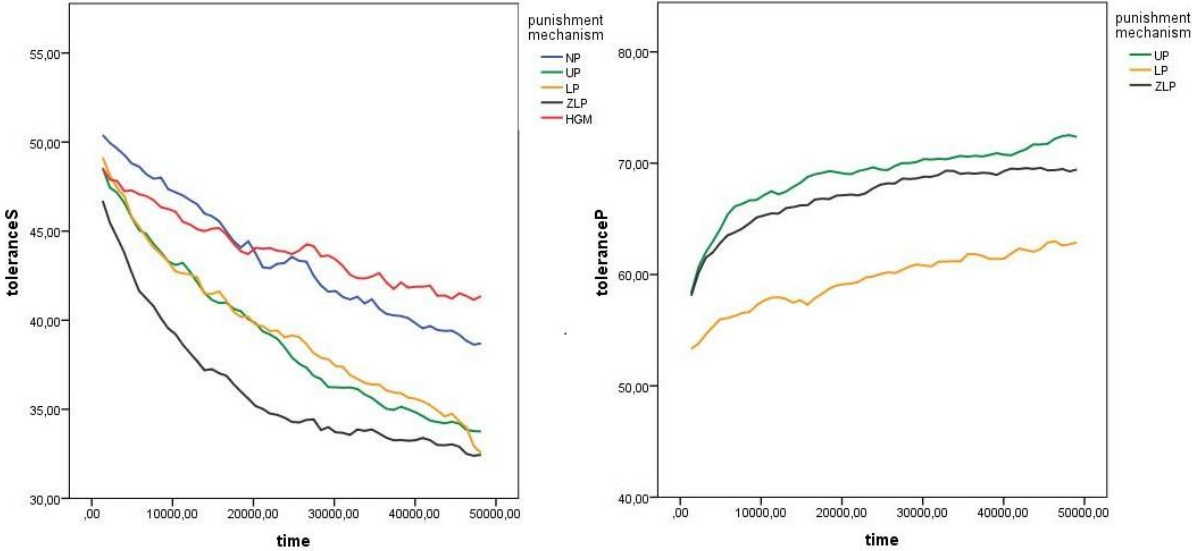


Figure 9: average toleranceS (left) and toleranceP (right) per iteration (averaged over availabilityOfResources)

References

- ANDREONI, J., & Gee, L. K. (2012). Gun for Hire: Delegated Enforcement and Peer Punishment in Public Goods Provision. *Journal of Public Economics*, 96, pp. 1036-1046.
- AXELROD, R. (1984). *Evolution of Cooperation*, Basic Books.
- BERG, A. G., Ostry, J. D., & Zettelmeyer, J. (2012). What makes growth sustained? *Journal of Development Economics*, 98(2), pp. 149-166.
- CHAUDHURI, A. (2011). Sustaining cooperation in laboratory public goods experiments: a selective survey of the literature. *Experimental Economics*, 14, pp. 47-83.
- DAWES, R. M., & Messick, D. M. (2000). Social dilemmas. *International Journal of Psychology*, 35, pp. 111- 116.
- FAILLO, M., Grieco, D., & Zarri, L. (2013). Legitimate Punishment, Feedback, and the Enforcement of Cooperation. *Games and Economic Behavior*, 77, pp. 271-283.
- FAJNZYLBER, P., Lederman, D., & Loayza, N. (2002). Inequality and Violent Crime. *Journal of Law and Economics*, 45(1), pp. 1-40.
- FARMER, J. D., & Foley, D. (2009). The economy needs agent-based modeling. *Nature*, 460(6), 685-686.
- FEHR, E., & Gächter, S. (2000). Cooperation and Punishment in Public Goods Experiments. *American Economic Review*, 90, pp. 980-994.
- FEHR, E., & Gächter, S. (2002). Altruistic Punishment in Humans. *Nature*, 415, pp. 137-140.
- GILBERT, N., & Troitzsch, K. (1999). *Simulation for the Social Scientist*. Buckingham: Open University Press.
- HENRICH, J. (2004). Cultural group selection, coevolutionary processes and large-scale cooperation. *Journal of Economic Behavior and Organization*, 53, pp. 3-53.
- HERRMANN, B., Thöni, C., & Gächter, S. (2008). Antisocial Punishment across Societies. *Science*, 319, pp. 1362-1367.
- HUGHES, H. P. N., Clegg, C. W., Robinson, M. A., & Crowder, R. M. (2012). Agent-based modeling and simulation: The potential contribution to organizational psychology. *Journal of Occupational and Organizational Psychology*, 85, pp. 487-502.
- JAFFE, K., & Zaballa, L. (2010). Co-Operative Punishment Cements Social Cohesion. *Journal of Artificial Societies and Social Simulation*, 13(3)4, <http://jasss.soc.surrey.ac.uk/13/3/4.html>
- KIM, O., & Walker, M. (1984). The free rider problem: Experimental evidence. *Public Choice*, 43, pp. 3-24.

- KLEIN, J. (2002). Breve: a 3D simulation environment for the simulation of decentralized systems and artificial life. *Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems*. The MIT Press.
- LEDYARD, J. (1995). Public Goods: A Survey of Experimental Research. In Kagel, J and Roth, A (eds): *Handbook of Experimental Economics* (pp. 111-181). Princeton: Princeton University Press.
- MACY, M. W., & Wiler, R. (2002). From Factors to Actors: Computational Sociology and Agent-Based Modeling. *Annual Review of Sociology*, 28, pp. 143-166.
- MAE, F., & Mac, F. (2010). Agents of change. *The Economist*. Retrieve from on 01.06.2013: <http://www.economist.com/node/16636121>
- MESSICK, D. M., & Brewer, M. B. (1983). Solving social dilemmas: A review. *Review of Personality and Social Psychology*, 4, pp. 11-44.
- NOWAK, M. A. (2006). Five rules for the evolution of cooperation. *Science*, 314, pp. 1560-1563.
- NOWAK, M. A., & Roch, S. (2007). Upstream reciprocity and the evolution of gratitude. *Proceedings of the Royal Society of London, Series B: Biological Sciences*, 274, pp. 605-610.
- OSTROM, E. (1990). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press.
- RAPOPORT, A., & Chammah, A. M. (1965). *Prisoner's Dilemma*, Univ. of Michigan Press.
- SAPOLSKY, R. (2005). Sick of Poverty. *Scientific American*, 293, pp. 92-99.
- SMALDINO, P. E., & Schank, J. C. (2012). Movement patterns, social dynamics, and the evolution of cooperation. *Theoretical Population Biology*, 82, pp. 48-58.
- SQUAZZONI, F. (2012). *Agent-Based Computational Sociology*. John Wiley & Sons.
- TROISI, A., Wong, V., & Ratner, M. A. (2004). An agent-based approach for modeling molecular self-organization. *Proceedings of the National Academy of Science of the United States of America*, 102(2), pp. 255-260.
- WILLIAMSON, O. E. (1981). The Economics of Organization: The Transaction Cost Approach. *The American Journal of Sociology*, 87, pp. 548-577.
- WOOLDRIDGE, M., & Jennings, N. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10, pp. 115-152.
- YE, H., Tan, F., Ding, M., Jia, Y., & Chen, Y. (2011). Sympathy and Punishment: Evolution of Cooperation in Public Goods Game. *Journal of Artificial Societies and Social Simulation*, 14(4)20, <http://jasss.soc.surrey.ac.uk/14/4/20.html>

Appendix

In the following, you see a semi-formal description of objects active during the simulation. Objects contain a list of variables, the Init-method (executed when the object is created), an Iterate-method (executed every iteration, beginning one iteration after creation), and if needed a set of extra functions. The Controller is created as soon as the simulation starts and creates and coordinates all other objects. Variables are written in italics, the name of the object bold.

CONTROLLER

Variables:

availability-

OfResources {50, 100, 200}. The larger this value the more new resources will be set into the environment per iteration

condition {NP, UP, LP, ZLP, HGM}

pot Energy that was distracted from agents that got punished (double)

Init: create 50 agents and *availabilityOfResources* resources;
if (*condition* == HGM):
 create 9 guns and distribute them evenly in the field;

Iterate: create *availabilityOfResources/25* new resources;
 every 50 iterations: tournament ();
 if (*condition* == ZLP):
 splitPot ();

splitPot (): for every agent: *energy* += *pot/50*;
pot = 0;

tournament (): best = agent with highest *energy*;
 worst = agent with lowest *energy*;
 worst.toleranceS = (best.toleranceS + randomGauss);
 worst.toleranceP = (best.toleranceP + randomGauss);

AGENT

Variables:

energy A positive double initially 100 that decreases with consume, sharing, and punishment. Increases with every resource eaten or when another agent is sharing

reputation decreases over time to a minimum of 1. If an agent shares this value increases. (double)

toleranceS threshold value (positive double), if another agent in the neighborhood has *energy* < (own *energy* – *toleranceS*) agent will share with this agent

toleranceP threshold value (positive double), if another agent in the neighborhood has $energy > (own\ energy + toleranceP)$ agent will punish this agent

neighbors list of agents within a 40 units radius.

position Two dimensional vector (x,y)

Init: $position = (random(-100, 100), random(-100, 100))$;
 $toleranceS = random(0, 100)$;
 $toleranceP = random(0, 100)$;
 $energy = 100$;
 $reputation = 1$;

Iterate: $energy -= (energy/100)^2$;
if ($energy < 6$):
 end iterate;
a = agent in neighbors with lowest *energy* ;
if ($a.energy < (energy - toleranceS)$):
 share (a, $reputation/a.reputation$) ;
if ($controller.condition \neq HGM$):
 a = agent in neighbors with highest *energy* ;
if ($a.energy > (energy + toleranceP)$):
 if ($controller.condition == LP$):
 if ($a.reputation \leq reputation$):
 punish (a);
 else
 punish (a);

share (a, p): if ($random(0, 1) \leq p$):
 $energy -= 6$;
 $a.energy += 5$;
 $reputation += 5$;
after 200 iterations: $reputation -= 5$;

punish (a): $energy -= 1$;
 $a.energy -= 5$;
if ($controller.condition == ZLP$):
 $controller.pot += 5$;

GUN

Variables:
neighbors list of agents within a 50 units radius.
position Two dimensional vector (x,y)

Iterate: every 10 iterations:
a = neighbor with lowest *reputation* ;
punish (a);

punish (a): $a.energy -= 5$.

RESOURCE

Variables:

position Two dimensional vector (x,y)

Init: *position* = [random (-100, 100), random (-100, 100)];

Iterate: wander around randomly in field;
if (contact to agent):
 agent.*energy* += 50;
 free self;