Optimised Reputation-Based Adaptive Punishment for Limited Observability

Samhar Mahmoud
Department of Informatics
King's College London,
London WC2R 2LS,
United Kingdom
samhar.mahmoud@kcl.ac.uk

Daniel Villatoro
Artificial Intelligence Research Institute (IIIA),
Spanish National Research Council (CSIC),
Bellatera, Barcelona, Spain
dvillatoro@iiia.csic.es

Jeroen Keppens
Department of Informatics
King's College London,
London WC2R 2LS,
United Kingdom
jeroen.keppens@kcl.ac.uk

Michael Luck
Department of Informatics
King's College London,
London WC2R 2LS,
United Kingdom
michael.luck@kcl.ac.uk

***Abstract*—The use of social norms has proven to be effective in the self-governance of decentralised systems in which there is no central authority. Axelrod's seminal model of norm establishment in populations of self-interested individuals provides some insight into the mechanisms needed to support this through the use of *metanorms*, but is not directly applicable to real world scenarios such as online peer-to-peer communities, for example. In particular, it does not reflect different topological arrangements of interactions. While some recent efforts have sought to address these limitations, they are also limited in not considering the point-to-point interactions between agents that arise in real systems, but only interactions that are visible to an entire neighbourhood. The objective of this paper is twofold: firstly to incorporate these realistic adaptations to the original model, and secondly, to provide agents with reputation-based mechanisms that allow them to dynamically optimise the intensity of punishment ensuring norm establishment in exactly these limited observation conditions.**

***Keywords*-norms; metanorms; punishment; adaptation.**

I. Introduction

At the advent of the social networking era, interactions in virtual environments are increasing dramatically yet, due to the magnitude and speed of these interactions, their regulation is becoming expensive, and even infeasible. Social norms offer a means to provide for the self-organisation of systems and societies, by delegating to the population itself the power (and responsibility) to impose appropriate behavioural standards [1], [2]. In this context, many have been concerned with the development of mechanisms to ensure the emergence of such social norms (e.g., [3], [4], [5], [6], [7]). In particular, researchers from many scientific areas have considered *punishment* as a key motivating element for norms to be established [8], [9], [10], [11]. Here, punishment is a second-order public good, typically incurring a cost for the punisher, but bringing a potential benefit to the population as a whole when correctly applied. While little attention has been paid to the mechanisms that motivate agents to bear the associated costs, some researchers [9], [10] have provided cognitive explanations for the emergence of punishment, suggesting that the salience of norms has an impact on the defence of these norms.

This paper is compatible with such views, although here we explore the alternative mechanism of metapunishment, proposed by Axelrod [12] as a means of ensuring not that norms are complied with, but that they are enforced, through agents metapunishing those who fail to punish a defecting agent.

Axelrod's model adopts two strong assumptions (one structural and another individual) that compromise the resemblance with the type of systems we are interested in simulating. First, the interactions among agents are fully accessible to the rest of the population (including the interactions themselves, the punishments and the metapunishments). Second, punishment is considered static and unchanging despite the existence of different levels of violations, or repetitions of violations.

In this paper, we adapt Axelrod's model so that interactions become dyadic, in line with the peer-to-peer philosophy (our domain of interest). While this reduces the available information for the population as a whole, it increases the privacy of individual agents and their behaviour. Assuming that different levels of violations can only be deterred with proportional intensities of punishment, which implies proportional costs to the punisher, agents need mechanisms to intelligently adapt the strength of punishment optimising the self-governance costs.

Because of the limitations associated with the realistic interaction topology, agents need to be given with more complex mechanisms than originally to personalise and optimise punishment. We have elsewhere considered the possibility of *experience-based adaptive punishment* [13], by which agents are able to determine an appropriate punishment for others based on their prior experience with these others. Experiments with such techniques suggest that it is indeed possible to achieve norm establishment with varying levels of punishment, optimising the costs for self-policing. However, in that model, agents interact only with their direct neighbours, building up experience about them through these interactions. In this paper, in contrast, we generalise this so that agents can interact with any other agent in the population, but this is hindered by a lack of experience about these others, since they vary rapidly and are not limited to a small pool. While the consequence of this approach is that experience-based adaptive punishment loses much of its value, the use of reputation can offer a solution in this new context. The main focus of this paper is thus to consider the use of reputation in supporting adaptive punishment in the context of dyadic interactions. Our results show that reputation allows agents to overcome the lack of direct experience of the broader set of agents involved in order to determine the most appropriate level of punishment.

The paper continues in Section 2 with a review of the metanorm model and its variants, as well as prior work on adaptive punishment in restricted environments. Section 3 then describes the adaptation of this model to facilitate pair-wise interactions at a distance, together with experimental results. In turn, Section 4 builds on this with the integration of *reputation*-based adaptive punishment and shows how this addresses the problems arising by extending the model in Section 3 to these more realistic scenarios. Finally, related work is discussed in Section 5 and Section 6 concludes.

II. Metanorm Model

Axelrod's metanorm game is similar to the extended prisoner's dilemma game (with the punishment phase), but also includes a third phase in which agents can metapunish those agents who have not punished defectors. Since punishment is a second-order public good, the lack of punishment can be understood as a second-order violation that can be corrected through the use of metapunishment. Such metapunishment is essentially an incentive mechanism for agents to maintain punishment in an earlier stage of the process.[1] The model introduced here is an adaptation of the original model, and can be divided into three different parts introduced in what follows: the game dynamics, the agent model, and adaptive punishment.

A. Game Dynamics

As has been described elsewhere, the metanorm model aims to simulate a realistic distributed system in which a community of self-interested agents is encouraged, without being instructed to do so by a central authority, to adhere to a behavioural constraint, or *norm*, that benefits the community but not the individual agent adhering to the norm. The simulation of this model provides an experimental setting that enables the test of under which conditions a situation arises in which the norm governs the behaviour of individual agents.

Inspired by Axelrod's model [12], our simulation focusses only on the essential features of the problem. In the simulation presented in Algorithm 1, agents play a game iteratively; in each iteration, agents first make binary decisions in a social dilemma situation to comply with the norm (providing a benefit to the society) or to defect (benefiting from the contributions of the other agents while avoiding the costs). Defection brings a reward for the defecting agent called *temptation*, and a penalty to all other agents called *hurt*. However, each defector risks being observed *by the other agents*[2] in the population, and punished as a result. These other agents thus decide whether to punish agents that were observed defecting, with a low penalty for the punisher known as *enforcement cost* and a high penalty for the punished agent known as *punishment cost*. Agents that do not punish those observed defecting risk being observed themselves, and potentially incur metapunishment. Thus, finally, each agent decides whether to metapunish agents observed to spare defecting agents. Again, metapunishment comes at a high penalty for the punished agent and a low penalty for the punisher, through the punishment cost and enforcement cost, respectively.

Now, in order to capture a key feature of computational systems such as on-line virtual communities, we adapt Axelrod's classic model by introducing a topological structure [14] that determines observability among agents, so that an agent's neighbours are the only witnesses of its interactions. If we apply this observability restriction to punishment, then metapunishment is only imposed by a non-punishing agent's neighbour.

B. Agent Model

The decisions of agents are driven by two private variables: *boldness*, and *vengefulness*. Boldness determines the probability that an agent defects, and vengefulness is the probability that an agent punishes or metapunishes another agent. These values are initialised randomly following a uniform distribution. In each round, agents are given a fixed number of opportunities to defect, in which boldness determines the probability that an agent defects, and vengefulness is the probability that an agent punishes or metapunishes another agent. Thus, the boldness and vengefulness of an agent are said to comprise that agent's *policies*. After several rounds of the game, each agent's rewards and penalties are tallied, and successful and unsuccessful strategies are identified. By comparing themselves to other agents on this basis, the policies of poorly performing agents are revised such that features of successful strategies are more likely to be retained than those of unsuccessful ones.

1) Policy Learning Mechanism: A major problem with Axelrod's model is due to the evolutionary approach adopted (as identified in [15], [16]). First, this evolutionary approach causes norms to collapse in the long term due to the noise created from the rapid combination of elimination and multiplication of agents in the population, and the mutation

[1] In contrast to other cognitive explanations [9], Axelrod's model rests on the premise that the motivation for agents to punish is to avoid the costs associated with metapunishment.

[2] Notice that in Axelrod's original model each defector could potentially be observed by any other agent in the population.

Algorithm 1 interact()

```
for each agent $i$ do
    for each opportunity to defect $o$ do
        if defect $(B_i)$ then
            $DS_i = DS_i + T$
            for each agent $j \in NB_i$: $j \neq i$ do
                $TS_j = TS_j + H$
                if punish $(j, i, V_j)$ then
                    $DS_i = DS_i + P$
                    $PS_j = PS_j + E$
                else
                    for each agent $k \in NB_j : k \neq i \wedge k \neq j$ do
                        if punish $(k, j, V_j)$ then
                            $PS_k = PS_k + E$
                            $NPS_j = NPS_j + P$
                        end if
                    end for
                end if
            end for
        end if
    end for
end for
```

involved in the process. Second, the original model assumes the existence of a central authority with access to all agents' scores in order to determine the evolutionary process, yet this is unreasonable.

In consequence, we replace this original approach with a reinforcement learning algorithm that limits accessibility to global information, and instead allows agents to learn from their own experience [17]. In this algorithm (Algorithm 2), agents adapt their policies (boldness and vengefulness) at the end of each round of the simulation through a form of q-learning [18], a reinforcement learning technique embedded in each agent. Here, agents track the utility gained or lost from choosing the different actions available, and modify the relevant action policy in the direction that either increases or decreases the chances of performing these actions in the future, which should improve their utility.

In particular, each agent keeps track of three different scores: one related to boldness and the other two related to vengefulness. The boldness-related score is the *defection score* (*DS*) (since boldness is responsible for defection) and is concerned with any utility gained (temptation) or lost (punishment) from defection. Agents increase their boldness if their defection score is greater than zero and decrease it otherwise.

The vengefulness-related scores are the *punishment score* (*PS*) and *non punishment scores* (*NPS*). The former is the utility lost from punishment, which consists of the cumulative enforcement cost that agents must pay when punishing or metapunishing, while the latter is the utility loss that results from the agent *sparing* another agent and can be seen as any metapunishment that is applied to the agent as a consequence. Agents increase their vengefulness if the *NPS* is better than the *PS* and decrease it in the opposite case.

Algorithm 2 learn($\gamma, oneLevel$)

```
$TS_i = TS_i + DS_i + PS_i + NPS_i$
if explore($\gamma$) then
    $B_i = random()$
    $V_i = random()$
else
    $\delta B_i =$ BAdaptiveLearning$(i, DS_i, oneLevel)$
    if $DS_i < 0$ then
        $B_i = max(B_i - \delta B_i, 0)$
    else
        $B_i = min(B_i + \delta B_i, 1)$
    end if
    $\delta V_i =$ VAdaptiveLearning$(i, PS_i, NPS_i, oneLevel)$
    if $PS_i < NPS_i$ then
        $V_i = max(V_i - \delta V_i, 0)$
    else
        $V_i = min(V_i + \delta V_i, 1)$
    end if
end if
```

2) Adaptive Policy Learning: In our model, agents do not adapt their policies in the same way: a policy that results in a low utility is altered differently to a policy that is not as bad. Therefore, agents change their policies proportionally to their success, following the WoLF philosophy [19], so that if the utility lost from taking a certain action is high, then the change to the policy is greater, and if the utility lost is low then the change to the policy is low.

In fact, the policy adaptation approach presented here is an enhanced version of an earlier approach [20], which depends on extreme cases (the best or worst scores that an agent can *possibly* obtain) that might not actually occur in real settings. As a result, this alternative approach depends on cases that actually do occur during agent interactions. This means that an agent changes its policies according to the difference between its current utility and the best or worst utility obtained in its history of interactions.

First, and with regard to boldness, the required change to an agent's boldness is calculated using the BAdaptiveLearning function of Algorithm 3. Here, agents keep track of two boldness-related historical variables: $HMaxDS_i$ is the maximum obtained defection score in i's history of interaction, and $HMinDS_i$ is the minimum obtained defection score in i's history of interaction. These two variables are updated according to the current obtained defection scores DS_i. Then, $factorB$, which determines the change that should be made to an agent i's boldness, is calculated based on the division of DS_i by $HMaxDS_i$ if DS_i is greater than zero, or on the division of DS_i by $HMinDS_i$ if DS_i is negative.

Given this, we now need to determine how $factorB$ can be used to change an agent's policy. In order to avoid dramatic policy movements that could lead to violent fluctuations, we limit the change that can be applied to a maximum value. In this case, the maximum is the difference between

Algorithm 3 BAdaptiveLearning($i, DS_i, oneLevel$)

```
if $DS_i < 0$ then
    $HMinDS_i = min(HMinDS_i, DS_i)$
    $factorB_i = DS_i / HMinDS_i$
else
    if $DS_i > 0$ then
        $HMaxDS_i = max(HMaxDS_i, DS_i)$
        $factorB_i = DS_i / HMaxDS_i$
    else
        $factorB_i = 0$
    end if
end if
$\delta B_i = oneLevel \times factorB_i$
return $|\delta B_i|$
```

Algorithm 4 VAdaptiveLearning($i, PS_i, NPS_i, oneLevel$)

```
$differV_i = |PS_i - NPS_i|$
$HMinPS_i = min(HMinPS_i, PS_i)$
$HMinNPS_i = min(HMinNPS_i, NPS_i)$
if $NPS_i < PS_i$ then
    $factorV_i = differV_i / HMinPS_i$
else
    if $NPS_i > PS_i$ then
        $factorV_i = differV_i / HMinNPS_i$
    else
        $factorV_i = 0$
    end if
end if
$\delta V_i = oneLevel \times factorV_i$
return $|\delta V_i|$
```

levels as in Axelrod's original model, of $\frac{1}{7}$, which we make a constant, *oneLevel*. Thus, an agent modifies its boldness by δB_i, which is calculated by multiplying $factorB$ with the maximum change of $oneLevel$, so that it can maximally change its boldness by one level (or by $\frac{1}{7}$) when $factorB$ is 1.

Second, with relation to vengefulness (as in the VAdaptiveLearning function of Algorithm 4), agents keep track of two historical variables: $HMinPS_i$ is the minimum obtained punishment score in i's history of interactions; and $HMinNPS_i$ is the minimum obtained non punishment score in i's history of interactions. Having the current obtained punishment score PS_i and non punishment score NPS_i, agents update the historical variables accordingly. Then, $factorV$, which determines the change that should be made to an agent i's vengefulness, is calculated based on division of $differV_i$ (the difference between PS_i and NPS_i) by $HMinPS_i$ if PS_i is better than NPS_i or on the division of $differV_i$ by $HMinNPS_i$ if PS_i is worse than NPS_i.

The very first change that to an agent's policy is always the maximum possible change, because there is nothing in the history, and the current scores will be considered as maximum or minimum scores for later steps. However, these maximum or minimum scores can change if the agent obtains better or worse scores in later interactions.

C. Adaptive Punishment

1) Adaptive Punishment and Static Punishment: In Axelrod's original model, punishment is static and determined at design time, so that all norm violators receive punishment with the same magnitude, implying a constant cost also for the punisher.

As agents are rational and they seek to maximise their utility, the reinforcement learning approach presented previously fits well in the model. However, the original vision of punishment presented by Axelrod is inefficient for both parties involved in this self-governance activity, and we propose an improvement through an adaptive approach. In Axelrod's original model, the punishment applied to defectors was fixed off-line. This eases the computational costs involved, but lacks efficacy, since it does not consider the degree of violation nor the frequency of violation. Both of these two factors should be considered in order to specify the appropriate punishment for defectors.

Our variation of the model allows the possibility of managing the existence of different types of defectors, classified not only by the intensity of their instantaneous defection (producing a higher defection in one interaction) but also by the frequency of defections (defection during multiple timesteps). These different types of defections must be responded to with appropriate values of punishment. This adaptive adaptation of punishment results in an optimisation process that positively affects both parties: (1) the punisher, as it reduces the costs associated with punishment to the minimum functional value, and (2) the punished, as it only receives the necessary amount of punishment to be deterred from future violations.[3]

For example, in the case of peer-to-peer (P2P) file sharing, agents are able to download files from each other, with the norm that these agents must upload the files they have downloaded in order to share them with others and to maintain their availability on the network. However, since uploading consumes bandwidth, the absence of an appropriate punishment can lead self-interested agents to choose not to share files and preserve their bandwidth. In the case of frequent occurrences of such selfish behaviour, the efficiency of the whole P2P network is threatened. As a possible solution, de Pinninck et al. [22] suggest that a suitable punishment for such behaviour is to ostracise norm-violators. Here, agents that violate the norm are blocked for the same time period (as determined by the system designer) for each occurrence of the violation, and the challenge is to determine the most appropriate blocking period.

For example, while a long blocking period may cause some agents to cease defection, others may do so with only a very short blocking period, yet if the longer period is used, then the network loses the participation of the latter for more

[3] It has been shown elsewhere that larger punishments than necessary can have a detrimental effect on cooperation.[21]

time than necessary. Conversely, the longer period may not be sufficient to cause still more agents not to defect if the utility gain from violation is much greater than the loss from being blocked. Identifying the appropriate blocking period can thus be crucial to the performance of the system, but is in general impossible, simply because no single fixed blocking period can be effective for all agents. In contrast, adaptive punishment adapts punishment to suit the circumstances; it can be increased when insufficient and decreased when excessive. Agents that do not change their behaviour after punishments can thus be punished for longer on subsequent violations, until they start to comply, while agents that start to comply occasionally (even if still violating) can have their punishments decreased. In this way, the punishment need not be fixed at design time, giving a more flexible mechanism to deal with different types of agents according to their behaviour.

2) Experience-Based Adaptive Punishment: As discussed earlier, agents adapt their behaviour according to a utility-maximising policy, where the adaptation is proportional to the rewards associated with each action. Agents therefore need to be provided with a punishment-optimising mechanism whose task is twofold: (1) calculate the appropriate punishment to deter a defector from future violations; and (2) lower the cost for the punisher, because of the proportionality relationship between the cost of punishment and its damage (by allowing the punisher to adapt the intensity of punishment to be applied, the cost associated with it adapts consequently).

Since agents can observe the degree of an instantaneous defection, in order to calculate the appropriate punishment, an agent needs to consider the past behaviour of the specific violator. To achieve this, the identity and actions of the various other interacting agents in the environment must be recorded. Now, an agent's memory is limited to a particular window size so that only the most recent interactions are recorded, and an agent whose behaviour changes is not punished severely just because of defection in the distant past. Then, if an agent continues to defect regularly, any new punishment should be stronger than the previous one. Similarly, a generally compliant agent that only recently defected should be punished less that an agent that regularly defects, to avoid using unnecessary power.

In relation to a particular agent j, there are two main values that must be stored: the number of previous instances of defection of agent j (nd_j), and the number of previous instances of compliance of agent j (nc_j), both in the context of the window size. From these values we obtain the *defection proportion* (dp_j), representing the percentage of defections compared to the total number of decisions, by dividing nd_j by the total of nd_j and nc_j, as follows.

$$dp_j = \frac{nd_j}{nd_j + nc_j} \tag{1}$$

However, the absolute number of defections itself has an additional effect on punishment, since an agent that violates a norm 10 times is more determined than an agent that violates it just once. An agent that violates the norm 1 out of 10 times should be punished less than an agent that violates the norms 10 out of 100 times. This is represented in what we call the local defection view of agent i on agent j, and is specified as follows:

$$LocalView : AGENT \times AGENT \rightarrow \mathbb{R} \tag{2}$$

with:

$$\forall ag_i, ag_j \in AGENT : LocalView(ag_i, ag_j) = dp_j \times nd_j$$

where:

- ag_i is the punishing agent;
- ag_j is the defecting agent;
- dp_j is the defection proportion of agent j in agent i's memory; and
- nd_j is the number of defections of agent j in agent i's memory.

In order to allow agents to apply punishment with the appropriate intensity, punishment needs to change according to the defector's previous history, optimising its cost. However, an initial punishment value is needed as a base that is adapted depending on the type of defection. This punishment unit (pu) is used to determine the punishment value, by multiplying it by the defection proportion and the absolute number of defections. Punishment is thus a function that takes two agents and returns the punishment value applied by the first agent to the second, as follows:

$$ExpPunish : AGENT \times AGENT \rightarrow \mathbb{R} \tag{3}$$

with:

$$\begin{aligned}&\forall ag_i, ag_j \in AGENT,\\ &\quad ExpPunish(ag_i, ag_j) = LocalView(ag_i, ag_j) \times pu\end{aligned}$$

The value of metapunishment is calculated similarly, with the number of defections representing the number of instances of sparing defectors, and the number of instances of compliance representing the number of instances of punishing defectors.

The cost of punishment is fixed to 1 unit for punishers reducing the utility of violators by 4 units (1:4 punishment technology is used because it has been shown [23] to be more effective in promoting cooperation).

III. The Metanorm Model and One-to-One Interactions

As discussed above, Axelrod's original model does not capture the dyadic interactions that are essential for any P2P-based environment. In order to enable its application to such environments, the model needs to be modified so

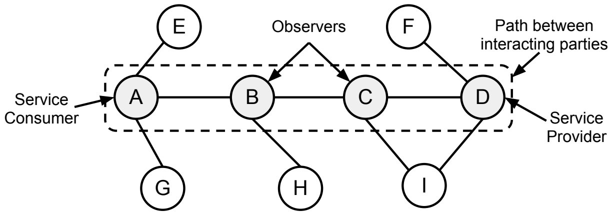

Figure 1. A one-to-one interaction between A and D

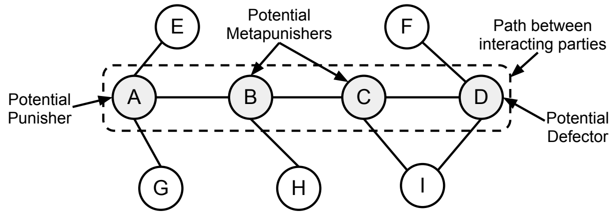

Figure 2. Potential punishments in a one-to-one interaction between A and D

that it enables emergence of norms when the fundamental principles of interaction are changed. In this section, we explain how the model is modified in support of this and the results that the new model brings.

A. One-to-One Interactions

Unlike the original model, agents in P2P environments must be able to interact with any other agent in the network, yet managing such location and connection between individuals is challenging. When looking for particular services, for example, consumers can ask central registries for information about who can provide them with the service they seek, information that can also be obtained by mobilising neighbours in their social network. Indeed, networks are an effective way to obtain information — for example provided through word-of-mouth — and represent an alternative source, with respect to traditional methods. While conventional approaches in multi-agent systems, such as *registries* or *matchmakers*, partially address this problem [24], in highly dynamic environments, there is a valuable amount of information that cannot be stored in centralised repositories. In some cases, much of this information (such as up-to-date details about the quality of service or the availability of the service) may be accessed only by using social networks of interaction. In this paper, a hybrid approach is proposed: similarly to the *white pages* of UDDI [25], our agents query a central server to obtain pointers to service providers, and then all other important information about the service is provided by the service providers themselves. The operation of the system is similar to that implemented by Napster.

This can be seen as an agent asking any other agent about a specific file in a peer-to-peer file sharing system. Since the two interacting agents might not be directly connected (they are not neighbours), the interaction takes place through a route that links the two agents but involves various other agents along the way, each of which is capable of observing all communications involved in the interaction. Such a scenario is illustrated in Figure 1, which shows an interaction between two agents A and D, where agent A is the service consumer requesting a service from agent D. The dashed line represents the interaction that is taking place through a path involving both agents B and C, which are the observers.

With regard to punishment decisions, the changes to the interaction protocol and the observability of interaction affect some decisions, especially in relation to punishment and metapunishment. First, punishment is only applied by the party that is requesting the service if the service provider defects and does not actually provide the service. This is because the service consumer is the only agent capable of punishing since it is the only agent that is directly affected by the defection. However, other observers record the defection in order to be able to make relevant decisions about subsequent interactions in the future. Second, and as a result of the aim of incentivising agents to respond appropriately to defections, any agent that observes the consumer not punishing the provider for defecting is able to metapunish the consumer. Figure 2 shows the example introduced in Figure 1, but now in relation to punishment and metapunishment. Since agent D is the service provider for agent A, D is a potential defector (according to its likelihood of defection or *boldness*) towards A. As a result, A is a potential punisher (according to its likelihood of punishing or *vengefulness*) towards D. In addition, and because they are observers of the interaction, both agents B and C are potential metapunishers (again, according to their likelihood of punishing or vengefulness).

B. Experience-Based Adaptive Punishment Results

In light of these modifications, and for the model to reflect the one-to-one interaction scenario, a set of experiments was undertaken to show the effect of this new arrangement on the effectiveness of the model in achieving norm establishment (where norm establishment is defined as resulting in a situation in which there is a majority of agents with high vengefulness and low boldness). In these experiments, the population of agents consists of 1,000 agents whose initial boldness and vengefulness are generated by using a uniform distribution function. With regard to the underlying structure of the system, the focus of our work is on scale-free networks, since it is more representative of the domain of interest (peer-to-peer networks). Agents are thus located in a scale-free network (that represents theoretical social networks [26], [27]) generated using the *Barabási-Albert* (BA) model [28], with a starting value of the basic punishment unit being -1, which is presented with other parameter values as shown in Table I. A final remark is that since communication cost is irrelevant to the phenomenon

Table I
EXPERIMENTAL SET UP

Number of agents	1,000
Number of timesteps	1,000,000
Temptation value	3
Hurt value	-1
Enforcement cost	-2
oneLevel	$\frac{1}{7}$
γ	0.01
pu	-1
Memory Window Size	20

under investigation, it is assumed that communication that is required for agents to exchange information is free. Figure 3 illustrates the results obtained from 1,000 independent runs, where each point represents the final average boldness and vengefulness of the whole population. This shows that in all runs, the population ends with both high boldness and high vengefulness, which means that agents in this population defect frequently (since they have high boldness) and also punish and metapunish regularly (since they have high vengefulness).

Having analysed the results, it is clear that agents are still defecting, despite the punishments applied, because these punishments do not exceed the utility gain that agents receive from defecting (via the temptation value), and are thus not effective. This is because a punishing agent has insufficient experience with the defecting agent due to limited chances of frequent interaction and the limited memory window size of each agent, preventing the punisher from determining an appropriate punishment to apply to the defector. However, because metapunishment is possible by multiple observers, it guarantees that the level of vengefulness is still high enough for punishment and metapunishment to take place.

This explanation is supported by Figure 4, which shows the total punishment applied in each round, and the total temptation gained in the same round. (Note that, for clarity, we have plotted the results of only the first 50 rounds, even though there are actually many more.) The figure shows that the total temptation (utility agents gain from defecting) is significantly higher than the punishment (utility agents lose from defecting), reinforcing our claim that punishment is ineffective.

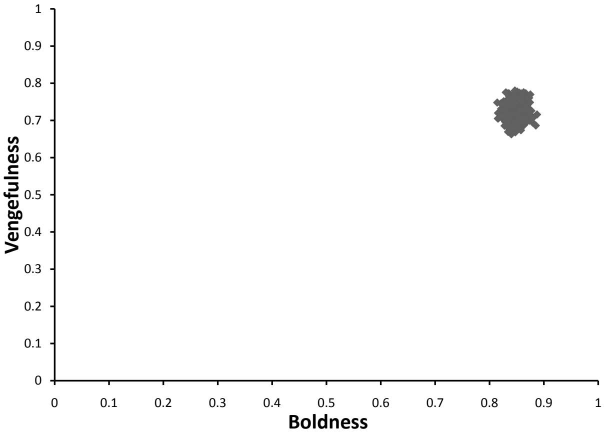

Figure 3. Results for 1,000 agents in a scale-free network

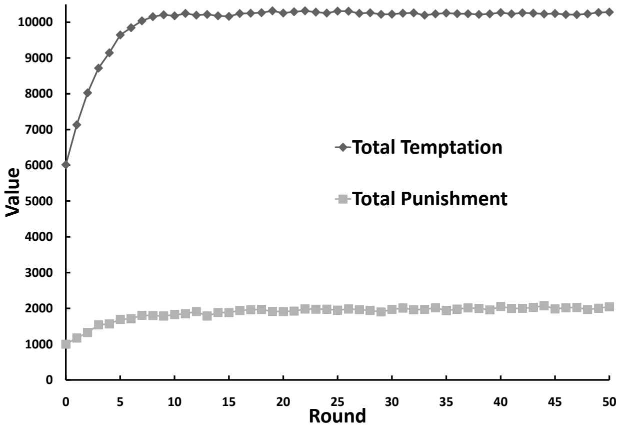

Figure 4. A Comparison between Temptation and Punishment Levels in Each Round

IV. REPUTATION-LIKE TECHNIQUE

Unlike Axelrod's original model in which all agents are connected, and all can observe and thus punish others, this new model provides observers only along the path of interaction. As shown above, in this constrained context, experience-based adaptive punishment alone is not adequate to achieve norm establishment due to the limited experience of agents with each other. To address this, agents need more information about those involved in the interactions. They can seek such information from other agents that are known to have more experience with the relevant agent, in a fashion that can be seen as a form of *reputation* for that agent. Agents can then make use of this reputation to determine a more appropriate punishment decision.

Clearly this relies on agents providing truthful reports of the behaviour of others; we could argue that it is in the population's self interest to establish the norm, so that agents are intrinsically motivated to provide reliable information and not lie. However, it is out of the scope of this paper to investigate the effects of non-reliable (cheating) agents. In what follows, therefore, we outline the simple mechanism by which an agent establishes reputation for use in determining appropriate punishments. The key point to note is that this is intended not as a sophisticated contribution to work on reputation, but as an illustration of how reputation (even in a very simple form) can help to support regulation.

A. Reputation Model

The basic idea of the reputation model is that agents aggregate the information they obtain from others with the information that they already have as a result of their own individual experience. Thus, if agent A decides to punish agent D for defecting, it first sends a request to all agents along the path of interaction, asking for information about D. Then, all other agents (B and C in our example) calculate the defection proportion of D and send it back to A. Having received these different values, A then aggregates them with its own assessment of defection proportion to form a total defection proportion. The aim is to augment the rather limited *local view* of defection for D with a *global view* that takes into account a broader range of experience to give a more appropriate punishment.

Now, since the method of calculating the local view of defection has already been introduced in Section II-C2, we here introduce the $GlobalView$ that returns the sum of local views of all agents in the path of interaction (excluding the two main interacting parties), and dividing this by the number of agents on the path (to maintain a value between 0 and 1). Here, $PATH$ is the set of all possible paths (essentially sets of agents along those paths) in the network.

$$GlobalView : AGENT \times PATH \to \mathbb{R} \tag{4}$$

with:

$$\forall ag_i, ag_j \in AGENT, p \in PATH, ag_k \in p, ag_i, ag_j \notin p :$$
$$GlobalView(ag_j, p) = \frac{\sum_{k=1}^{|p|} LocalView(ag_k, ag_j)}{|p|}$$

where:

- ag_i is the punishing agent;
- ag_j is the defecting agent;
- p is the the path of interaction involving all other agents;
- ag_k is an observing agent that belong to P
- $LocalView(ag_k, ag_j)$ is the local view by agent k of defection of agent j (calculated using the function defined in formula 2); and
- pu is the basic punishment unit used in the model.

This global view can now be used to specify the method of determining the *total* defection view, which incorporates both the local and global view of the defecting agent and can be calculated as follows.

$$TotalView : AGENT \times AGENT \times PATH \to \mathbb{R} \tag{5}$$

with:

$$\forall ag_i, ag_j \in AGENT, p \in PATH :$$

$$TotalView(ag_i, ag_j, p) = \frac{LocalView(ag_i, ag_j) + GlobalView(ag_j, p)}{2}$$

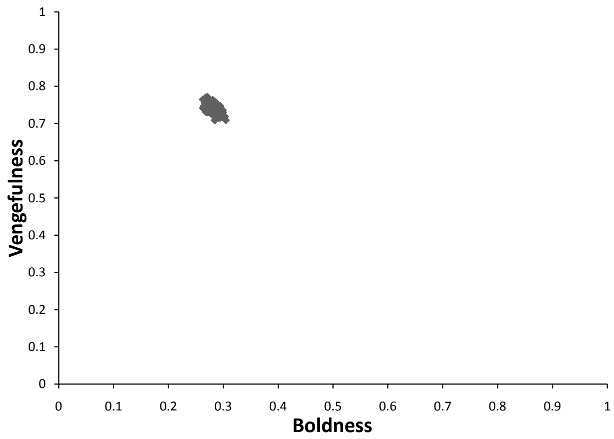

Figure 5. Reputation-based results for 1,000 agents in a scale-free network

B. Reputation-Based Adaptive Punishment

In this reputation-based punishment technique, the value of punishment is calculated similarly to experience-based punishment, but with the replacement of the local defection view with the total defection view, as follows.

$$RepExpPunish : AGENT \times AGENT \to \mathbb{R} \tag{6}$$

with:

$$\forall ag_i, ag_j \in AGENT :$$
$$RepExpPunish(ag_i, ag_j) = TotalView(ag_i, ag_j) \times pu$$

C. Results

Given this new reputation-based model, we ran experiments to show the effect of the technique on the defection rate. These experiments were set up like those introduced earlier (using the parameters in Table I), and provided results indicating the successful application of levels of punishment appropriate to establishing the norm. A sample result of one experiment involving 1,000 runs is shown in Figure 5, in which there is a noticeable improvement in the results. First, the population still has a high level of vengefulness, which means that punishment and metapunishment are active. Second, and most importantly, the level of boldness has dropped significantly, because agents that are faced with a defecting agent can gather much more information about this agent. As a result, their punishment can be much more appropriate in limiting the opportunities for this agent to defect in the future.

As before, Figure 6 shows the total temptation and punishment in each round. This time, however, the results indicate that punishment does indeed exceed temptation, suggesting that punishment is much more appropriate in preventing agents from defecting. Interestingly, the punishment values vary significantly over time, also indicating that they are continually adjusted in line with experience to give the level most suitable for the circumstances.

V. Related Work

Even though Axelrod's simulation model is limited, he provides a good and valuable explanation for the emergence of cooperation and the stability of punishment. Since then however, many others [8], [2], [11] have been concerned about the evolution of altruistic punishment, and some authors in particular have empirically shown that the existence of punishment allows for the emergence and stability of cooperative strategies within human populations. Indeed, in the last decade, an important body of work concerned with mutiagent systems and punishment has developed, analysing all aspects related to the regulation of normative behaviour [29].

Prior work in this area has mainly addressed the use of different forms of punishment in order to obtain the desired system behaviour. For example, de Pinninck et al. [30] takes the use of reputation to its most extreme, by allowing agents to definitively remove interactions with norm-violators. Here, ostracism leads to satisfactory results in the presented P2P example, but this approach suffers from the weakness that the norm-violators lose all possibility of interaction, and are not allowed to adapt and alter their behaviour after punishment. In the context of applying a punishment to alter the behaviour of the punished agent, Villatoro et al. [9] introduces a simple heuristic for adaptive punishment in a prisoner's dilemma setting. This adaptive punishment approach obtains good results but involves adaptation time.

In contrast, the contributions of this paper are distinct in providing a monetary punishment that alters the behaviour of agents; this punishment is dynamically calculated and adapted for each specific agent, based on its previous behaviour. The information required to do so is either obtained through direct interactions or communicated by other peers; by allowing this communication, agents avoid the gap of the adaptation found in [9], providing the most suitable punishment instantly.

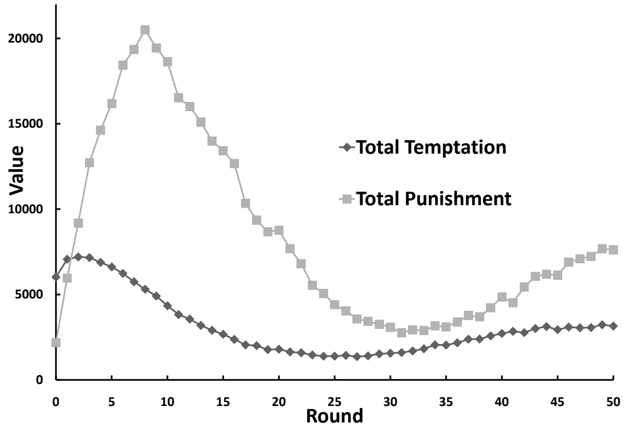

Figure 6. A Comparison between Temptation and Punishment Levels in Each Round

VI. Conclusion and Future Work

Norm emergence is an important and valuable phenomenon that has applications to self-organising systems such as peer-to-peer networks and wireless sensor networks in which there is no interference from any central or outside authority. While there has been much work on this phenomenon (as discussed earlier), punishment has generally been considered to be static (though with some exceptions). In response, our work in this paper focusses on adaptive punishment and its use for achieving norm establishment. To this end, we used a previously adapted version of Axelrod's metanorm model, and investigated the effect of adaptive punishment on establishing the norm in the context of limited observability.

In particular, our results show that experience-based adaptive punishment fails to stop agents defecting when observation is limited. This is due to agents not having enough information about each other, so they are not able to estimate efficient punishments. However, by introducing reputation into the model to provide extra information, this does not remain the case. Reputation enriches agents' information, and allows them to determine appropriate punishment decisions that are able to regulate agent behaviour and prevent them from defecting. While our reputation model is very simple, it is perfectly adequate for our aim of investigating the use of reputation in building an adaptive punishment mechanism. Having seen that reputation can be valuable in the development of such a mechanism, our future work will focus on investigating the use of more complex reputation models and their effect on improving the efficiency of adaptive punishment even further.

References

[1] R. Boyd, H. Gintis, S. Bowles, and P. J. Richerson, "The evolution of altruistic punishment," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 6, pp. 3531–3535, 2003.

[2] N. Nikiforakis, "Punishment and counter-punishment in public good games: Can we really govern ourselves?" *Journal of Public Economics*, vol. 92, pp. 91–112, 2008.

[3] J. M. Epstein, "Learning to be thoughtless: Social norms and individual computation," *Comput. Econ.*, vol. 18, no. 1, pp. 9–24, Aug. 2001.

[4] F. Flentge, D. Polani, and T. Uthmann, "Modelling the emergence of possession norms using memes," *Journal of Artificial Societies and Social Simulation*, vol. 4, no. 4, 2001.

[5] B. T. R. Savarimuthu, M. Purvis, and M. Purvis, “Social norm emergence in virtual agent societies,” in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, ser. AAMAS ’08. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1521–1524.

[6] T. Yamashita, K. Izumi, and K. Kurumatani, “An investigation into the use of group dynamics for solving social dilemmas,” in *Multi-Agent and Multi-Agent-Based Simulation*, ser. Lecture Notes in Computer Science, P. Davidsson, B. Logan, and K. Takadama, Eds., vol. 3415. Springer Berlin / Heidelberg, 2005, pp. 185–194.

[7] D. Villatoro, S. Sen, and J. Sabater-Mir, “Topology and memory effect on convention emergence,” in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02*, ser. WI-IAT ’09. IEEE Computer Society, 2009, pp. 233–240.

[8] E. Fehr and S. Gachter, “Altruistic punishment in humans,” *Nature*, vol. 415, pp. 137–140, 2002.

[9] D. Villatoro, G. Andrighetto, J. Sabater-Mir, and R. Conte, “Dynamic sanctioning for robust and cost-efficient norm compliance,” in *IJCAI*, T. Walsh, Ed. IJCAI/AAAI, 2011, pp. 414–419.

[10] F. Giardini, G. Andrighetto, and R. Conte, “A cognitive model of punishment,” in *Proceedings of the 32nd Annual Conference of the Cognitive Science Society Austin, TX: Cognitive Science Society*, S. O. . R. Catrambone, Ed. Portland, Oregon, 2010, pp. 1282–1288.

[11] D. Helbing, A. Szolnoki, M. Perc, and G. Szab, “Punish, but not too hard: how costly punishment spreads in the spatial public goods game,” *New Journal of Physics*, vol. 12, no. 8, p. 083005, 2010.

[12] R. Axelrod, “An evolutionary approach to norms,” *American Political Science Review*, vol. 80, no. 4, pp. 1095–1111, 1986.

[13] S. Mahmoud, J. Keppens, M. Luck, and N. Griffiths, “Efficient norm emergence through experiential dynamic punishment (to appear),” in *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, 2012.

[14] S. Mahmoud, J. Keppens, M. Luck, and N. Griffiths, “Norm establishment via metanorms in network topologies,” in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03*, ser. WI-IAT ’11. IEEE Computer Society, 2011, pp. 25–28.

[15] S. Mahmoud, J. Keppens, M. Luck, and N. Griffiths, “An analysis of norm emergence in axelrods model,” in *NorMAS’10: Proceedings of the Fifth International Workshop on Normative Multi-Agent Systems*. AISB, 2010.

[16] J. M. Galan and L. R. Izquierdo, “Appearances can be deceiving: Lessons learned re-implementing Axelrod’s evolutionary approach to norms,” *Journal of Artificial Societies and Social Simulation*, vol. 8, no. 3, 2005.

[17] S. Mahmoud, J. Keppens, M. Luck, and N. Griffiths, “Overcoming omniscience in axelrod’s model,” in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03*, ser. WI-IAT ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 29–32.

[18] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[19] M. Bowling and M. Veloso, “Rational and convergent learning in stochastic games,” in *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, ser. IJCAI’01. Morgan Kaufmann Publishers Inc., 2001, pp. 1021–1026.

[20] S. Mahmoud, J. Keppens, M. Luck, and N. Griffiths, “Norm emergence: Overcoming hub effects in scale free networks,” in *Proceedings of the AAMAS 2012 Workshop on Coordination, Organizations, Institutions and Norms*, 2012.

[21] B. Rockenbach and E. Fehr, “Detrimental effects of sanctions on human altruism,” *Nature*, vol. 422, pp. 137–140, 2003.

[22] A. Perreau De Pinninck, C. Sierra, and M. Schorlemmer, “Distributed Norm Enforcement: Ostracism in Open MultiAgent Systems,” *Computable Models of the Law: Languages, Dialogues, Games, Ontologies*, pp. 275–290, 2008.

[23] N. Nikiforakis and H.-T. Normann, “A comparative statics analysis of punishment in public-good experiments,” *Experimental Economics*, vol. 11, no. 4, pp. 358–369, 2008.

[24] K. Decker, K. P. Sycara, and M. Williamson, “Middle-agents for the internet,” in *IJCAI (1)*, 1997, pp. 578–583.

[25] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, “Unraveling the web services web: An introduction to soap, wsdl, and uddi,” *IEEE Internet Computing*, vol. 6, pp. 86–93, 2002.

[26] M. E. J. Newman, “The structure and function of complex networks,” *SIAM REVIEW*, vol. 45, pp. 167–256, 2003.

[27] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Review of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.

[28] A.-L. Barabási and R. Albert, “Emergence of Scaling in Random Networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[29] D. Grossi, H. Aldewereld, and F. Dignum, “Ubi lex, ibi poena: Designing norm enforcement in e-institutions,” in *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, M. Dignum, N. Fornara, and E. Matson, Eds. Springer, 2007, pp. 101–114.

[30] A. P. de Pinninck, C. Sierra, and M. Schorlemmer, “Friends no more: norm enforcement in multiagent systems,” in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS ’07. ACM, 2007, pp. 92:1–92:3.