Beating Cheating: Dealing with Collusion in the Non-Iterated Prisoner's Dilemma

N. Höning [a] T. Kozelek [b] M.C. Schut [a]

[a] *VU University, Amsterdam,* {nhg400,mc.schut}@few.vu.nl
[b] *Charles University, Prague*, tomas.kozelek@gmail.com

Abstract

The Iterated Prisoner's Dilemma (IPD) is a well-known challenging problem for researching multi-agent interactions in competitive and cooperative situations. In this paper, we present the Ask-First (*AF*) strategy for playing multi-agent non-Iterated PD (nIPD) that is based on evolving trust chains between agents. Each agent maintains a (relatively small) table containing trust values of other agents. When agents are to play each other, they ask their neighbours what trust they put in the opponent. Chains are then followed until an agent is found that knows the opponent and the trust value is propagated back through the chain. The played move is then decided based upon this trust value. When two agents have played each other, they update their trust tables on the basis of the outcome of the game. The strategy is first evaluated in a benchmark scenario where it is shown that it outperforms a number of benchmark strategies. Secondly, we evaluate the strategy in a scenario with a group of colluding agents. The experiments show that the *AF* strategy is successful here as well. We conclude that the *AF* strategy is a highly flexible, scalable and distributed way (the chain topology adapts to the way that agents are picked to play each other) to deal with a difficult multi-agent nIPD problem (i.e., robust against collusions).

1 Introduction

In open e-commerce systems (e.g., an electronic market), *collusion* is a serious threat to the honest operation of the system. Collusion is the agreement between two or more persons to deceive others to obtain an secretive (or: illegal) objective. On eBay, this happens by so-called *shil bidding*, where friends of the seller bid on an item solely with the intention to raise the price. These friends can even be other eBay accounts of the seller himself and not necessarily real persons. Another example of collusion happens when in recommender systems, items are evaluated positively by the seller's friends (which may, again, be fake accounts of the seller). In such situations, it is very important for a potential buyer (who is truthful) to judge the reliability of the bids or the recommendations. Ideally, one wants to achieve this without having to rely on a trusted third-party or authority.

In this paper, we suggest a way to do exactly this based on the idea of *trust networks*. In such networks, agents maintain the trust that they have in each other and communicate this with each other when necessary. The intuition behind this method is that in real life, we ask our friends how they feel about a certain item or some particular shop. This intuition has lead us to develop the very simple Ask-First (*AF*) strategy that faithful agents can employ to protect them from malicious or unreliable agents.

Within the context of Iterated Prisoner's Dilemma (IPD), the strategy works as follow (we provide more detailed information later in the paper). Firstly, an *AF* strategy agent maintains a (relatively small) table containing trust values of other agents. Then, when agents are to play each other, they ask their neighbours what trust they put in the opponent. Next, chains are then followed until an agent is found that knows the opponent and the trust value is propagated back through the chain. Then, the played move is then decided based upon this trust value. Finally, when two agents have played each other, they update their trust tables on the basis of the outcome of the game.

The research objective of this paper is two-fold: 1) to show that the *AF* agents outperform the benchmark agents (all-cooperators and all-defectors); and 2) to show that the *AF* agents outperform collusion agents. For objective 2), we also want to show that the *AF* agents use the created network chains effectively and that colluding agents cannot invade these chains.

This paper has the following structure. In Section 2, we present the background for our work: computational trust and reputation, referral networks and social networks. We also explain the non-Iterated Prisoner's Dilemma. Section 3 lays out the details of our framework and the Ask-First (*AF*) strategy. In Section 4 we show by a number of experiments the effectiveness of the *AF* strategy. Finally, in Section 5, we draw conclusions about the performed study and provide some pointers for future work.

2 Background

The work presented in this paper, builds further on research on computational trust and reputation, referral networks, and social networks. In this paper, we place all within the context of the Iterated Prisoner's Dilemma (IPD). In this Section, we briefly touch upon each of the topics.

2.1 Trust, Reputation and Referral

The concept of *trust* is essential in societies and open systems in order to maintain 'good' social interactions and commitments between individuals. As mentioned above, on eBay, you want to trust the person from whom you are buying your items. In real life, we have many (albeit implicit, unconscious) mechanisms operating for managing trust; in virtual life (internet), we have not yet established such mechanisms [1]. Much research in the last decade has been dedicated to the question how to use trust as a mechanism for regulating social interaction. In the early 1990s, Marsh [7] presented a formalisation of trust that pins down a number of defining properties of trust in order to facilitate a precise discussion about trust. The formalism was implemented for a multi-agent system in a PD scenario, demonstrating a recognisable behaviour of trust among the agents.

An often-used mechanism for managing trust is by means of *reputation*: a societal indication of how much you can trust someone, reflecting its past actions. Like the concept of 'trust', 'reputation' is a convoluted term and needs to be clarified unambiguously for precise discussion and application; Mui *et. al* [9] do this by giving a concise overview of the notion within the context of multi-agent systems. In terms of applications, current major websites (eBay, Amazon) have *centralised* reputation mechanisms in place, where people's ratings about each other are collected, processed and communicated back. However, such centralisation of reputation is not always possible, e.g., for peer-to-peer service provision [14]. It can be expected that future systems will become increasingly more open and require such decentralised mechanisms for maintaining trust and reputation.

A recent development in the search for "decentralised reputation management" concerns so-called *referral networks*. In such networks, agents communicate information about trust and reputations are built based on this information. The networks are used for query-based searching for information and expertise in a person's social network [5]. In a referral network, nodes have neighbours (whom they query directly) and acquaintances (whom they query only when referred to); both sets are dynamic (neighbours can become acquaintences and vice versa) and usually limited. Studies of referral networks by Singh *et. al* [14] have looked at how network structure evolved under various circumstances; how agent learning models (enabling agents to learn about each other in terms of expertise – producing correct answers, and sociability – providing good referrals) affects the quality of the work; and how to design self-organising referral networks.

2.2 Social Networks

The basis of the referral networks described above is the *social network* that connects individuals within a collective system. In such a network of individuals, there are links between these individuals that represent, for example, friendship, kinship or values. Social network analysis views these networks as graphs, in which the nodes are individuals and the edges are the links. Within the context of this paper, research on social networks includes, for example, the study of decentralised search algorithms [6] and investigation of the relationship between social network topologies and emergent behaviour [2].

A particular stream of research worth mentioning is the investigation of so-called *small-world networks*. Since the formalisation of the small world problem (originally coined by Milgram [8]) by Watts and Strogatz [13] in the nineties, much research on social networks looks at such networks being small-world networks: it takes relatively few steps to go from one random node to another. This type of networks has two important properties: a short average path length (hence, the relatively few steps to reach other nodes), and a high clustering coefficient (number of a node's neighbours that also know each other). The experiments presented later in this paper also consider small-world networks.

2.3 Non-Iterated Prisoner's Dilemma

The well-known Iterated Prisoner's Dilemma (IPD) is a generic abstract representation of complex social dilemmas [3]. It is a (non-zero-sum) game where two players make decisions simultaneously whether to defect or cooperate and receive rewards based on the combination of their two actions. The ordinance of the payoffs makes this game a dilemma: [dc] > [cc] > [dd] > [cd], where $[xy]$ denotes the reward for player 1 where player 1 plays x, player 2 plays y and $x, y \in \{$c(ooperate), d(efect)$\}$. The dominant strategy for the one-shot PD (where the game is played once) is for both players to defect. The iterated version of the game (where it is played for a number of times) does not have a single dominant strategy. In a competition-based evaluation of the IPD, it was found that an extremely simple strategy beats the other strategy: *tit-for-tat*, where the opponent's last action was simply copied (with default action 'cooperate').

In this paper, we have a population of agents that repeatedly plays pairwise PD games. Note, however, that it is not an implementation of IPD, because the agents change their interaction partner in every round and have a limited memory of their previous opponents. Like in [4, 12], we call this version the *non-Iterated PD* (nIPD).

While in the one-shot and iterated PD it is almost certain that the players will ultimately defect each other, in the nIPD this is not the case at all. Much research has been done on the *evolution of cooperation* in nIPD, addressing the important question how cooperation can arise between selfish agents. Nowak and May [11] showed that in a two-dimensional cellular automata (where a cell could be in a 'cooperate' or 'defect' state) if the PD was used as the update rule, patterns of cooperation and defection emerged. In follow-up work by Olifant [12], it is shown that in a society of agents, *spatiality* plays an important role: agents that are close by each other are more likely to play than those that are further away. This work gives rise to the issue of how agents are connected to each other: can the structure of the connection network affect the evolution of cooperation?

In more recent work, Ellis and Yao [4] address this issue by looking at a *social network* inspired approach for the evolution of cooperation. In their (non-spatial) nIPD, links are formed between cooperating agents. These links are reinforced by repeated cooperation, while defection breaks a link. All links taken together represent the social network of the agents. Ellis and Yao present a strategy that can exploit this network: the *discriminator* agent bases its move (cooperate or defect) on the *centrality* of its opponent. (This centrality represents the agent's reputation among its neighbours.) The experiments for this strategy showed that the discriminator strategy is more successful than all-cooperators or all-defectors. In another series of experiments, agents are able to evolve strategies while playing – ranging from all-cooperate to all-defect. The evolving parameters included 1) the probability that an agent interacts with another agent that has a lower reputation than some threshold value, and 2) this threshold. The experiments showed that when agents are able to observe the centrality, then the population evolves to all-cooperators.

3 The *Ask-First* Strategy

The main idea of the *Ask-First* strategy is to *ask* around before you *act*. For the nIPD, this means asking a neighbour for advise about whether to (c)ooperate with or (d)efect an unknown opponent. The asking process is recursive: when you are asked for advise about someone you do not know either, you forward the question to a neighbour of your own, and so forth. This asking around leads to so-called *information chains* between agents over which trust information is communicated. Note that the recursive nature of these information chains in our approach sets it apart from referral networks (which were explained above), because these work *iteratively*.

The protocol for each agent employing the *AF* strategy[1] involves the following steps: select best neighbour, ask this neighbour for advise, process the advise, play the PD game and do some aftermath (updating). We describe each of these steps in the following Sections. Before this, we first explain the dynamics of the social network of the agents.

[1] In terms of reciprocity, the reputation-based *AF* strategy is a *downward* kind of indirect reciprocity [10, p.1292]: individual a has helped b and *therefore* receives help from c.

Network Dynamics We are interested in testing the *AF* strategy in a set of agents that are connected with each other through a small-world network. In other words, this paper is not about showing that the evolved network is small-world; that is our starting point and we show that the *AF* agents can successfully exploit such networks. However, we do not want to simply impose the small-world property on the network; instead, we select PD-playing couples in such a way that a small-world social network evolves. As mentioned above, small-world networks have low average path lengths L (comparable to L of a random graph of the same size) and a high clustering coefficient C (much higher than C of a random graph). For building small-world social networks, one could follow Watts and Strogatz [13]: often connect neighbouring agents and occasionally remote agents. Here, we use a more refined method that employs a concept of *distance*, based on an approach suggested by Kleinberg [6].

Let us define the concept of distance as follows. Firstly, let each agent x be uniquely identified by an integer $I_x \in [1, N]$, where N is the total number of agents. We assume that agents whose identifiers are close to each other are more likely to interact with each other. For example, in the context of eBay agents, this could express some common interests in goods. In the context of our nIPD agents, it simply means that close agents are more likely to play each other. Let a *distance* measure $dist(a, b)$ represent how close agent a is to agent b; in general: $dist(x, y) = min(abs(I_x - I_y), N - abs(I_x - I_y))$, where x, y are agents and I_x, I_y their identifiers, respectively. (Note that this function makes agents I_1 and I_N close to each other.)

For building small-world networks, agents are connected with each other if there is little distance between them. In other words. If a, b and c are agents, then a should have a higher probability to be connected with b then with c if $dist(a, b) < dist(a, c)$. Formally, let x, y be agents and the probability that they are connected be $\frac{1}{dist(x,y)^\alpha}$ (according to Kleinberg), where α is some given constant; then small-world networks emerge when α is around 2. With these notions, we use Kleinberg's probability-vector approach [6] to evolve the desired small-world social networks. For the experiments presented below, we have obtained empirical evidence supporting this. We have not included these measurement for reasons of space.

3.1 Select Best Neighbour

Let an agent a have a *neighbourhood* Ω, containing the names of at most K other agents. For each of the agents in the neighbourhood, agent a has assigned a *trust value* representing how much a trusts this agent: let this trust value be $\tau \in \mathbb{R}$, where $\tau \in [0, 1]$ (representing minimum to maximum trust, respectively). If agent a plays a neighbour and this neighbour defects, then its trust decreases and vice versa (later we explain the calculation of this value in detail).

When faced with an opponent o that is not in the agent's neighbourhood, the agent chooses one of its neighbours, let this be agent b, to ask for advise about playing o. The choice for a specific neighbour is based on 1) how much agent a trusts agent b, and 2) what the distance between agents b and z is. We represent the trust of agent a in agent b by $trust(a, b)$; the distance between them by $dist(a, b)$. The two measurements are integrated into the function $eval(a, b, o)$ that expresses the evaluation of neighbour b by agent a about opponent o. In general, $eval(a, b, o) = \omega_{trust} * trust(a, b) + \omega_{distance} * (1/dist(b, o))$, where $a, b \in \Omega$, $o \notin \Omega$, $\omega \in [0, 1]$ and $\sum_\omega = 1$. This function thus *weighs* trust and distance: the higher the weight of distance, the more probable it is that the opponent is reached soon; a high weight of trust may produce more reliable chains but it is also more likely to fail in finding the opponent at all. Finally, an agent decides to ask the neighbour whom it evaluates highest: $max_{b \in \Omega}\, eval(a, b, o)$.

3.2 Ask for Advise

After agent a has decided to ask neighbour b about opponent o, he thus asks agent b for advise. Asking is recursive, thus when agent b does not have o in his Ω, then he selects his best neighbour to ask for advise about o. If it is not possible to find an agent that knows the opponent within 6 steps[2], then the chain is not built and agent a uses his default behaviour (here: cooperate).

If a successful chain was built (i.e., o was found within 6 steps), then the trust value is propagated back to agent a, multiplied by the trust values in the information chain I: $\Pi_{a \in I} trust(a, bn(a))$, where $bn(a)$ is the best neighbour of a. Consider, for example, that the built chain goes from a via b to agent c who knows o. Then the propagated trust value received by a is $trust(c, o) \times trust(b, c) \times trust(a, b)$.

[2]The number '6' was chosen because of the infamous "six degrees of separation" [8].

3.3 Process Advise

If agent a has the propagated trust value t, then it can still decide whether to use the received advise or not. This decision is based on a system-wide *chain trust* threshold θ_{chain} (here: 0.3). If $t > \theta_{chain}$, then a uses the advise; otherwise not (and a uses his default behaviour – cooperate). After that, a's decision whether to cooperate or not[3] is based on the (also system-wide) threshold θ_{trust} (here: 0.5). If $t > \theta_{trust}$, then a cooperates; otherwise he defects.

3.4 Play and Aftermath

Playing the game is straightforward: each player makes his move and receives a payoff based according to a given nIPD payoff table. If an agent has finished a game, three things happen. Firstly, the opponent is added to the neighbourhood (if he was not in there yet). Secondly, the trust value of the opponent is updated according to the following rule[4]:

$$\textbf{if } (cooperation) \textbf{ then } \tau_{new} = \tau_{old} + (1 - \tau_{old})/2$$

$$\textbf{if } (defection) \textbf{ then } \tau_{new} = \tau_{old} - (\tau_{old})/2,$$

where $\tau_{old} = 0.5$ iff the opponent was not yet a neighbour. Note that these function are both sigmoid curves. This means that if an opponent changes his behaviour, it will have a quick effect on the trust in him. This follows recommendations by Axelrod [3] in that successful strategies should quickly retaliate and forgive. The opponent is then added to Ω with the Kleinberg probability[5] $1/dist(a, o)^{\alpha}$. If the opponent is added and the neighbourhood size exceeds the allowed K entries, then a random agent is removed from Ω (in order to keep the size within K).

4 Experimental Evaluation

We present an experimental evaluation of the effectiveness of the *AF* strategy, in which we compare the performance of agents employing the *AF* strategy with *colluding* agents (who do not reveal true trust values). As mentioned, all the experiments involve a nIPD scenario, in which agents repeatedly play PD games with other agents. Each experiment consists of a fixed number of iterations; in each iteration, an opponent is chosen once for each agent, after which both players remain in the pool. This means that each agent plays at least one game in each iteration, and probably more (in which case the payoffs within one iteration are accumulated). Every third iteration, an *evolutionary update* happens (like in [4]): two random agents are selected, of which the one with lowest average payoff[6] is replaced by a new agent that follows the strategy of the other agent (removed agents are also removed from all neighbourhoods). In this Section, we subsequently present 1) the research hypotheses, 2) the strategies of the agents, 3) the experimental design and setup, and 4) the results and analysis.

4.1 Research Hypotheses

The objectives stated in the introduction translate into the following 3 hypotheses:

- **Hypothesis 1:** The AF strategy is successful against defection.
- **Hypothesis 2:** The AF strategy is successful against collusion.
- **Hypothesis 3:** Colluding agents cannot spread false information over the information chains.

We measure the successfulness of a strategy by means of the proportion of agents that employ that strategy. Additionally, we measure *Utilitarian Social Welfare*, which is the sum of payoffs all agents achieved in the current iteration. For hypothesis 3, we look at the information chains and compare the number of *built* chains with the number of *used* chains and the number of *only-AF* chains (those chains that only contain *AF* agents).

[3] If agent a has opponent o in his own neighbourhood, then he uses this exact same decision procedure (where he knows t himself).
[4] Note that the only trust value to be updated is thus the opponent's one, not the trust value of the best neighbour.
[5] Earlier we explained that based on this probability, it is decided which agents are playing each other. Here, the constant α is relatively high to support locality.
[6] This is the total payoff accumulated over its lifetime.

4.2 Strategies

We employ some simple strategies to evaluate their interplay. If not mentioned otherwise, all agents can be asked to build up information chains and report their true trust evaluations about agents they know.

Always Cooperate (AC) – This strategy cooperates in every interaction.

Always Defect (AD) – This strategy always defects the opponent.

Ask First (AF) – This is the strategy that was explained in detail in Section 3.

Simple Collusion (SC) – This strategy behaves like AD, i.e., it always defects. However, when this agent, say m, is asked for the trust value of a neighbour n, it returns $1 - trust(m, n)$. The consequence of this is that SC agents give each other good ratings and cooperating agents bad ratings.

4.3 Design and Setup

We considered three different experiments, between which we varied the initial populations: in experiment 1, the initial population consists $\frac{1}{2}$ AD agents and $\frac{1}{2}$ AF agents; in experiment 2, it consists of $\frac{1}{3}$ AD agents, $\frac{1}{3}$ AC agents and $\frac{1}{3}$ AF agents; in experiment 3, it consists of $\frac{1}{2}$ AF agents and $\frac{1}{2}$ SC agents. In all experiments, there were 20 runs for each experiment; the population size (N) was 150; the number of iterations was 600 (for exp.2, we also report on an experiment with 3,000 iterations). Regarding the neighbourhoodsize and the weights, we tested each with two values and use the average of the obtained results for the analysis: we let $K \in [0.7 * logN * \sqrt{logN}, 1.1 * logN * \sqrt{logN}]$, and $\omega_{distance} \in [0.3, 0.7]$.

4.4 Results and Analysis

Figures 1–3 show the results of experiments 1, 2 and 3, respectively. This section reviews our research hypotheses.

Hypothesis 1 – For experiment 1, this hypothesis holds. We conducted a Welsh Two-Sample T-Test ($t = 51.1728, df = 712.562, p\text{-}value < 2.2e - 16$) to test the difference between the population proportions of both strategies at iteration 600 (the mean population size of AF was 183.99 and the mean population size of AD was 28.41). Figure 1(a) shows the strategy proportions over all iterations. It is noteworthy to say that if a cooperating strategy like AF takes over the population, this is not only self-beneficiary. Figure 1(b) shows the Utilitarian Social Welfare that is maximised as AF agents take over the population. The high spike in the beginning results from the initially trusting behaviour of AF agents facing unknown AD agents. Quickly, AF agents learn not to trust AD agents and spread that information via information chains to other AF agents.

For experiment 2, this hypothesis does not hold: the AF strategy loses from the AD strategy (see Figure 2(a)). We conducted another Welsh Two-Sample T-Test ($t = 28.7613, df = 1024.961, p\text{-}value < 2.2e - 16$), showing that the AD agents significantly outnumber the AF agents (the mean population sizes were 63.47 and 142.78 for AF and AD, respectively). However, Figure 2(b) shows that the AF strategy recovers once there are no pure cooperators left.

Hypothesis 2 – This hypothesis holds. We conducted a Welsh Two-Sample T-Test ($t = 67.04, df = 669, p\text{-}value < 2.2e - 16$) to test on the difference of the population representation of both strategies in the end of runs in experiment 3 (AF, SC). The results show with strong significance that the AF strategy outnumber the SC strategy after $N * 4$ iterations (see Figure 3(a)). The mean of AF was 205.26 and the mean of SC was 19.74.

Hypothesis 3 – To understand the setting of AF agents against SC agents better, we monitored the use of information chains. In particular, we protocolled the number of chains that were successfully built (i.e. the original asker, an AF agent, got an answer of someone who had his opponent in his trust table), the number of information chains that were actually used (i.e. the returning trust information exceeded the chain trust threshold) and the number of the used chains which only contained AF agents. Figure 3(b) shows that AF agents will use exclusively use chains that only consists of other AF agents. This may seem strange at first, but it is a consequence of the design. Defecting agents will seldom be chosen for building the next step in a chain as they quickly rank low in trust at all AF agents, due to the retaliating nature of the AF strategy. If they are chosen, they lower the overall returned trust, making it highly unlikely that the asking AF agent will use the chain for his decision. In this sense, the AF strategy could be called unforgiving: defecting agents will not be asked for information[7].

5 Conclusions

We proposed and implemented a distributed reputation system, that effectively exploits a small-world social network. The introduced *Ask-First* strategy employed by the agents is based on ask-first-act-later: if faced with an unknown opponent, these agents ask around their social network, and decide on dealing with the opponent based on the advise received from this network. By means of a series of experiments, we showed that our strategy can handle malicious agents that do not reveal honest advise when they are asked for. We tested the strategies in a multi-agent non-Iterated Prisoner Dilemma scenario, but expect that it can be useful for a number of practical applications, e.g., distributed web service provision or for distributed trust management in open recommender systems (eBay or Amazon). In future work, we will go more into putting the strategy into practice.

References

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the Hawaii International Conference on System Sciences*, 2000.

[2] G. Abramson and M. Kuperman. Social games in a social network. *Phys. Rev. E*, 63(3), 2001.

[3] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[4] T.S. Ellis and X. Yao. Evolving cooperation in the non-iterated prisoner's dilemma: a social network inspired approach. In *IEEE Congress on Evolutionary Computation*, 2007.

[5] Henry A. Kautz, Bart Selman, and Mehul A. Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997.

[6] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians*, 2006.

[7] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computing Science, University of Stirling, 1994.

[8] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[9] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 280–287. ACM Press, 2002.

[10] A. Nowak and K. Sigmund. Evolution of indirect reciprocity. *Nature*, 437:1291–1298, 2005.

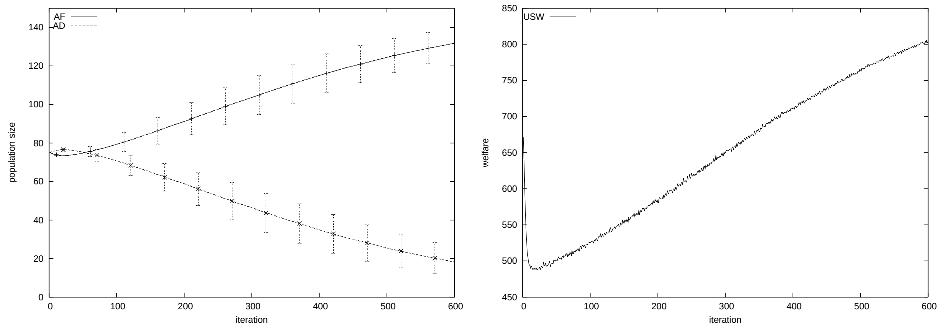
[11] M.A. Nowak and R.M. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, 1992.

[12] M. Oliphant. Evolving cooperation in the non-iterated prisoner's dilemma: The importance of spatial organization. In R. Brooks and P. Maes, editors, *Proceedings of Artificial Life IV*, pages 349–352. MIT Press, 1998.

[13] J. Duncan Watts and H. Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393, June 1998.

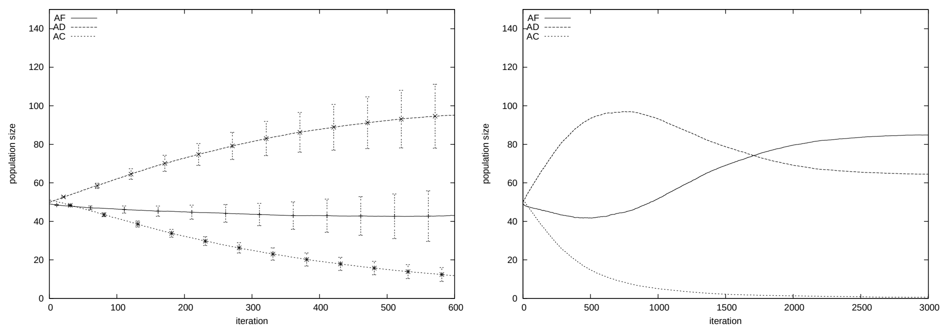
[14] B. Yu, M.P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Proceedings of the First IEEE Symposium on Multi-Agent Security and Survivability*, pages 1–10, 2004.

[7] In some follow-up experiments (not reported here), we tested the forgiveness of the *AF*-strategy by putting them in a world with *alternating* agents (who change from always-cooperate to always-defect agents every x iterations. The results of these experiments showed that *AF* agents respond quite rapidly to such changes.



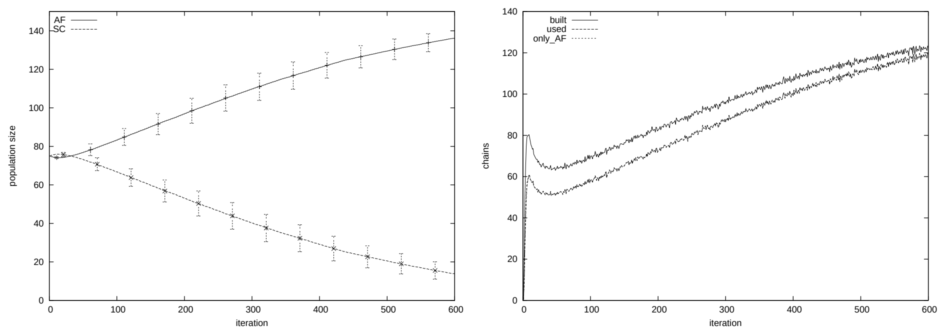
(a) Strategy populations (b) Utilitarian Social Welfare

Figure 1: AD vs AF in experiment 1, N=150



(a) Strategy populations (b) Strategy populations (3,000 iterations)

Figure 2: AD vs AC vs AF in experiment 2, N=150



(a) Strategy populations (b) Information chain usage

Figure 3: AF vs SC in experiment 3, N=150