2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Overcoming Omniscience in Axelrod's Model

Samhar Mahmoud, Jeroen Keppens, Michael Luck
Department of Informatics
King's College London
London WC2R 2LS, UK.
samhar.mahmoud@kcl.ac.uk

Nathan Griffiths
Department of Computer Science
University of Warwick
Coventry CV4 7AL, UK.

***Abstract*—Norms are a valuable mechanism for establishing coherent cooperative behaviour in decentralised systems in which no central authority exists. In this context, Axelrod's seminal model of norm establishment in populations of self-interested individuals [1] is important in providing insight into the mechanisms needed to support this. However, Axelrod's model suffers from significant limitations: it adopts an evolutionary approach, and assumes that information is available to all agents in the system. In particular, the model assumes that the private strategies of individuals are available to others, and that agents are omniscient in being aware of all norm violations and punishments. Because this is an unreasonable expectation, the approach does not lend itself to modelling real-world systems such as peer-to-peer networks. In response, this paper proposes alternatives to Axelrod's model, by replacing the evolutionary approach, enabling agents to learn, and by restricting the metapunishment of agents to only those where the original defection is perceived, in order to be able to apply the model to real-world domains.**

I. 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 adhere to certain constraints, or *norms*. In peer-to-peer file sharing networks, for example, we require (at least a proportion of) peers to provide files in response to others' requests, while in wireless sensor networks nodes must share information with others for the system to determine global properties of the environment. There is typically a temptation in such settings for individuals to deviate from the desired behaviour. For example, to save bandwidth peers may not provide files and to conserve energy the nodes in a sensor network may not share information. It is therefore desirable to minimise the temptation for agents to deviate from the desired behaviour, and encourage the emergence of cooperative norms.

Axelrod's seminal investigation of norm establishment in populations of self-interested individuals [1] provides an analysis of the conditions in which norms can be established, but makes several assumptions that are unrealistic. In particular, in many domains it is not possible to remove unsuccessful agents and replicate those that are more successful, and there is no centralised control that could oversee this process. Instead, if we enable individuals to compare themselves to others, and adopt more successful strategies, then we can take a *learning interpretation* of the evolutionary mechanism [2], without needing to remove and replicate individuals. However, this learning interpretation requires that the private strategies of individuals are available for observation by other agents, which is again an unreasonable assumption. Furthermore, as has been shown elsewhere, Axelrod's model is unable to sustain cooperation over a large number of generations [3], and relies on agents being able to punish both those that defect and those who fail to punish defection, which assumes omniscience through agents being aware of all norm violations and punishments.

In this paper we investigate alternatives that allow use of the mechanisms resulting from Axelrod's investigations in more realistic settings. Specifically, we remove the assumption of omniscience and constrain the ability of agents to punish according to the defections they have observed. Finally, to obviate the need for information on others' private strategies we propose a learning algorithm through which individuals improve their strategies based on experience.

II. Axelrod's Model

In Axelrod's *norms game*, each agent in the population has four opportunities (o) in which it can choose to *defect* by violating a norm, and such behaviour has a particular known probability of being observed, or *seen* (S_o). An agent i has two decisions, or strategy dimensions, as follows. First, it must decide whether to defect, determined by its *boldness* (B_i); and second, if it sees another agent defect in a particular opportunity (with probability S_o) it must decide whether to punish this defecting agent, determined by its *vengefulness* (V_i), which is the probability of doing so. If $S_o < B_i$ then i defects, receiving a *temptation payoff*, $T = 3$, while *hurting* all other agents with payoff $H = -1$. If a defector is *punished* (P), it receives an additional punishment payoff of $P = -9$, while the punishing agent pays an *enforcement cost*, $E = -2$. The initial values of B_i

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.162

and V_i are chosen at random from a uniform distribution of a range of 8 values between $\frac{0}{7}$ and $\frac{7}{7}$.

Axelrod's simulation had 20 agents, each having four opportunities to defect, and the chance of being seen for each drawn from a uniform distribution between 0 and 1. After playing a full round (all four opportunities), scores for each agent are calculated to produce a new generation, as follows. Agents that score better or equal to the average population score plus one standard deviation are reproduced twice in the new generation. Agents that score one standard deviation or more under the average score are not reproduced, and all others are reproduced once. Finally, mutation is used to enable new strategies to arise. B_i and V_i (which determine agent behaviour) take eight possible values, so they are represented by three bits, to which mutation is applied (by flipping a bit) when an agent is reproduced, with a 1% rate.

In this model, cooperative norms are established when V_i is high and B_i is low for all members of the population, so that defection is unlikely, and observed defections are likely to be punished. Over 100 generations, Axelrod found only partial establishment of a norm against defection, so introduced an additional mechanism to support norms in his *metanorm* model, providing further encouragement for enforcing a norm. In the *metanorms* game, if an agent sees a defection but does not punish it, this is itself considered as a form of defection, and others in turn may observe this defection (with probability S_o) and apply a punishment to the non-enforcing agent. As before, the decision to punish is based on vengefulness, and punishment and enforcement costs are the same as before. The metanorm game gives runs with high vengefulness and low boldness, which is exactly the kind of behaviour needed to support the establishment of a norm against defection.

However, Axelrod's analysis of results was limited. As has been shown subsequently, allowing Axelrod's *metanorm game* to run for an extended period (1,000,000 generations) ultimately results in norm collapse [4]. As Mahmoud et al. have shown [3], this norm collapse arises as a consequence of two aspects. First, a sufficiently long run (compared to Axelrod's limited run of 100 generations) provides the opportunity for a sequence of mutations to cause norm collapse even after a norm has been established in the population. Second, such mutation is magnified by the evolutionary manner of replication generating a new population of agents.

III. Observation of Defection

In Axelrod's model, an agent *Z* is able to punish another agent *Y* that does not punish a defector *X*, even though *Z* did not see the defection of *X*. However, in reality, such metapunishment is not possible if the original defection is not observed: guaranteed observation of the original defection is an unreasonable expectation in real-world settings. In consequence, we adjust the model so that metapunishment is only permitted if an agent observes the original defection.

IV. Strategy Improvement

Reinforcement learning offers an alternative to Axelrod's evolutionary approach to improving performance of the society while keeping agent strategies and decision outcomes private. There are many reinforcement techniques in the literature, such as Q-learning [5], PHC and WOLF-PHC [6], which we use as inspiration in developing a learning algorithm for strategy improvement in the metanorms game.

A. Q-learning

Q-learning is a reinforcement learning technique that allows the learner to use the (positive or negative) reward gained from taking a certain action in a certain state in deciding which action to take in the future in the same state. Here, the learner keeps track of a table of Q-values that records an action's quality in a particular state, and updates the corresponding Q-value for that state after each action. The new value is a function of the old Q-value, the reward received, and a learning rate, δ, and the action with the highest updated Q-value for the current state is chosen. For us, Q-learning suffers from two drawbacks. First, it considers an agent's past decisions and corresponding rewards, which are not relevant here; doing so would inhibit an agent's ability to adapt to new circumstances. Second, actions are precisely determined by the Q-value; there is no probability of action, unlike Axelrod's model. To address this latter limitation, Bowling and Veloso [6] proposed policy hill climbing (PHC), an extension of Q-learning in which each action has a probability of execution in a certain state, determining whether to take the action. Here, the probability of the action with the highest Q-value is increased according to a learning rate δ, while the probabilities of all other actions are decreased to maintain the probability distribution, with each probability update occurring immediately after the action. In enhancing the algorithm, a *variable* learning rate is introduced, which changes according to whether the learner is winning or losing, inspired by the WOLF technique (win or learn fast). This suggests two possible values for δ: a low one to be used while an agent is performing well and a high one while the agent is performing poorly.

However, in one round of Axelrod's game, an agent can impose multiple punishments (potentially one per defection and non-punishment observed), while only having a small number of opportunities to defect (four in Axelrod's configuration). Therefore, punishment and metapunishment actions would be considered much more frequently than defection, leading to disproportionate update of probabilities of actions, with some converging more quickly than others. To address this imbalance we can restrict learning updates to occur only at the end of each round, rather than after each individual action, so that boldness and vengefulness are reconsidered once in each round and evolve at the same speed. The aim here is to change the probability of action significantly when

Table I
EFFECTS OF DECISIONS ON SCORES

Decision	Effects
Defect	Gain temptation payoff Hurts all other agents Potentially suffer punishment cost
Cooperate	—
Punish	Punisher pays enforcement cost Defector pays punishment cost
Not punish	Potentially suffer metapunishment (incurring punishment cost)
Metapunish	Punisher pays enforcement cost Defector pays punishment cost
Not metapunish	—

losing, while changing it much less when winning, providing more opportunities to adapt to good performance.

While basic Q-learning is not appropriate because of the lack of a probability of taking action, PHC-WOLF suffers from a disproportionate update of probabilities of action. Nevertheless, the use of the variable learning rate approach in PHC-WOLF is valuable in providing a means of updating the B and V values in determining which action to take. However, since agents that perform well need not change strategy, we can consider only one learning rate.

B. BV Learning

To address the concerns raised above, in this section, we introduce our BV learning algorithm. This requires an understanding of the relevant agent actions and their effect on boldness and vengefulness, as summarised in Table I, which outlines the different actions available to an agent and the consequences of each on the agent's score.

Now, since boldness is responsible for defecting, an agent that obtains a good score as a result of defecting should increase its boldness, and an agent that finds defection detrimental to its performance should decrease its boldness. Learning suitable values for vengefulness is more complicated, since while it is responsible for both punishment and metapunishment, these also cause enforcement costs that decrease an agent's score. Low vengefulness allows an agent to avoid paying an enforcement cost, but can result in receiving metapunishment. Vengefulness thus requires a consideration of all these aspects. This intuition is formalised within the whole simulation control loop in Algorithm 1, as follows. (Note that we use subscripts to indicate the relevant agent only when needed.)

First, and in order to determine the unique effect of each individual action on agent performance, note that we decompose the single combined total score (*TS*) of the original model into distinct components, each reflecting the effect of different classes of actions. The defection-cooperation action brings about a change only if an agent defects (Line 4): the agent's score increases by a *temptation payoff*, T (Line 5), but it *hurts* all others in the population, whose scores

Algorithm 1 The Simulation Control Loop

```
1.  for each round do
2.    for each agent i do {Decision Making}
3.      for each opportunity to defect o do
4.        if B_i > S_o then
5.          DS_i = DS_i + T;
6.          for each agent j: j ≠ i do
7.            TS_j = TS_j + H;
8.            if see(j,i,S_o) then
9.              if punish (j, i, V_j) then
10.               DS_i = DS_i + P;
11.               PS_j = PS_j + E;
12.             else
13.               for each agent k : k ≠ i ∧ k ≠ j do
14.                 if see(k,j,S_o) then
15.                   if punish (k, j, V_k) then
16.                     PS_k = PS_k + E;
17.                     NPS_j = NPS_j + P;
18.   for each agent i do {Learning}
19.     TS_i = TS_i + DS_i + PS_i + NPS_i;
20.     if TS_i < avgS then {avgS is the mean score of all agents}
21.       if explore(γ) then
22.         B_i = random(); V_i = random();
23.       else
24.         if DS_i < 0 then
25.           B_i = B_i − δ;
26.         else
27.           B_i = B_i + δ;
28.         if PS_i < NPS_i then
29.           V_i = V_i − δ;
30.         else
31.           V_i = V_i + δ;
```

decrease by H (line 7). If an agent cooperates, no scores change. We can therefore use just one distinct value to keep track of this score, referred to as the *defection score* (*DS*), which determines whether to increase or decrease boldness.

Conversely, punishment and metapunishment both have two-sided consequences: if an agent j sees agent i defect in one of its opportunities (o), with probability S_o (Line 8), and decides to punish it (which it does with probability V_j; Line 9), i incurs a punishment cost, P, to its *DS* (Line 10), while j incurs an enforcement cost, E, to a different score, its *punishment score*, *PS* (Line 11). As the name suggests, *PS* captures the total score obtained by an agent as a result of punishing another, and applies to both punishment and metapunishment (enforcement costs). There is also a different change (resulting from potential subsequent received metapunishment) if it decides not to punish (Line 12). If j does not punish i, and another agent k, which has already observed i's defection, sees this (Line 14) and decides to metapunish (Line 15), then k incurs an enforcement cost, E, to its *PS*, and j incurs a punishment cost P to its *no punishment score*, *NPS* (obtained from not punishing, and comprising the metapunishment cost alone).

In Axelrod's original model, agents that are one standard deviation or more below the mean are eliminated and replaced in the subsequent population generation with new agents following the strategy captured by the B and V values of those agents that are one standard deviation or more above

the mean. Thus, poorly performing agents are replaced by high-performing ones. In contrast, in our model, we distinguish more simply between good and poor performance, with only agents that score below the mean reconsidering their strategy. Thus, for each agent, we combine the various component scores into a total, *TS* and, if the agent is performing poorly (in relation to the average score, $avgS$ in Line 20), it reconsiders its boldness and vengefulness.

Now, in order to ensure a degree of exploration (similar to mutation in the original model's evolutionary approach) and to enable an agent to step out of the learning trend, we adopt an *exploration rate*, γ, which regulates adoption of random strategies from the available strategies universe (Line 21). If the agent does not *explore*, then, if defection is the cause of a low score (Line 24), an agent decreases its boldness, and increases it otherwise. Similarly, agents increase their vengefulness if they find that the effect of not punishing is worse than the effect of punishing (Line 28), and decrease vengefulness if the situation is reversed. As both *PS* and *NPS* represent the result of two mutually exclusive actions, their difference for a particular agent determines the change to be applied to vengefulness. For example, if $PS > NPS$, then punishment has some value, and vengefulness should be increased. Finally, given a decision on whether to modify an agent's strategy, the degree of the change, or *learning rate* (δ), must also be considered. Since vengefulness and boldness have eight possible values from $\frac{0}{7}$ to $\frac{7}{7}$, we adopt the conservative approach of increasing or decreasing by one level at each point, corresponding to a learning rate of $\delta = \frac{1}{7}$. Thus, an agent with boldness of $\frac{5}{7}$ and vengefulness of $\frac{3}{7}$ that decides to defect less and punish more will decrease its boldness to $\frac{4}{7}$ and increase its vengefulness to $\frac{4}{7}$.

C. Evaluation

The algorithm is designed to mimic the behaviour of Axelrod's evolutionary approach as best possible, while relaxing its unrealistic assumptions. This allows us to replicate Axelrod's results and investigate his approach in more realistic problem domains. The analysis of a sample run reveals that agents with low vengefulness and agents with high boldness start changing their strategies. Here, agents with high boldness defect frequently, and are punished as a result, leading to a very low *DS*, in turn causing these agents to decrease their boldness. Agents with low vengefulness do not punish and are consequently frequently metapunished; as a result, their *PS* is much better than their *NPS*, causing them to increase their vengefulness. The population eventually converges to comprise only agents with high vengefulness and low boldness. While noise is still introduced via the exploration rate causing random strategy adoption, the learning capability enables agents with such random strategies to adapt quickly to the trend of the population. Requiring the original defection to be observed in order to apply a metapunishment, we ran experiments over different periods,

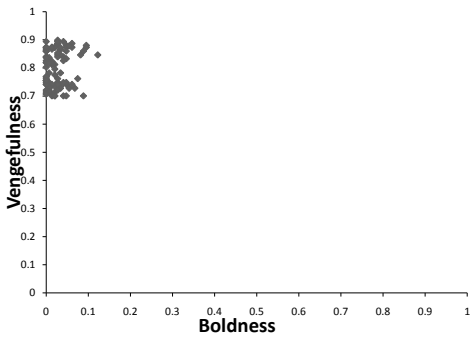

Figure 1. Strategy improvement with defection observation ($\gamma = 1\%$)

with results indicating that norm establishment is robust in all runs. The result of 1000 runs for 1,000,000 timesteps is shown in Figure 1, with each point representing the average boldness and vengefulness of the population after a single run. This shows that the learning algorithm is able to avoid the weakness of the original model in the long term [3].

V. Conclusion

In systems of self-interested autonomous agents, we often need to establish cooperative norms to ensure the desired functionality. Axelrod's work on norm emergence [1] gives valuable insight into the mechanisms and conditions in which such norms may be established, but suffers from limitations relating to assumptions of omniscience. In response, this paper has (i) investigated those aspects of Axelrod's investigation that are unreasonable in real-world domains, and (ii) proposed *BV learning* as an alternative mechanism for norm establishment that avoids these limitations.

Through this approach we have shown that not only it is possible to avoid the unrealistic assumption of knowledge of others' strategies, but also that we can avoid norm collapse, even with observation constraints on metapunishment, by enabling individuals to incrementally change their strategies.

References

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

[2] R. Riolo, M. Cohen, and R. Axelrod, "Evolution of cooperation without reciprocity," *Nature*, vol. 414, pp. 441–443, 2001.

[3] S. Mahmoud, N. Griffiths, J. Keppens, and M. Luck, "An analysis of norm emergence in axelrods model," in *Proc of NorMAS '10*. AISB, 2010.

[4] 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.

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

[6] M. Bowling and M. Veloso, "Rational and convergent learning in stochastic games," in *Proc of IJCAI'01*, 2001, pp. 1021–1026.