Cooperation Emergence under Resource-Constrained Peer Punishment

Samhar Mahmoud, Simon Miles, Michael Luck
King's College London, London, UK
{samhar.mahmoud, simon.miles, michael.luck}@kcl.ac.uk

ABSTRACT

In distributed computational systems with no central authority, social norms have shown great potential in regulating the behaviour of self-interested agents, due to their distributed cost. In this context, peer punishment has been an important instrument in enabling social norms to emerge, and such punishment is usually assigned a certain enforcement cost that is paid by agents applying it. However, models that investigate the use of punishment as a mechanism to allow social norms to emerge usually assume that unlimited resources are available to agents to cope with the resulting enforcement costs, yet this assumption may not hold in real world computational systems, since resources are typically limited and thus need to be used optimally. In this paper, we use a modified version of the metanorm model originally proposed by Axelrod [1] to investigate this, and show that it allows norm emergence only in limited cases under bounded resources. In response, we propose a resource-aware adaptive punishment technique to address this limitation, and give an experimental evaluation of the new technique that shows it enables norm establishment under limited resources.

Keywords

Metanorm, Emergence, Limited Enforcement Cost

1. INTRODUCTION

In many application domains, engineers of distributed systems may choose, or be required, to adopt an architecture in which there is no central authority, and the overall system consists solely of self-interested autonomous agents. The rationale for doing so can range from efficiency reasons to privacy requirements. In order for such systems to achieve their objectives, it may nevertheless be necessary for the behaviour of the constituent agents to be cooperative. In peer-to-peer file sharing networks, for example, it is required that (at least a proportion of) peers provide files in response to the requests of others, while in wireless sensor networks nodes must share information with others for the system to determine global properties of the environment. However, there is typically a temptation in such settings for individuals to deviate from the desired behaviour, which is known as the problem of free riding behaviour. For example, to save bandwidth, peers may choose not to provide files, and to conserve energy, the nodes in a sensor network may choose not to share information. Therefore, some form of mechanism is needed to outweigh such temptations and to encourage cooperation among self interested agents.

Norms that are imposed and monitored by central authorities, have been proposed by many (e.g., [3, 4, 8, 15, 19]) as a valuable mechanism for regulating or constraining the behaviour of self-interested agents. However, in virtual environments, interactions can be of high magnitude and speed, and thus their regulation is expensive, and may even be infeasible. Social norms offer a means to provide distributed mechanisms for the self-regulation of virtual systems and societies, by delegating to the population itself the responsibility to impose appropriate behavioural standards [6, 28].

In this context, many have been concerned with the development of mechanisms to ensure the emergence of such social norms (e.g., [9, 12, 30, 32, 34]). In particular, researchers from many scientific areas have considered *punishment* as a key motivating element for norms to be established [10, 14, 17, 31]. Here, punishment is a monetary incentive, typically incurring an enforcement cost for the punisher, but bringing a potential benefit to the population as a whole when correctly applied. Work that investigates the use of punishment as a means for social norms to emerge has assumed that agents applying such punishment have unlimited resources, allowing them to bear the resulting enforcement cost. This assumption is significant in real world settings in which resources are limited and require more careful exploitation. For example, sensors in wireless networks have limited energy and thus need to optimise their use of it.

In response, this paper[1] seeks to address such limitations by investigating the effect of scarce resources on norm emergence, based on which a resource-aware adaptive punishment mechanism is proposed and evaluated through experimental simulations[2]. For this purpose, we first integrate the constraint of limited resources within the metanorm model originally proposed by Axelrod [1] and adapted by Mahmoud et al. [23, 22]. The metanorm model has been shown to be capable of regulating distributed computational systems under various settings. Moreover, it is equipped with an

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

[1] A 2-page abstract on a small part of the work contained here appears elsewhere [25].

[2] This material is based upon work supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF under Award No. FA9550-15-1-0092.

adaptive punishment mechanism that we believe to be vital when dealing with limited resources. We then investigate the limitations of both static and adaptive punishment mechanisms of this model under limited resources constraints, before proposing our resource-aware adaptive punishment.

The remainder of this paper is organised as follows. Section 2 introduces related work, and Section 3 discusses the implications of limiting resources for both static and adaptive punishment mechanisms. Section 4 then builds on this with the integration of an enhanced adaptive punishment mechanism that takes limited resources into consideration and, through experimental analysis, shows that it succeeds in establishing a norm. Finally, the paper concludes in Section 5.

2. RELATED WORK

Achieving a particular desired behaviour among a population of autonomous individuals has received much attention. Many have been concerned with the evolution of altruistic punishment and its effect on the emergence and stability of cooperative strategies within human populations. Indeed, in the last decade, an important body of work concerned with multi-agent systems and punishment has developed, analysing all aspects related to the regulation of normative behaviour [16]. For example, Fehr et al. [11, 10] study the effect of distributed punishment in improving cooperation among self-interested agents. Their results show that heavy punishment succeeds in deterring free riding and promoting cooperation. However, Nikiforakis et al. [28] show that this does not hold when free riders are given the chance of counter-punishment. The threat of such counter-punishment weakens the desire to punish free riders, thus leading to less cooperation. To study these effects, Helbing et al. [17] show that the addition of punishing strategies to a classical public goods game, to allow rewarding cooperators or punishing defectors, increases cooperation among the entire population in the case of well-mixed interactions. Similarly, Savarimuthu et al. [30] show that peer-to-peer punishment is effective in achieving norm emergence in virtual on-line societies when the cost of punishing is low.

These previous models employ static punishment mechanisms in which punishment is a fixed value set at design time. In contrast, a mechanism for dynamically determining punishment values was proposed by Mahmoud et al. [21], using the prior experience of agents with their interaction partners in order to specify the appropriate level of punishment. However, this approach is dependent on repeated interactions between agents, which is not common in certain domains such as peer-to-peer file sharing. Miller et al. [26, 27] and Jurca et al. [18] suggest the use of explicit payment schemes as incentives to encourage honest reporting of information (such as reputation [18] and product feedback [27]) among an agent community. Specifically, the rating provided by an agent is assessed for trustworthiness by comparing it against those of other agents, and a financial reward is estimated (applying specific rules) and issued correspondingly if the agent is reporting the truth.

The above approaches rely on some form of punishment to deter malicious behaviour. However, they all assume that resources are unlimited in applying such punishment, yet this is an unrealistic assumption in many real world domains. In what follows, we investigate the effect of introducing the constraint of limited resources on the process of regulating self-interested agent behaviour using a version of perhaps the best known model in which punishment is a crucial aspect, Axelrod's metanorm model.

3. PEER PUNISHMENT AND LIMITED RESOURCES

Peer punishment has been widely used as a method to regulate the behaviour of participants in a distributed system. Such punishment usually carries a cost for the agent applying the punishment, known as an enforcement cost. Most existing models that study this phenomenon make use of *static* punishment, where there is an enforcement cost paid by the enforcer that is fixed at design time. Considering the free riding phenomenon in a peer-to-peer (P2P) file sharing system, de Pinninck et al. [7] suggest that the punishment for such behaviour should be *blocking*, by which all other agents cease interacting with an agent that is observed not to share files after downloading them. The blocking period can be for a specific length of time, which needs to be set at design time. In this case, the enforcement cost paid is the loss of access to files from the blocked agents, considering that prior to blocking them, free riders can still be sharing a small set of the file they downloaded, while withholding the majority of these files.

Determining an appropriate blocking period can be crucial to the performance of the overall system, especially those that rely on the participation of members for their functionality and effectiveness (as with P2P networks). Because of the dynamism of these systems and the cost incurred by both the agent that is applying the punishment and the agent that is being punished, choosing a single punishment value that is effective to be used against all agents may not be feasible. Therefore, mechanisms that support punishments whose value can be adapted at run-time are much better suited. This is *adaptive* punishment. Returning to our example, a blocking time of 30 minutes may be enough to deter the behaviour of agent i, but not agent j. The next time that agent j defects, a blocking time that is longer than 30 minutes can be used to try to force agent j to comply and start sharing files.

In real world distributed systems, resources can be crucial and optimising their use can be vital if these systems are to function effectively. The period duration for which peers in P2P systems can block other peers in the example above is limited, since they need to maintain a certain level of availability and interactions in the system.

As mentioned above, Axelrod's metanorm model is a strong candidate for use in analysing the effects of limited resources, since it has been shown to establish norms under various distributed systems settings. Moreover, in Mahmoud's variant, it employs both static and adaptive punishment mechanisms. In what follows, we first introduce the metanorm model, and then modify the model to consider the situation of limited resources. Finally, we provide an experimental evaluation to show the effect of these limited resources on norm establishment.

3.1 Metanorm Model

In Axelrod's metanorm model [1], a population of agents play a game in which each agent has to decide between cooperation and defection. The agent population evolves through a number of iterations, with a mechanism whereby successful behaviour (as measured by the scoring system) tends to be replicated and unsuccessful behaviour tends to be discarded. A major problem with Axelrod's model is due to the evolutionary approach adopted (as identified in [20, 13]). In consequence, this original approach has been replaced with a reinforcement learning algorithm that limits accessibility to global information, and instead allows agents to learn from their own experience [23]. Moreover, in order to capture a key feature of computational systems such as on-line virtual communities, Axelrod's classic model has been adapted by introducing a topological structure [22] that determines observability among agents, so that an agent's neighbours are the only witnesses of its interactions. This indicates that an agent only imposes punishment on its defecting neighbours, and metapunishment on its non-punishing neighbours.

This is the model we introduce next, which can be divided into four different parts: the interaction model, the agent model, the policy learning capability and the punishment mechanism.

3.1.1 Interaction Model

In the metanorm model, agents play a game iteratively. In each iteration, each agent must decide between cooperation and defection. Defection brings a reward for the defecting agent called a *temptation* value, and a penalty to all other agents called a *hurt value*. However, each defector risks being observed *by the other agents* 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 the *enforcement cost* and a high penalty for the punished agent known as the *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.

3.1.2 Agent Model

With regard to agent decision making, 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.

3.1.3 Policy Learning Capability

Having performed a set of actions in a particular round, agents are able to adapt their policies according to the positive or negative outcomes of these policies using a policy learning algorithm. In this algorithm, agents adapt their policies (boldness and vengefulness) at the end of each round of the simulation through a form of q-learning [33], 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.

However, agents do not adapt their policies in the same manner: 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 [5], 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 less.

3.1.4 Punishment Mechanism

There are two different punishment mechanisms that have been employed by the metanorm model: Axelrod's original static punishment mechanism [1], and Mahmoud et al.'s adaptive punishment mechanism [21]. Using static punishment, agents apply the same amount of punishment or metapunishment in every instance. This amount is fixed at design time. However, agents adapt their policies in a utility-maximising way, where the adaptation is proportional to the positive or negative utility associated with each action. Agents therefore are provided with an adaptive punishment 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).

In order to calculate the appropriate punishment, an agent needs to consider the past behaviour of the specific violator, which constitutes the *image* of this agent in terms of frequency of defection and cooperation. 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. Thus, 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, and waste resources.

3.2 Limiting Enforcement Resources

As mentioned previously, agents usually have limited resources that can be used for enforcement. Thus, once an agent is in a position to apply punishment to a violator, the punishment can only take place if sufficient resources exist to supplement the enforcement cost that can result from the punishment. The decision making mechanisms of agents in the metanorm model above therefore needs to be updated to take available resources into account before applying any form of punishment or metapunishment. We model this formally as follows.

First, an agent ag_i needs to be able to identify the amount of resources available as follows:

$$Res : AGENT \rightarrow \mathbb{R} \qquad (1)$$

where $\forall ag_i \in AGENT, Res(ag_i)$ is the amount of resource units available for agent ag_i, and $AGENT$ is the set of all agents.

In addition, ag_i needs to estimate the resources (enforcement cost) required to apply punishment p to ag_j:

$$EC : AGENT \times AGENT \times \mathbb{R} \rightarrow \mathbb{R} \qquad (2)$$

where $\forall ag_i, ag_j \in AGENT, p \in \mathbb{R} : EC(ag_i, ag_j, p)$ is the enforcement cost related to the current punishment's instance.

Based on this, an agent's decision to apply a punishment can be achieved using the $CanPunish$ function:.

$$CanPunish : AGENT \rightarrow \{TRUE, FALSE\} \qquad (3)$$

such that:

$$\forall ag_i \in AGENT :$$

$$CanPunish(ag_i) = \begin{cases} TRUE & \text{if } Res(ag_i) \geq EC(ag_i) \\ FALSE & \text{if } Res(ag_i) < EC(ag_i) \end{cases}$$

The above function can be used for the purpose of both punishment and metapunishment since they both require enforcement costs. Here $TRUE$ means that the agent can punish, while $FALSE$ means punishment is not possible.

Having verified that a punishment is possible, agent ag_i can punish the defecting agent ag_j, which results in the resources of agent ag_i decreasing by the relevant enforcement cost. This affects future decisions that ag_i can make with regard to punishment and metapunishment.

In addition, it is worth mentioning that resources considered here are those that are limited over a particular period of time. This means that once a particular agent spends all of its resources, it cannot apply any form of punishment or metapunishment until the restriction period has passed, after which resources are renewed. The model is round based, and in every round each agent is given multiple opportunities to defect. Each defection generates many punishment and metapunishment decisions, each of which consume resources. Therefore, we assume that resources are renewed every round for each agent, which needs to optimise the use of such resources within each round.

In what follows, using experiments, we show the effect of this new limited resources restriction on norm establishment using the metanorm model.

3.3 Experimental Evaluation

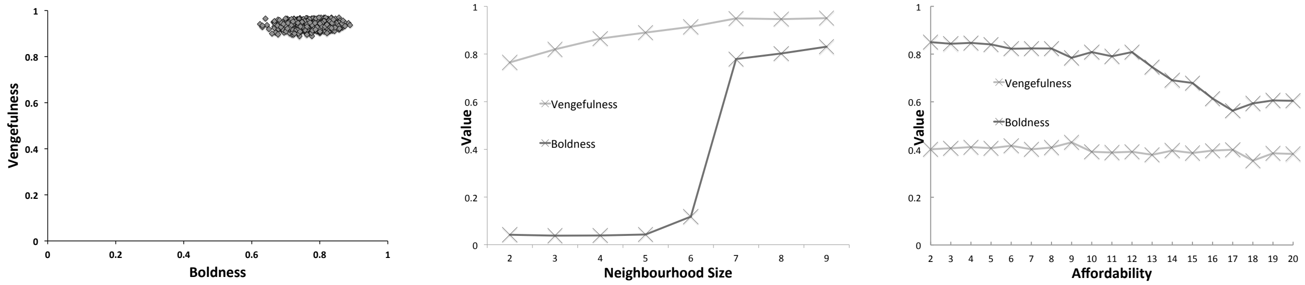
Before discussing experimental results, it is important to clarify the different possible results and what they show. The most desirable results are those with high vengefulness and low boldness, which are referred to as *norm establishment*. This is because low boldness means that agents defect rarely, and high vengefulness means that agents are generally willing to punish another agent that defects. Results where both low vengefulness and low boldness are observed are also good, because they indicate rare defections. However, with the absence of punishment, boldness tends to increase, causing a high defection rate, which is referred to as *norm collapse*. Other results involving midrange or high level of boldness are also referred to as norm collapse, since they involve a high number of defections. The evaluation of the different variations of the metanorm model introduced in this paper is based on how successful they are in bringing about norm establishment and avoiding norm collapse. It is also important to point out the effect of different decisions that an agent can make on norm establishment and collapse. Metapunishment is required for high vengefulness, without which the cost of punishment leads to low vengefulness due to the enforcement costs paid by the punishing agent. Low vengefulness leads to less punishment, which in turns leads to high boldness and norm collapse.

Topologies are an important component of this type of simulation, and they have different effects on norm establishment [22]. For the purposes of this paper, we use a lattice and a scale free topology for which norm establishment has been achieved previously. In a (one-dimensional) lattice with neighbourhood size n, agents are situated on a ring, with each agent connected to its neighbours n or fewer hops (lattice spacings) away, so that each agent is connected to exactly $2n$ other agents. Thus, in a lattice topology with $n = 1$, each agent has two neighbours and the network forms a ring. In a lattice topology with $n = 3$, each agent is connected to 6 neighbours. Such a topology has a regularity in the number of connections shared among all agents, which helps in studying the effects of the new setting in isolation of other network factors that can influence the results of the model [24]. In contrast, in a scale free topology, connections between nodes follow the power law distribution. Thus, few nodes (hubs) have a vast number of connections, but the majority have very few connections. These properties of scale-free networks suggest an imbalance in connections, which have been shown to affect norm establishment using the metanorm model [24]. In this section, we evaluate the ability of the metanorm model to establish a norm under limited resources. This is achieved using two sets of experiments: a first set where agents have static punishments with fixed enforcement costs, and a second set, in which agents can apply adaptive punishments with proportionate enforcement costs. The results of these experiments are described next, but first the parameter set-up is introduced.

Moreover, in many simulations that investigate the use of punishment, including the version of the metanorm model described above, agent selection is sequential. So the order in which the agents have opportunities to defect is the same every time. This is acceptable when resources are unlimited, since the order by which agents interact has no impact. However, with limited resources, a fixed selection mechanism means that agents spend all their resources after enforcing the same subset of agents (those earlier in the order) every time, and the remaining agents can escape punishment most of the time. Therefore, the simulation is updated to allow a random selection order of agents to take a decision about defecting, and a random selection order of agents that can apply punishment and metapunishment to those who defect. However, this random selection mechanism ensures that each neighbour observes the defection of every other neighbour, and observes the same neighbour sparing a defector from punishment.

3.3.1 Experimental Setup

The general parameter set-up used in the experiments conducted is presented in Table 1. The punishment cost and enforcement cost are those of the static punishment technique. In addition, we make sure to run the model on a relatively large population of agents, a large number of runs, and for a lengthy period per run to avoid obtaining misleading results.



(a) affordability of 12, lattice, $n = 8$ (b) affordability of 12, lattice, various n (c) various affordability, scale free

Figure 1: impact of limited resources with static punishment on final B and V

Table 1: Parameters Set-up

Term	Value
Boldness	Uniform distribution from 0 to 1
Vengefulness	Uniform distribution from 0 to 1
Number of opportunities to defect per round	4
Temptation to defect	$+3$
Hurt suffered by others	-1
Cost of being punished	-9
Cost of punishing	-2
Cost of being metapunished	-9
Cost of metapunishing	-2
Number of Agents	$1,000$
Number of Rounds	$1,000,000$
Number of Runs	$1,000$

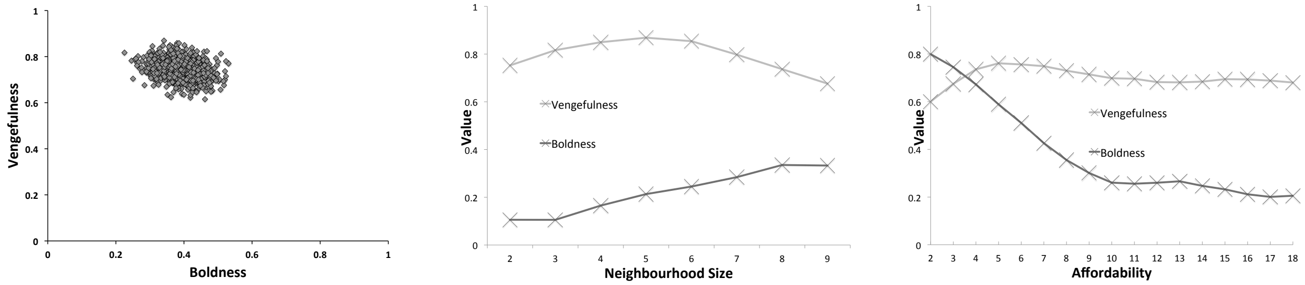
3.3.2 Static Punishment Experimental Results

We first evaluate the effects of limiting resources on the outcome of the metanorm model with static punishment. Axelrod originally suggested the proportion of (-9,-2) between punishment cost and enforcement cost, and our work is consistent with this. Moreover, Galan et al. [13] have experimentally determined the most effective proportions with static punishment, confirming that this is optimal. The interaction model of agents involves the following sequence of actions. For every defection opportunity that an agent i has, all of i's neighbours have a chance to punish i. If a neighbour j decides to spare i from punishment, then all of j's neighbours have the chance of metapunishing j. Assuming that every agent has 4 distinct neighbours, this means that for every single defection, 4 punishment decisions need to be taken, and if all these punishment decisions result in sparing the defector, $4 \times 4 = 16$ metapunishments can arise. Based on this, it is clear that agents will invest most of their resources on punishment and metapunishment as a result of the outcome of the first few defections, with scarce resources left to regulate the behaviour of the remaining agents.

The above explains the results obtained from using the metanorm model with constrained enforcement resources and static punishment, shown in Figure 1(a), where the diamonds represent the value of the mean average boldness and vengefulness of the final round of a particular run. In this experiment, each agent is provided with 12 resource units that are renewed every round. These results show that the model fails to establish the norm with the average boldness of agents remaining very high, and reflecting a very high rate of defection. The surprise here is that the average vengefulness is also high. Previous reported results of the metanorm model have shown that high vengefulness and high boldness is not a stable state for the population and usually leads to a norm establishment state with high vengefulness and low boldness. The case here is different because of the introduction of the limited resources. High vengefulness is due to two factors. First, for the first few occurrences of defection, sufficient resources remain available for metapunishment. Second, resources run out quickly, so no more enforcement costs are being paid by agents to cause vengefulness to drop. This last factor can also be used to explain the high boldness, with insufficient punishment taking place to deter defecting agents by outweighing the temptation gained.

This suggests that neighbourhood size plays a major role in the obtained results. Therefore, further experiments were conducted with varied neighbourhood size. The results reported in Figure 1(b) are for lattice topologies with neighbourhood sizes between 2 and 9, and limited resources of 12 units. Each point in the graph represents an average of $1,000$ runs with a particular neighbourhood size. It is clear from this that the effectiveness of the amount of resource available is limited by the number of neighbours. A static punishment mechanism with 12 units manages to establish the norm up to a neighbourhood size of 6, and fails after that. A similar outcome is found using other amounts of limited resources. For example, with limited resources of 6 units, norm establishment is observed up to a neighbourhood size of 3, and up to a neighbourhood size of 4 with limited resources of 8 units.

Previous analysis of the effect of scale free networks on the metanorm model [24] with unlimited resources has shown that hubs are instrumental to norm establishment. This is because of the vast number of connections that hubs have, which means they punish many other agents for defecting, and consequently pay a very high cumulative enforcement cost. To investigate the effect of limiting resources, we ran 1000 experiments on a scale-free network with 1000 agents, five of which were *hubs* (having a large number of connections) and the others (which we call *outliers*) having at least two connections to other agents in the population, and typically no more than four connections (according to Barabasi's



(a) affordability of 12, lattice, $n = 8$ (b) affordability of 12, lattice, various n (c) various affordability, scale free

Figure 2: impact of limited resources with adaptive punishment on final B and V

algorithm [2]). We repeated these experiments with various amounts of limited resources between 2 and 20, and the results are shown in Figure 1(c), where each point in the graph represents an average of $1,000$ runs with a particular amount of limited resources. The results indicate norm collapse, as all runs end with high boldness and midrange vengefulness. This is because hubs, which are the deriving agents of the population, run out of resources quickly, and do not allow boldness to drop. However, there are enough resources for outliers, which keep vengefulness at a midrange level. We can also see that at a level of resources of 13 upwards, boldness starts to drop to a midrange level. In fact, some experiments which are not shown here for clarity suggest that at least 50 units of resources are needed for the norm to be established with static punishment.

3.3.3 Adaptive Punishment Experimental Results

While static punishment does not always establish the norm under limited resources settings, it seems that adaptive punishment should do better, since the punishment is adapted according to the *image* of the agent under punishment or metapunishment. Thus resources should be used efficiently. The results shown in Figure 2(a) confirm this. In this experiment the punishment was factorised based on a basic punishment unit of -1, and the cost of punishment (enforcement cost) is set to 1 unit for punishers, reducing the utility of violators by 4 units (1:4 proportion is used because it has been shown [29] to be more effective in promoting cooperation). The results are better since the level of boldness drops to a midrange level. However, this does not reflect norm establishment, since a considerable number of defections still take place in every round. This is because adaptive punishment allows better use of available resources to regulate the behaviour of other agents, but resources are still not being made best use of, since the adaptive punishment mechanism in its current form depends only on the image of the current defector. Similar to the static punishment case, some experiments with different neighbourhood sizes were conducted, with the results of those with limited resources of 12 units shown in Figure 2(b). These results indicate that norm establishment is achieved with a relatively small number of neighbours, with this being weakened as the neighbourhood size is increased. Similar observations are obtained with other amounts of limited resources, too.

Norm establishment is also observed in scale free networks with limited resources of 10 upwards (Figure 2(c)). Below that, boldness starts to drop gradually, but not to a level at which the norm can be considered to be established. The results have clearly improved, which is due to more efficient use of resources to deal with defectors. However, this is not enough especially with hubs that are connected to too many agents with a high probability of frequent defection. So even when using the basic adaptive punishment, resources are still not optimally distributed. However, if we consider the available resources, or the remaining potential defectors that still need to be responded to, we may be able to help solve the problem, as we investigate next.

4. PUNISHMENT WITH CONSIDERATION OF LIMITED RESOURCES

The above approach does not take into account further defections that are yet to happen until resources are renewed, which explains the poor results. In this section, we introduce a new version of the adaptive punishment technique that is capable of allocating an appropriate amount of resources for the current enforcement action, taking into account the possibilities of future violations and the resources needed to deal with such violations.

In what follows, we first introduce a modified version of the adaptive punishment mechanism that takes into account resource limitations, followed by the results of an experimental evaluation of this new mechanism.

4.1 Resource-Aware Punishment Model

The basic idea of adaptive punishment is that each agent is supplied with a memory of limited size, in which they store information about their observations of actions taken by their direct neighbours. So, in the case of observing a defection, an agent stores the identity of the defecting agent together with the fact that this agent has defected, and the same in the case of cooperation. With regard to second level violation, an agent that spares a defector from punishment is also a defector, while one that punishes a defector is a cooperator. These facts are also stored in the memory, together with the identity of agents.

In order to allow agents to apply punishment with the appropriate intensity, punishment needs to change according to the defector's previous history. Based on this, and in relation to a particular *defecting* agent j, two main factors can be calculated: the number of previous instances of defection of agent j (dentoted by nd_j), and the number of previous instances of compliance of agent j (dentoted by nc_j), both in the context of the window size. From these values we obtain the *defection proportion* (dentoted by 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. This can be obtained as follows.

$$dp_j = \frac{nd_j}{nd_j + nc_j} \tag{4}$$

This is represented in what we call the *local image* of agent j from agent i prespective, and is specified as follows:

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

with $\forall ag_i, ag_j \in AGENT : LocalDefImage(ag_i, ag_j) = dp_j$ where:

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

However, as shown in Section 3.3.3, considering only the past experience of the agent under punishment is not sufficient when resources are limited. Therefore, agents need to take into account other agents that they need to deal with in the future, which are those that are observable and, according to the metanorm model, these are the direct neighbours specified by the interaction topology. This can be achieved by taking into account the average image with regard to defection of all agents in the neighbourhood, which can be calculated as follows.

First, the agent needs to obtain a set of neighbours by applying function NB:

$$NB : AGENT \to 2^{AGENT} \tag{6}$$

where

$$\forall ag_i \in AGENT : NB(ag_i) = \{ag_j | ag_j \text{ is connected to } ag_i\}$$

Having obtained the set of neighbours, an agent is capable of calculating the average image of those agents using the following function:

$$AvgDefImage : AGENT \times 2^{AGENT} \to \mathbb{R} \tag{7}$$

where

$$\forall ag_i \in AGENT \text{ and } NB(ag_i) \in 2^{AGENT} :$$
$$AvgDefImage(ag_i) = \frac{\sum_{ag_j \in NB(ag_i)} LocalDefImage(ag_i, ag_j)}{|NB(ag_i)|}$$

The comparison of the local image and the average image provides useful information on which to base the punishment decision. If the local image is greater than the average image, then the current defector has worse behaviour than most other agents in the neighbourhood, and needs more resources devoted to enforcing its behaviour. If the local image is less than average, then the current agent is not as bad as others in the neighbourhood and less resource can be used for its punishment. Based on this, we can estimate the deviation of the defecting agent's past behaviour from the behaviour pattern in the neighbourhood as follows:

$$Deviation : AGENT \times AGENT \to \mathbb{R} \tag{8}$$

where

$$\forall ag_i, ag_j \in AGENT :$$
$$Deviation(ag_i, ag_j) = \frac{LocalDefImage(ag_i, ag_j)}{AvgDefImage(ag_i)}$$

Thus, an agent can calculate a suitable amount of punishment that should be applied in this particular instance. But first an agent calculates a uniform amount of resources available for punishment based on equal distribution of resources. This is achieved by dividing the available resources by the number of neighbours, as follows:

$$UniformRes : AGENT \to \mathbb{R} \tag{9}$$

where

$$\forall ag_i \in AGENT : UniformRes(ag_i) = \frac{Res(ag_j)}{|NB(ag_i)|}$$

It is worth emphasising that the resources that are available to an agent are used as an enforcement cost, which is a particular enforcement cost percentage (ECP) of the applied punishment. The above function calculates the average resources available to be used for enforcement costs and not as a punishment value. Therefore, such a value needs to be converted into an equivalent punishment value as follows:

$$EquivPunish : AGENT \times \mathbb{R} \to \mathbb{R} \tag{10}$$

where

$$\forall ag_i \in AGENT \text{ and } ECP \in \mathbb{R} :$$
$$EquivPunish(ag_i, ECP) = UniformRes(ag_i) \times ECP$$

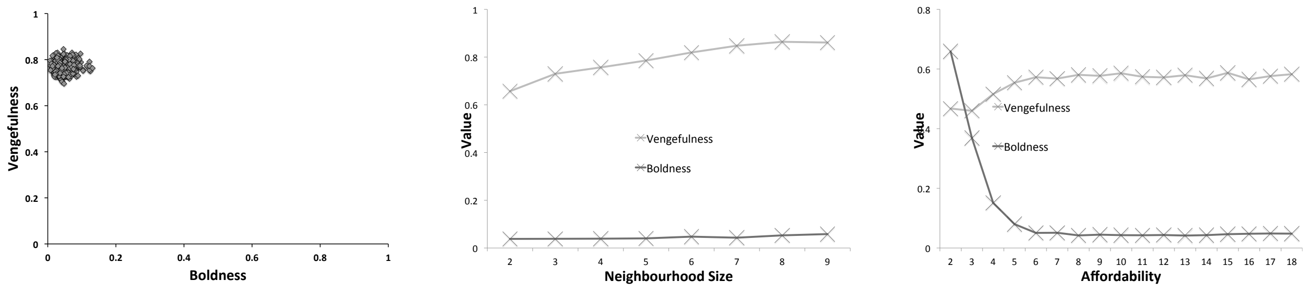
Having calculated the average punishment value, it needs to be adapted according to the deviation of the current defecting agent from the average defecting behaviour in the neighbourhood of the punishing agent. Also, an initial punishment value is needed as a base to be adapted depending on the type of defection. This punishment unit (pu) is used to determine the punishment value. Punishment is thus a function that takes two agents and returns the punishment value applied by the first agent to the second, as follows:

$$AdaptPunish : AGENT \times AGENT \to \mathbb{R} \tag{11}$$

where

$$\forall ag_i, ag_j \in AGENT, AdaptPunish(ag_i, ag_j) =$$
$$Deviation(ag_i, ag_j) \times EquivPunish(ag_i, ECP) \times pu$$

In summary, the value of the adaptive punishment in this case is a factor of the behaviour of the defecting agent with regard to defection, in comparison to the rest in the neighbourhood, the uniform resources available to deal with agents on an equal basis and an initial punishment unit. This means that if the agent at hand has a defection rate that is larger than average, then the uniform resources are scaled up to deal with this agent, and in the case that the agent defection rate is less than average, then the uniform resources are scaled down. 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.



(a) Affordability of 12, lattice, $n = 8$ (b) Affordability of 12, lattice, various n (c) various affordability, scale free

Figure 3: impact of limited resources with resource aware adaptive punishment on final B and V

4.2 Evaluation

With a similar parameter set-up to the experiments reported in Section 3.3, we ran experiments that involve the resource-aware adaptive punishment mechanism introduced in the previous section. The results of these experiments are shown in Figure 3(a), which indicate considerable improvements from the previous results, as norm establishment has been observed in all runs.

Even when the neighbourhood size is increased, norm establishment is achieved. This is due to the capability of the new approach to consider the number of neighbours that an agent may have to deal with, so punishment may be less even with more neighbours. However, the increased number of neighbours compensates for this. To simplify, assuming that 5 resource units available for enforcement, in the case of neighbourhood size of 5, an agent i will have 10 neighbours. If all of these neighbours are regular defectors, then the punishment that i applies to a defecting neighbour j is approximately $\frac{5}{10} = 0.5$, which is less than the gain that the j obtains from defecting (3). However, there are another 9 neighbours that can still punish j, and the combined punishment of these agents is enough to overcome the utility gained by j from defecting. This is illustrated by the results shown in Figure 3(b), where norm establishment is achieved with renewable resources of 12 units regardless of the neighbourhood size of the connection topology. In fact, larger neighbourhood sizes lead to better norm establishment due to the higher number of potential metapunishment opportunities that can be triggered by sparing a defector.

Much better results, reported in Figure 3(c), are also achieved over scale free networks. Even with limited resources (as low as 4), hubs are able to divide equally resources among the large number of agents they are connected to. Such resources, combined with the resources resulting from other outliers' punishment, are enough to cause norm establishment. Clearly, resources of 2 and 3 are not enough for the norm establishment to happen.

5. CONCLUSION AND FUTURE WORK

There have been many models proposed for norm emergence among groups of self interested agents. Punishment has been used as the core mechanism in most of these models. Such punishment is usually assigned a particular enforcement cost, and the general assumption is that unlimited resources are available for agents to cope with this cost. This paper has studied the effect of integrating a limited resource constraint within the well established metanorm model. Experimental results show that both the static and adaptive punishment mechanisms of the metanorm model fail to establish the norm with the absence of significant amount of resources. This is mainly because resources are not being used optimally.

In response, an enhanced adaptive punishment technique was proposed, which takes into account the amount of resources available to the agent and the number of punishment decisions that this agent may need to apply using such resources. The experimental evaluation showed that the new technique succeeds in establishing the norm with larger neighbourhood sizes than the static and original adaptive punishment mechanisms. Moreover, this new adaptation of the metanorm model allows designers of distributed computational systems to determine the amount of resources that is needed in order for the system to be appropriately regulated. As future work, we want to investigate the effect of the limited resources constraint on other norm emergence mechanisms in the literature.

REFERENCES

[1] R. Axelrod. An evolutionary approach to norms. *American Political Science Review*, 80(4):1095–1111, 1986.

[2] A. L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[3] G. Boella, L. Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12(2-3):71–79, Oct. 2006.

[4] M. Boman. Norms as constraints on real-time autonomous agent action. In *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent Rationality*, pages 36–44. Springer-Verlag, 1997.

[5] 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*, IJCAI'01, pages 1021–1026. Morgan Kaufmann Publishers Inc., 2001.

[6] 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*, 100(6):3531–3535, 2003.

[7] A. P. de Pinninck, C. Sierra, and M. Schorlemmer.

Distributed Norm Enforcement: Ostracism in Open MultiAgent Systems. In *Computable Models of the Law*, volume 4884 of *LNCS*, pages 275–290. Springer, 2008.

[8] F. Dignum. Autonomous agents with norms. *Artificial Intelligence and Law*, 7:69–79, 1999.

[9] J. M. Epstein. Learning to be thoughtless: Social norms and individual computation. *Computational Economics*, 18(1):9–24, Aug. 2001.

[10] E. Fehr and S. Gächter. Altruistic punishment in humans. *Nature*, 415(6868):137–140, Jan. 2002.

[11] E. Fehr and S. Gächter. Cooperation and punishment in public goods experiments. *The American Economic Review*, 90(4):pp. 980–994, 2000.

[12] F. Flentge, D. Polani, and T. Uthmann. Modelling the emergence of possession norms using memes. *Journal of Artificial Societies and Social Simulation*, 4(4), 2001.

[13] 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*, 8(3), 2005.

[14] F. Giardini, G. Andrighetto, and R. Conte. A cognitive model of punishment. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, pages 1282–1288. Austin: Cognitive Science Society, 2010.

[15] J. P. Gibbs. Norms: The problem of definition and classification. *The American Journal of Sociology*, 70(5):586–594, 1965.

[16] 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*, volume 4386 of *Lecture Notes in Computer Science*, pages 101–114. Springer-Heidelberg, 2007.

[17] D. Helbing, A. Szolnoki, M. Perc, and G. SzabÃş. Punish, but not too hard: how costly punishment spreads in the spatial public goods game. *New Journal of Physics*, 12(8):083005, 2010.

[18] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 1026–1027. ACM, 2003.

[19] F. López y. López and M. Luck. Modelling norms for autonomous agents. In *Proceedings of the 4th Mexican International Conference on Computer Science*, ENC '03, pages 238–245. IEEE Computer Society, 2003.

[20] S. Mahmoud, N. Griffiths, J. Keppens, and M. Luck. An analysis of norm emergence in Axelrod's model. In *NorMAS'10: Proceedings of the Fifth International Workshop on Normative Multi-Agent Systems*. AISB, 2010.

[21] S. Mahmoud, J. Keppens, N. Griffiths, and M. Luck. Efficient norm emergence through experiential dynamic punishment. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 576–581. IOS Press, 2012.

[22] 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*, WI-IAT '11, pages 25–28. IEEE Computer Society, 2011.

[23] 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*, WI-IAT '11, pages 29–32. IEEE Computer Society, 2011.

[24] 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*, pages 136–150, 2012.

[25] S. Mahmoud, S. Miles, A. Taweel, B. Delaney, and M. Luck. Norm establishment constrained by limited resources. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pages 1819–1820, 2015.

[26] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting honest feedback in electronic markets. KSG Working Paper Series RWP02-039, 2002.

[27] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51:2005, 2005.

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

[29] N. Nikiforakis and H.-T. Normann. A comparative statics analysis of punishment in public-good experiments. *Experimental Economics*, 11(4):358–369, 2008.

[30] 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*, AAMAS '08, pages 1521–1524, 2008.

[31] D. Villatoro, G. Andrighetto, J. Sabater-Mir, and R. Conte. Dynamic sanctioning for robust and cost-efficient norm compliance. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 414–419. IJCAI/AAAI, 2011.

[32] 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*, WI-IAT '09, pages 233–240. IEEE Computer Society, 2009.

[33] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[34] 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*, volume 3415, pages 185–194. Springer-Heidelberg, 2005.