M. D. Farjam, M. Faillo, W.F.G. Haselager and I.G. Sprinkhuizen-Kuyper

Punishment Mechanisms and their Effect on Cooperation - A Simulation Study

CEEL Working Paper 2-13

Cognitive and Experimental Economics Laboratory

Via Inama, 5 38100 Trento, Italy

http://www-ceel.economia.unitn.it
tel. +39.461.282313

Punishment Mechanisms and their Effect on Cooperation - A Simulation Study

M. D. Farjam [1], M. Faillo [2], W.F.G. Haselager [3], and I.G. Sprinkhuizen-Kuyper [3]

Abstract

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 a well-chosen punishment mechanism can play a facilitating role for cooperation even if the cooperating system already adopted reciprocity.

[1] Department of Artificial Intelligence, Radboud University Nijmegen, The Netherlands
[2] Department of Economics and Management, University of Trento, Italy
[3] Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, The Netherlands

Keywords: Public Goods Games, Punishment, Cooperation, Reciprocity, Evolution of Cooperation

1. 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 him. 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 were cooperation leads to a high pay-off 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.

2. 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 clearly destructive 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 only agents that are good cooperators can punish agents that are bad cooperators. This 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).

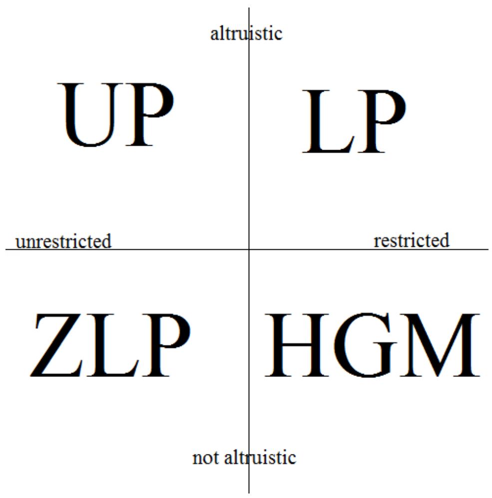

Figure 1: Placement of punishment mechanisms w.r.t. altruism and restriction involved

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.

3. Simulation

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.

The white cubes in figure 2 represent resources. During the simulation a constant supply of resources 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.

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). Throughout a simulation an agent stays at his 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: Decide if (and if yes, who) to punish, decide to share and consume energy. The way in which punishment works depends on the punishment mechanism and is described in the following paragraphs.

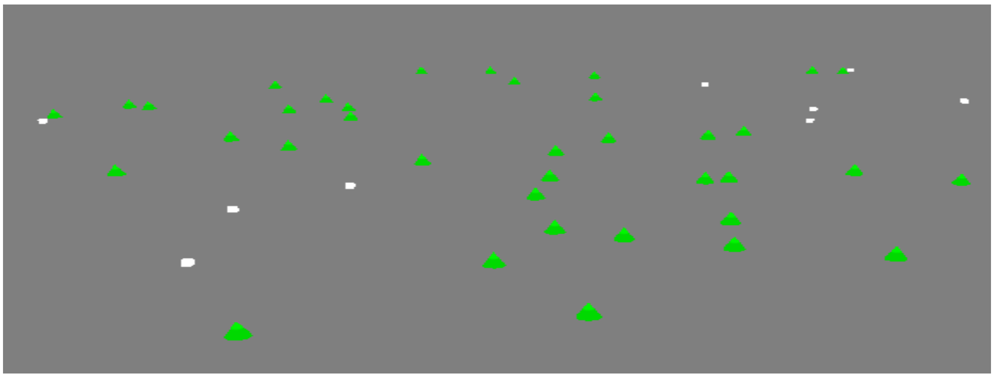

Figure 2: Screenshot (green cones are agents, white cubes resources)

Every agent is initialized with two fixed parameters in his genome: toleranceS and toleranceP (always positive doubles). Whenever an agent shares with another, this will increase his reputation by the amount he 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). Whenever an agent has the highest reputation he can be sure that others will share with him if he is the poorest neighbor of an agent. If his reputation is close to zero hardly anyone shares with him. This is the implementation of 'downstream reciprocity', as described in the first section (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 collectivly 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. Within its neighboorhood a gun punishes the agent with the lowest reputation every 10 iterations. In all punishment mechanisms an agent has to invest six energy points in order to share five with an agent. Sharing thus leads to a loss of one energy point to the system of agents.

Agents consume energy per iteration. If an agent has 0 energy he cannot punish nor share and his energy consumption is 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 himself, but the highest level of energy for the society is reached when all the agents share their energy.

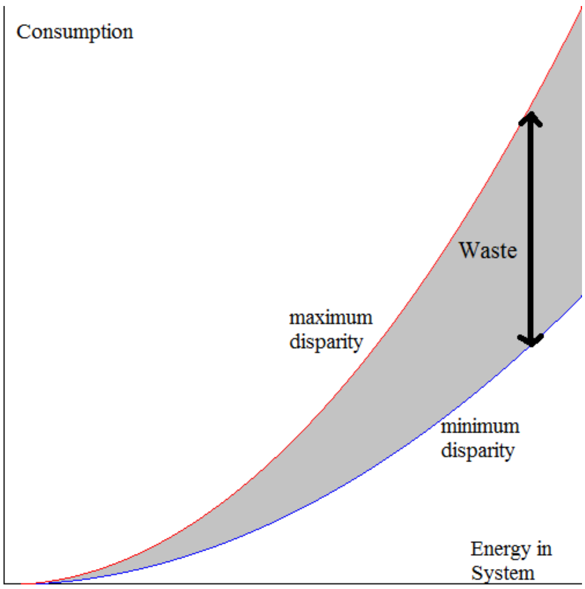

Figure 3: Showing how total energy consumption of agents per iteration increases quadratically when total energy in the simulation increases. Red, when one agent has all energy, blue, when all agents have equal energy

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. The fitness of an agent is equal to its energy. 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.

All simulations end after 50.000 iterations and per simulation there are 50 agents. The amount of resources set into the simulations varies. Possible values are 50 (low), 100 (mid) and 200 (high). This variable is called availabilityOfResources and it 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 proportionally 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.

4. 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.

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

Table 1: Operationalization of terms

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.

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'.

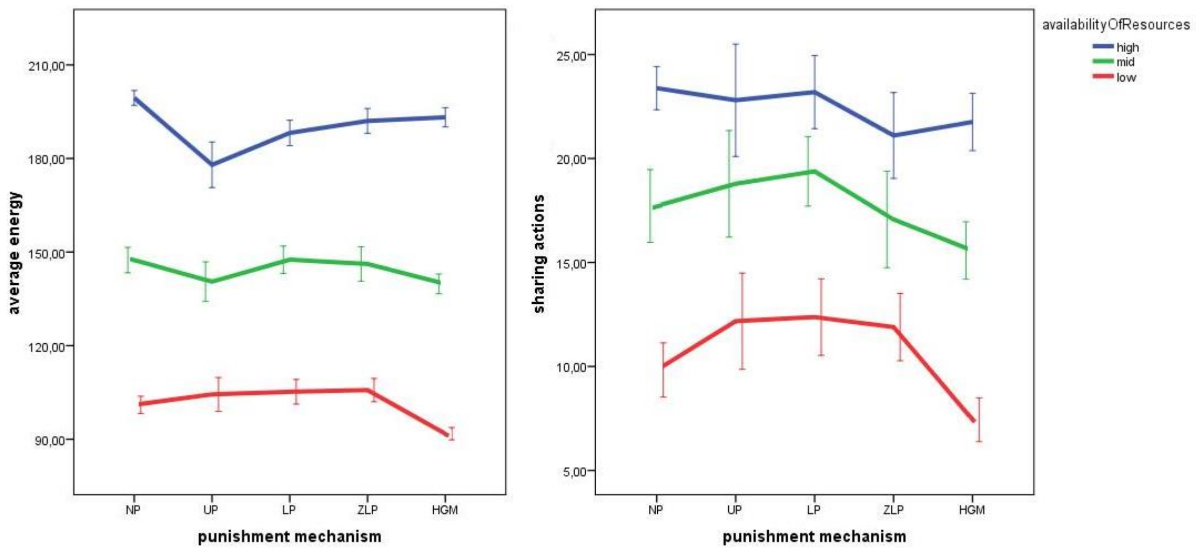

Figure 4: Average energy (left) and sharing actions (right) in various simulations. Bars indicate the 95 % confidence interval

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 mechanism 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 (note that in NP punishment was not possible). For the figure data from the simulations where availabilityOfResources was ‘high’ is used, but the picture is similar for ‘low’ and ‘mid’. 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. Antisocial punishment is not just a waste of resources but is also used to decrease the fitness of cooperators. This is clearly not effective. 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.

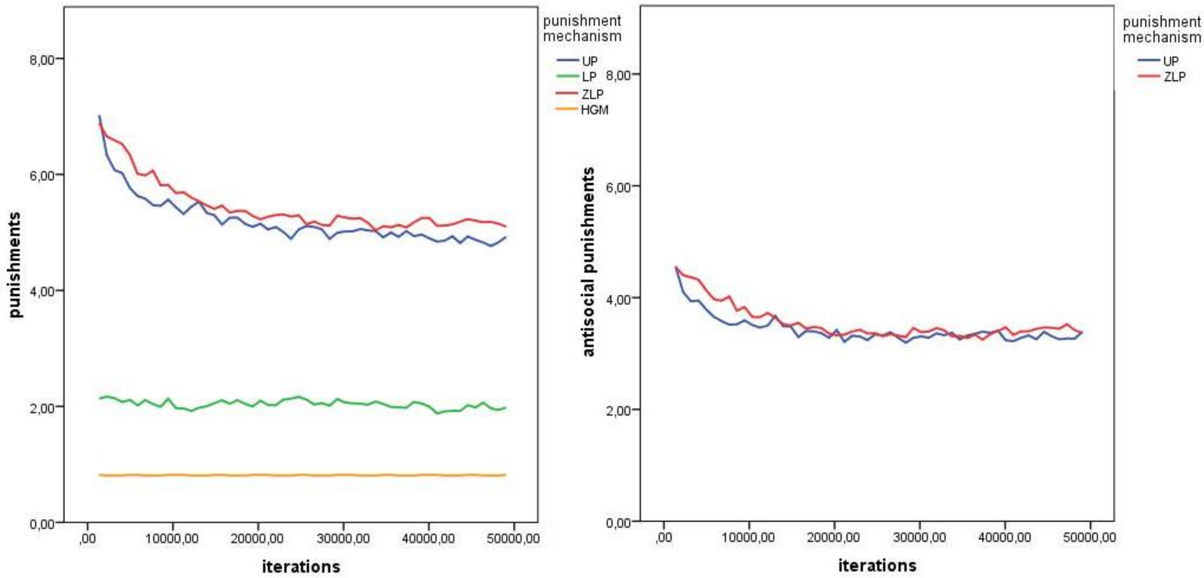

Figure 5: Development of punishment actions (left) and antisocial punishments (right) during simulations. For reasons of clarity every data point represents the average of 500 iterations.

It is also noticeable that after the first 10.000 iterations the amount of punishment actions was quite constant during the course of the simulations (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 not as effective when the availability of resources was high, than when it was low. 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.

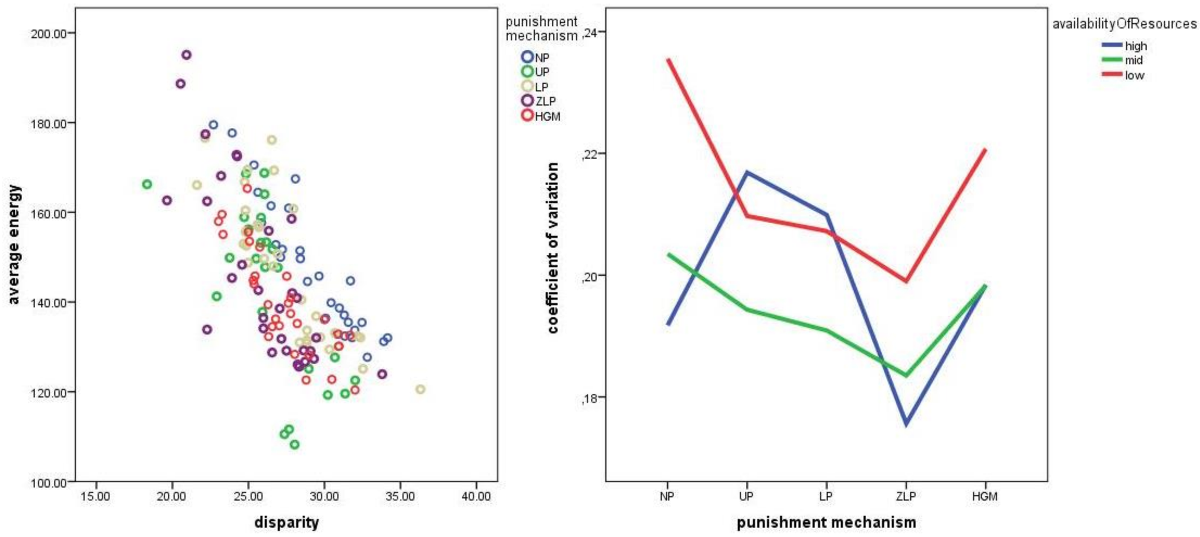

Figure 6: Left, disparity and average energy are negatively correlated (availabilityOfResources = 'mid'). Right, Coefficient of variation of agent's energy during the last 10.000 iterations.

5. 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. However, when which punishment mechanism works best and if it can play a facilitating for maximizing the efficiency of cooperation, even when it is not explicitly needed to keep it going. If punishment really facilitates cooperation depends on the kind of 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 toleranceS and toleranceP. 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 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. 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.

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.

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.

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.

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.

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.

WILLIAMSON, O. E. (1981). The Economics of Organization: The Transaction Cost Approach. *The American Journal of Sociology*, 87, pp. 548-577.

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 during every iteration), 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.

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* == 5):

create 9 guns and distribute them evenly in the field;

Iterate: create *availabilityOfResources*/25 new resources;

every 50 iterations: tournament ();

if (c*ondition* == 4):

spiltPot ();

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

pot = 0;

tournament (): b = agent with highest *energy;*

w = agent with lowest *energy;*

w.*toleranceS* = (b.*toleranceS* + randomGauss);

w.*toleranceP* = (b.*toleranceP* + randomGauss);

Agent

Variables:

energy	A double that decreases to zero with every iteration. Increases with every resource eaten.
reputation	decreases over time to a minimum of 1. If an agent shares this value increases. (double)
toleranceS	threshold value (double), if another agent in the neighborhood has *energy* < (own *energy* – *toleranceS*) agent will share with this agent
toleranceP	threshold value (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.

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* < *5*):

a = agent in neighbors with highest *energy*;

If (a.*energy* > (*energy* + *toleranceP*)):

punish (a);

share (a, p): If (random (0, 1) <= p):

energy -= 6;

a.*energy* += 5;

reputation += 5;

after 200 iterations: *reputation* -= 5;

punish (a): *energy* -= 1;

a.e*nergy* -= 5;

Gun

Variables:

neighbors list of agents within a 50 units radius.

Iterate: every 10 iterations:

a = neighbor with lowest *reputation;*

punish (a);

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

Resource

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

Iterate: wander around randomly in field;

If (contact to agent):

agent.*energy* += 50;

free self;