The Impact of Naive Agents in Heterogeneous Trust-aware Societies

Amirali Salehi-Abari and Tony White

School of Computer Science, Carleton University,
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada
{asabari, arpwhite}@scs.carleton.ca

Abstract. Autonomous agents require trust and reputation concepts in order to identify communities of agents with which to interact reliably in ways analogous to humans. Agent societies are invariably heterogeneous, with multiple decision making policies and actions governing their behaviour. Through the introduction of naive agents, this paper shows empirically that while learning agents can identify malicious agents through direct interaction, naive agents compromise utility through their inability to discern malicious agents. Moreover, the impact of the proportion of naive agents on the society is analyzed. The paper demonstrates that there is a need for witness interaction trust to detect naive agents in addition to the need for direct interaction trust to detect malicious agents. By proposing a set of policies, the paper demonstrates how learning agents can isolate themselves from naive and malicious agents.

1 Introduction

The concept of trust is crucial in driving decision making and relationships in human and artificial societies. According to Jarvenpaa et al.[5], trust is an essential aspect of any relationship in which the trustor has no control over the actions of a trustee, the decision is important, and the environment is uncertain.

Agents make use of trust and reputation models in deciding how, when and who to interact with in a specific context [7]. Stated another way, an agent must be able to model trustworthiness of potential interaction partners and make decisions based on that model. It is a commonly held position that the main utility of trust and reputation models is minimizing the risk of interacting with others by avoiding interacting with malicious agents. With this view in mind, the principal objective of such models is the detection of untrustworthy agents.

The majority of computational trust and reputation models are designed and evaluated based on the assumption that the agent society only comprises two types of agents: trust-aware and malicious. It is our view that an agent society should include another type of agent called *a naive agent*. Naive agents are naive in terms of deciding how, when and who to interact with while always cooperating with other agents. The effects of naive agents on trust-aware individuals and the whole of society have not been analyzed to date. This observation motivates the work reported in this paper. Agents types represented in this paper have extremely limited cognitive properties modeled with simple policies. This is deliberate as we wish to understand the importance of agent heterogeneity in societal dynamics. Our research goal is the understanding of self-organization of agents into trusted communities.

This paper makes the following contributions: the introduction of the concept of a naive agent; analyzing the impact of this agent class on agent societies using a game-theoretic model on a simulation platform, and a strategy proposal for trust-aware agents to deal with them. While ART [3] aims to provide a unified platform for trust model evaluation it does not consider variables that are central to the evaluation proposed in this paper. Therefore, in order to evaluate our model, we design our own testbed which is described in Section 4.

The remainder of this paper is organized as follows. After describing related work in Section 2, we discuss naive agents and the environment model of agents in Sections 3 and 4 respectively. We describe the proposed agent model in Section 5, and experiments in Section 6. Finally, conclusions and future work are explained in Section 7.

2 Related Work

The body of research on trust and reputation models is substantial; a review of which can be found in [7] and [9]. In this paper our discussion is limited to models that incorporate, or discuss, multiple information sources.

Regret [8] is a decentralized trust and reputation system which takes into account three different sources of information. The direct trust, witness reputation, neighborhood reputation and, system reputation are introduced in Regret. Except for the direct trust module, the rest of the model is not readily applicable here because it is unclear how each agent can build the social network.

Yu and Singh proposed a social reputation management system in which they represented an agent's trust ratings regarding another agent as a scalar and combined them with testimonies [14]. Yu et al. have proposed the trust model in large-scale peer-to-peer systems in which each peer has its own a set of acquaintances [13]. The acquaintance's reliability and credibility are included in this model but are not used to drive the selection of new acquaintances as proposed here. However, the trust model strongly influenced the model described in Section 5.

Huynh et al. introduce a trust and reputation model called FIRE that integrates a number of information sources to estimate the trustworthiness of an agent [4]. Specifically, FIRE incorporates interaction trust, role-based trust, witness reputation, and certified reputation to provide a trust metric. FIRE does not consider malicious witness providers because it assumes honest agent information exchange. The research reported here explicitly deals with inaccurate witness providers.

In the Social Interaction Framework (SIF) [11], agents are playing a Prisoner's dilemma set of games with a partner selection phase. Each agent receives the results of the game it has played plus the information about the games played by a subset of all players. An agent evaluates the reputation of another agent based on observations as well through other witnesses. However, SIF does not describe how to find witnesses, which the model reported here does.

There are few trust models which consider the existence of an adversary in providing witness information and present solutions for dealing with inaccurate reputation, essentially the problem of naive agents of interest here. TRAVOS [12] models an agent's trust in an interaction partner. Trust is calculated using probability theory that takes account of past interactions and reputation information gathered from third parties while coping with inaccurate reputations. Yu and Singh [15] is similar to TRAVOS, in that it rates opinion source accuracy based on a subset of observations of trustee behavior.

Most recently, Salehi-Abari and White [10] have empirically shown that exploitation resistance is important for trust and reputation models. They declared that exploitation resistance "implies that adversaries cannot take advantage of the trust model and its associated systems parameters even when they are known or partially known to adversaries."

3 Naive Agent

We define a naive agent in the following way: a naive agent is incapable of properly deciding how, when and with whom to interact. As such, it fails to detect and stop interacting with untrustworthy agents due to its inability to properly assess other agents. Such agents are optimistic; they consider all other agents to be completely trustworthy and always cooperate with every member of the society. Naive agents usually do not have any malicious intention.

Examples of naive agent can be seen in many places. On eBay, sellers receive feedback (+1, 0, -1) in each auction and their reputation is calculated as the sum of those ratings over the last six months. It has been observed that there are many users (buyers) who do not receive satisfactory goods or services but they rate the sellers highly and even continue interacting with them. In other words they fail to "complain". We see these users as naive users.

4 Environment Model

The majority of open distributed computer systems can be modeled as multi-agent systems (MAS) in which each agent acts autonomously to achieve its objectives. Autonomy is represented here by the evaluation of pre-determined policies that cause changes in agent trust and reputation models and subsequent changes in societal structure. Our model incorporates heterogeneous agents interacting in a game theoretic manner. The model is described in the following 3 subsections.

4.1 Interactions

An agent interacts with a subset of all agents. Two agents are *neighbors* if both accept each other as a neighbor and interact with one another continuously. An agent maintains the *neighborhood* set which is dynamic, changing when an agent is determined to be untrustworthy or new agent interactions are required. Agents have bounded sociability as determined by the maximal cardinality of the neighborhood set. Agents can have two types of interactions with their neighbors: *Direct Interaction* and *Witness Interaction.*

Direct Interaction. Direct interaction is the most frequently used source of information for trust and reputation models [9, 7]. Different fields have their own interpretation of direct interaction. For example, in e-commerce, direct interaction might be considered to be buying or selling a product.
Witness Interaction. An agent can ask for an assessment of the trustworthiness of a specific agent from its neighbors and then the neighbors send their ratings of that agent to the asking agent. We call this asking for an opinion and receiving a rating, a ***Witness Interaction***.

4.2 Games: IPD and GPD

Direct and witness interactions are modeled using two extensions of the Prisoner's Dilemma. The Prisoner's Dilemma is a non-zero-sum, non-cooperative, and simultaneous game in which two players may each "cooperate" with or "defect" from the other player. In the iterated prisoner's dilemma (IPD) [1], the game is played repeatedly. As a result, players have the opportunity to "punish" each other for previous uncooperative play. The IPD is closely related to the evolution of trust because if both players trust each other they can both cooperate and avoid mutual defection. We have modeled the direct interaction using IPD[1].

Witness Interaction is modeled by the Generalized Prisoner's Dilemma (GPD). GPD is a two-person game which specifies the general forms for an asymmetric payoff matrix that preserves the social dilemma [2]. GPD is compatible with client/server structure where one player is the client and the other one is the server in each game. The decision of the server alone determines the ultimate outcome of the interaction.

4.3 Cooperation and Defection

We define two kinds of **Cooperation** and **Defection** in our environment: (1) Cooperation/Defection in Direct Interaction (CDI/DDI) and (2) Cooperation/Defection in Witness Interaction (CWI/DWI).

CDI/DDI have different interpretations depending on the context. For example, in e-commerce, defection in an interaction can be interpreted as the agent not satisfying the terms of a contract, selling poor quality goods, delivering late, or failing to pay the requested amount of money to a seller [7]. CWI means that the witness agent provides a reliable rating for the asking agent regarding the queried agent. In contrast, DWI means that the witness agent provides an unreliable rating for the asker agent regarding the queried agent.

5 Agent Model

This section presents two types of trust variables that assist agents in determining with whom they should interact. Furthermore, three policy types will be presented: direct interaction policy, witness interaction policy, and connection policy which assist agents in deciding how and when they should interact with another agent.

[1] Our work is different from the well-known trust game appeared in the game theory literature.

5.1 Trust Variables

Based on the aforementioned cooperation/defection explained in section 4.3, two modeled dimensions of trust are proposed. The motivation for having two trust variables is that we believe trustworthiness has different independent dimensions. For instance, an agent who is trustworthy in a direct interaction is not necessarily trustworthy in a witness interaction.

Each trust variable is defined by $T_{i,j}(t)$ indicating the trust rating assigned by agent i to agent j after t interactions between agent i and agent j, with $T_{i,j}(t) \in [-1, +1]$ and $T_{i,j}(0) = 0$. One agent in the view of the other agent can have one of the following levels of trustworthiness: *Trustworthy*, *Not Yet Known*, or *Untrustworthy*. Following Marsh [6], we define an upper and a lower threshold for each agent to model different levels of trustworthiness. The agent i has its own upper threshold $-1 \leq \omega_i \leq 1$ and lower threshold $-1 \leq \Omega_i \leq 1$. Agent j is *Trustworthy* from the viewpoint of agent i after t times of interactions if and only if $T_{i,j}(t) \geq \omega_i$. Agent i sees agent j as an *Untrustworthy* agent if $T_{i,j}(t) \leq \Omega_i$ and if $\Omega_i < T_{i,j}(t) < \omega_i$ then the agent j is in the state *Not Yet Known*.

Direct Interaction Trust (DIT). Direct Interaction Trust (DIT) is the result of CDI/DDI. Each agent maintains $DIT_{i,j}(t)$ variables for the agents with which they have had direct interactions. We used the following trust updating scheme motivated by that described in [14]:

$$DIT_{i,j}(t+1) =$$
$$\begin{cases} DIT_{i,j}(t) + \alpha_D(i)(1 - DIT_{i,j}(t)) & DIT_{i,j}(t) > 0 \text{ , } CDI \\ (DIT_{i,j}(t) + \alpha_D(i))/(1 - min(|DIT_{i,j}(t))|\,, |\alpha_D(i)|) & DIT_{i,j}(t) < 0 \text{ , } CDI \\ (DIT_{i,j}(t) + \beta_D(i))/(1 - min(|DIT_{i,j}(t))|\,, |\beta_D(i)|) & DIT_{i,j}(t) > 0 \text{ , } DDI \\ DIT_{i,j}(t) + \beta_D(i)(1 + DIT_{i,j}(t)) & DIT_{i,j}(t) < 0 \text{ , } DDI \end{cases}$$

Where $\alpha_D(i) > 0$ and $\beta_D(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively for updating of the direct interaction trust variable of agent i. The value of $DIT_{i,j}(t)$, ω_i^{DIT} and Ω_i^{DIT} determine that the agent j is either *trustworthy*, *Not Yet Known* or *Untrustworthy* in terms of direct interaction from the perspective of agent i.

Witness Interaction Trust (WIT). Witness Interaction Trust (WIT) is the result of the cooperation/defection that the neighbors of an agent have with the agent regarding witness interaction (CWI/DWI). Agent i maintains a $WIT_{i,j}(t)$ variable for the agent j from whom it has received witness information. The updating scheme of $WIT_{i,j}(t)$ is similar to the one presented for $DIT_{i,j}(t)$ but CDI and DDI should be replaced by CWI and DWI respectively and $\alpha_D(i) > 0$ and $\beta_D(i) < 0$ is replaced with $\alpha_W(i) > 0$ and $\beta_W(i) < 0$ respectively. Where $\alpha_W(i) > 0$ and $\beta_W(i) < 0$ are positive evidence and negative evidence weighting coefficients respectively for updating of the witness interaction trust variable of agent i. The value of $WIT_{i,j}(t)$, ω_i^{WIT} and Ω_i^{WIT} determine that the agent j is either *Trustworthy*, *Not Yet Known* or *Untrustworthy* in terms of witness interaction from the perspective of agent i.

5.2 Agent Policy Types

The perceptions introduced above allow agents to determine the trustworthiness of other agents. Policies make use of agent perceptions, trust and reputation models in order to decide upon the set of agents with which they will interact and in what ways they will interact. Policies may cause the agent interaction neighborhood to change, for example. While the testbed is extensible, several policy classes have been defined for the research reported here; they are explained in the following subsections.

Direct Interaction Policy (DIP). This type of policy assists an agent in making decisions regarding its direct interactions.

Witness Interaction Policy (WIP). This type of policy assists an agent in making two categories of decisions related to its witness interactions. First, agents need to decide how to provide the witness information for another agent on receiving a witness request. Should they manipulate the real information and forward false witness information to the requester (an example of defection) or should they tell the truth? The second decision is related to when and from whom the agent should ask for witness information.

We defined two sub witness interaction policies: Answering policy (AP) and Querying policy (QP). The former covers the first category of decisions mentioned above while the latter is for the second category.

Connection Policy (CP). This policy type assists an agent in making decisions regarding whether it should make a request for connection to other agents and whether the agents should accept/reject a request for a connection.

Disconnection Policy (DP). DP aids an agent in deciding whether it should drop a connection to a neighbor or not.

5.3 Experimentally Evaluated Policies

This section described policies that were evaluated experimentally.

Direct Interaction Policies. Three kinds of DIPs used in our experiments are: Always Cooperate (AC), Always-Defect (AD), and Trust-based Tit-For-Tat (TTFT). Agents using the AC policy for their direct interactions will cooperate with their neighbors in direct interactions regardless of the action of their neighbor. In contrast, agents using the AD policy will defect in all neighbor interactions. Agents employing TTFT will start with cooperation and then imitate the neighbors' last move as long as the neighbors are neither trustworthy nor untrustworthy. If a neighbor is known as untrustworthy, the agent will defect and if a neighbor is known as trustworthy, the agent will cooperate with it.

Connection Policies. Three kinds of connection polices are used in our experiments: Conservative (C), Naive (N), and Greedy (G). Each of these policies has a property called the Socializing Tendency (ST). ST affects decisions for making a connection request and the acceptance of the connection request. All three connection policies use Algorithm 1 with different ST values.

According to Algorithm 1, any connection request from another agent will be accepted regardless of the value of ST but the agent will acquire unvisited agent IDs if its number of neighbors is less than ST. In our experiments the value of ST is 5, 15, and 100 for Conservative, Naive, and Greedy connection policies respectively. The motivation for these values is that malicious agents will tend to be greedy and try and exploit a large number of agents; trust-aware agents, like their human counterparts, will tend to have a small circle of trusted agents.

Algorithm 1 Connection Policies

```
{CRQ is a queue containing the connection requests}
if CRQ is not empty then
    j = dequeue(CRQ)
    connectTo(j)
end if
if $size(neighborhood) < ST$ then
    j = get unvisited agent from list of all known agents
    if $\exists j \neq null$ then
        requestConnectionTo(j)
    end if
end if
```

Witness Interaction Policies. Three kinds of answering policies are modeled: Honest (Ho), Liar (Li), and Simpleton (Si). All these sub-policies use the pseudocode presented in Algorithm 2 while differentiating in the assignment of opinion variable (refer to * in Algorithm 2). The asterisk should be replaced by $DIT_{i,j}(t)$, "$-1 * DIT_{i,j}(t)$", or 1 for Honest, Liar, or Simpleton policy respectively. An agent employing the Liar policy gives manipulated ratings to other agents by giving high ratings for untrustworthy agents and low ratings for trustworthy ones. The Simpleton policy ranks all other agents as trustworthy but the Honest policy always tells the truth to everyone. CWI/DWI will be sent based on whether the forwarding opinion agrees with the internal trust value of an agent or not. If the difference between them is less than the Discrimination Threshold (DT), an agent will send CWI otherwise DWI is sent. We can therefore say that: Liar always defects, Honest always cooperates, and Simpleton sometimes defects (by rating high untrustworthy agents) and sometimes cooperates (by rating low trustworthy agents) in providing the witness information. In the experiments reported here DT is set to 0.25.

Algorithm 2 Answering Policy

```
if receiving a witness request about $j$ from $k$ then
    $opinion = *$
    send opinion to $k$
    if $|opinion - DIT_{i,j}(t)| < DT$ then
        Send CWI to $k$ after $T_W$ time steps
    else
        Send DWI to $k$ after $T_W$ time steps
    end if
end if
```

By use of the querying policy presented in Algorithm 3, the agent asks for witness information from its neighbors regarding one of the untrustworthy agents which has already interacted with the given agent. As a result, the agent can understand which neighbors are capable of detecting untrustworthy agents.

Disconnection Policies. We have experimentally evaluated three kinds of disconnection policies: Lenient (Le), Moderate (Mo), and Strict (St). An agent will

Algorithm 3 Querying Policy

```
{BlackList: a list of known untrustworthy agents in terms of direct interactions}
if BlackList is not empty then
    $j$ = select randomly $j$ from BlackList
    Ask for witness information about $j$ from all neighbors
end if
```

never drop a connection when using the Lenient policy. An agent that uses the Moderate policy will disconnect from the neighbor known as an untrustworthy agent in terms of direct interaction. An agent employing the Strict connection policy disconnects from the neighbor which is known to be untrustworthy either in direct interactions or in witness interactions.

6 Experiments

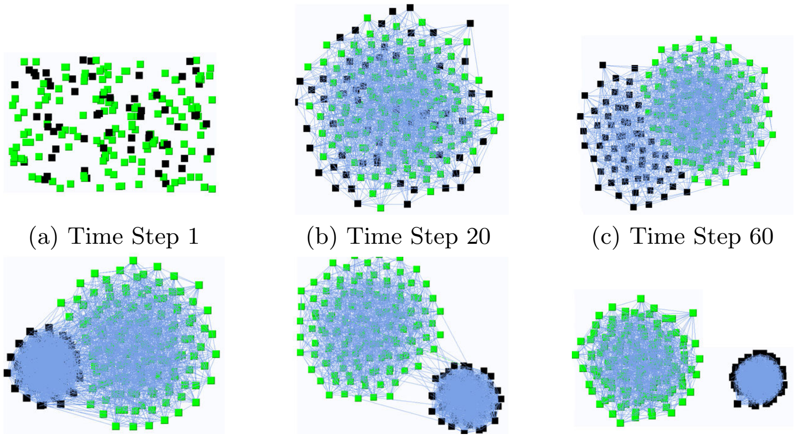
We have empirically analyzed our agent types at both microscopic and macroscopic levels. On the macro level, we studied how society structure changes over the course of many interactions. On the micro level, the utility of agents is examined. $\overline{U_{AT}(i)}$, the average of utilities for agents with the type of AT at time step i, is calculated by: $\overline{U_{AT}(i)} = \frac{\sum_{a \in AT} U_{Avg}(a,i)}{N_{AT}}$, where $U_{Avg}(a,i)$ is the average of utility of agent a over its interactions at time step i and N_{AT} is the total number of agents in the society whose type is AT. The utility of each interaction is calculated as follows: If agent i defects and agent j cooperates, agent i gets the Temptation to Defect payoff of 5 points while agent j receives the Sucker's payoff of 0. If both cooperate each gets the Reward for Mutual Cooperation payoff of 3 points, while if both defect each gets the Punishment for Mutual Defection payoff of 1 point. We have used the agent types presented in Table 1 for all experiments. In this paper, we use the same experimental values for our trust models as used by Yu and Singh in [14].

Name	Naive	Malicious	Trust-Aware(TA)	Trust-Aware$^+$(TA^+)
Trust	-	-	DIT	DIT&WIT
DIP	AC	AD	TTFT	TTFT
CP	N	G	C	C
DP	Le	Le	Mo	St
AP	Si	Li	Ho	Ho
QP	-	-	-	QP

Table 1. Agent Types and Specifications

Experiment 1. We run the simulation with the population size of 200 agents where TA agents cover 66% of population and the rest are Malicious agents. The objective of this experiment is to understand whether cooperation emerges between TA agents while they isolate themselves from Malicious agents.

Different stages of this simulation are depicted in Figure 1, where TA agents and Malicious agents are in green (light gray in white-black print) and in black respectively. Starting from an initially unconnected society (Figure 1a) Malicious agents are quickly discovered (Figure 1c) and are completely isolated by time step 400 (Figure 1f).

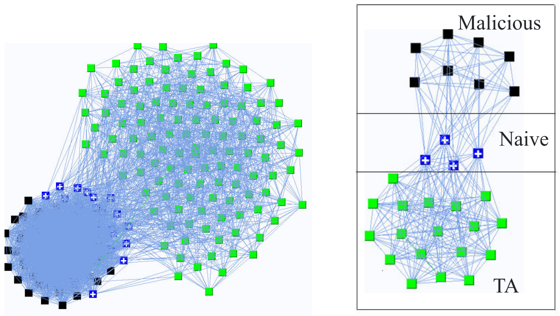


(a) Time Step 1 (b) Time Step 20 (c) Time Step 60

(d) Time Step 180 (e) Time Step 340 (f) Time Step 400

Fig. 1. Structural changes of Agents Society in Experiment 1

Experiment 2. We run 200 agents where 55%, 11% and 34% of population are TA, Naive and Malicious agents respectively. The structure of the agent society after 400 time steps is presented in Figure 2a. Malicious and Trust-Aware agents are shown with the same colors of the previous experiment and blue squares with white "+" represent Naive agents. With the introduction of Naive agents, we could not achieve separation of Malicious and TA agents seen in Experiment 1. Since TA agents perceived Naive agents as trustworthy agents in direct interaction so they maintain their connections with Naive agents. On the other hand, since Naive agents accept all connection requests and do not drop any connections, they will be exploited by Malicious agents. As illustrated in Figure 2a, TA agents are connected indirectly to Malicious agents by means of Naive agents. Figure 2b shows Naive agents acting a buffer between the 2 other agent communities for a 30 agent simulation.



(a) 200 Agents (b) 30 Agents

Fig. 2. The Final Society Structure in Exp. 2

Figure 3 shows the $\overline{U}$ of each agent type over the course of the simulation. $\overline{U}_{TA}$ increases over the simulation with small fluctuations. The more $\overline{U}_{TA}$ gets close to 3, the higher the proportion of interactions of TA agents are mutual cooperation. $\overline{U}_{Malicious}$ is increasing due to connecting to more Naive agents.

The $\overline{U}_{Naive}$ drops over the course of simulation since the number of their connections with Malicious agents increases. All three graphs stabilize before time step 350, which is the result of not establishing new connections by any agents. Not requesting any connections can be the result of reaching the ST threshold (e.g., Naive and Trust-Aware) or scanning all of the agents (e.g., Malicious agents).

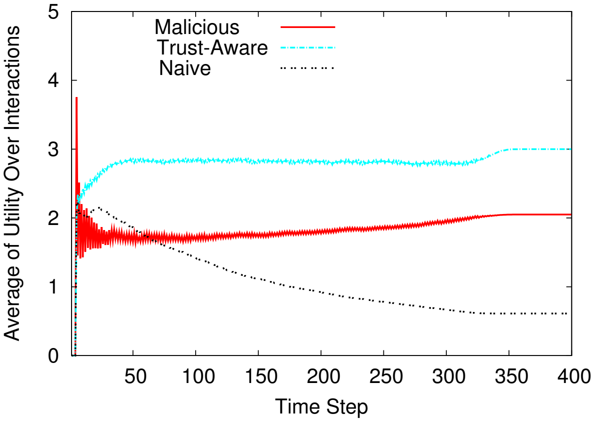


Fig. 3. $\overline{U}$ of agent types over simulation

Experiment 3. This experiment intends to show the effect of a varying proportion of Naive agents. We have run five simulations of 200 agents with different proportions of Naive and Trust-Aware agents while maintaining Malicious agents unchanged as shown in Table 2.

Agent Type	Population				
	Pop1	Pop2	Pop3	Pop4	Pop5
Malicious	34%	34%	34%	34%	34%
Naive	0%	11%	22%	33%	44%
Trust-Aware	66%	55%	44%	33%	22%

Table 2. Population Distributions of Experiment 3

Figure 4 presents $\overline{U}$ of each agent type at time step 400 for each of the runs. By increasing the proportion of Naive agents, $\overline{U}_{Malicious}$ increases considerably although the proportion of Malicious agents is unchanged. $\overline{U}_{TA}$ in all runs stays at 3 indicating that the proportion of Naive agents does not influence $\overline{U}_{TA}$. $\overline{U}_{Naive}$ increases slightly because Malicious agents have more choices to connect to Naive agents and to satisfy their ST threshold. For Pop5, the $\overline{U}_{Malicious}$ exceeds that of TA agents. In such societies, where malicious agents are unbounded in terms of their ability to exploit other agents,there is no incentive to be a Trust-aware agent since Malicious agents have better utility. That is all the outcome of having a high proportion of Naive agents in the society.

Experiment 4. We run 200 agents where 55%, 11% and 34% of the population are Trust-Aware$^+$ (TA$^+$), Naive and Malicious agents respectively. The structure of the agent society at three points in the simulation are presented in Figure 5. Malicious and Naive agents are shown with the same colors of previous experiments and TA$^+$ agents are presented in green. It is interesting to

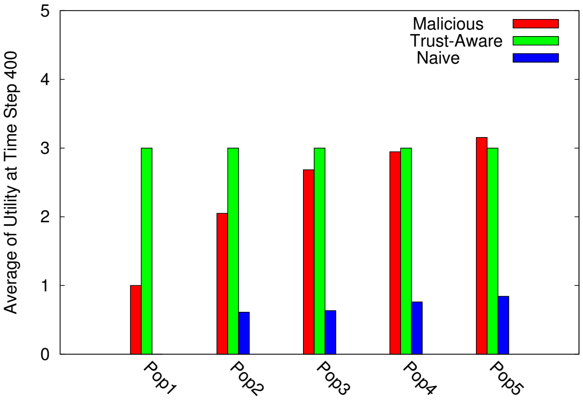
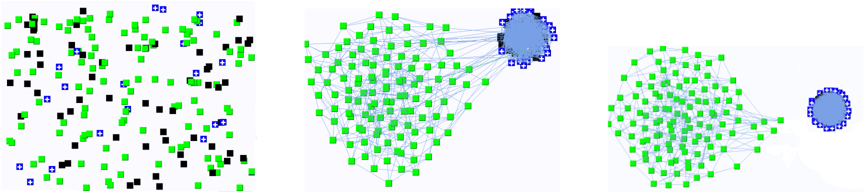


Fig. 4. $\overline{U}$ for five runs of Exp. 3

observe that Naive and Malicious agents are isolated from the TA$^+$ agents. By using multi-dimensional trust (DIT and WIT) and the Strict disconnecting policy, TA$^+$ agents could identify both Malicious and Naive agents to isolate them from their community. Naive agents are detected based on their failure to provide the appropriate witness information while Malicious agents are recognized by their defections in direct interactions.



(a) Time Step 1

(b) Time Step 180

(c) Time Step 400

Fig. 5. Structural changes of Agents Society in Experiment 4.

7 Conclusion and Future Work

The isolation of untrustworthy agents from a society of agents is considered one of the main objectives of trust models [14]. Experiment 1 demonstrates that malicious agents can be isolated using DIT when naive agents are absent. Experiments 2 and 3 demonstrate how the proportion of naive agents affects the utility of malicious agents and society structure. When this proportion exceeds some threshold, malicious agents have the best utility in the society and consequently there is no incentive for trust-aware agents to stay trustworthy. In contrast, they are motivated to be malicious to exploit naive agents too. As shown in experiment 3, it is important for a society to limit the ability of any agent to exploit another agent. Experiment 4 shows how adding WIT allows naive agents to be detected. In this sense, TA$^+$ agents assessed the ability of their neighbors in detecting malicious agents. Those agents which fail in this assessment turn out to be naive agents.

Naive agents strongly degrade the value of DIT in trust-aware agent societies. Our results demonstrate that naive agents help malicious agents survive by cooperating with them directly (by providing good services) and indirectly (by giving a good rating for them). The proposed model demonstrates that trust-aware agents need multi-dimensional trust models to separate malicious and naive agents from the trustworthy community and would benefit from maintaining networks for each dimension of trust.

We plan to extend the proposed trust model for other sources of information such as observed interactions and modeling agents that are naive in observing the results of interaction. It would be interesting to see the effect of naive agents in reputation variable (systems) where the ratings regarding the specific agents will be gathered from naive neighbors.

References

1. Robert Axelrod. *The Evolution of Cooperation.* New York: Basic Books, 1984.
2. Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04*, pages 102–111, New York, NY, USA, 2004. ACM.
3. Karen K. Fullam, Tomas B. Klos, Guillaume Muller, Jordi Sabater, Andreas Schlosser, Zvi Topol, K. Suzanne Barber, Jeffrey S. Rosenschein, Laurent Vercouter, and Marco Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *AAMAS '05*, pages 512–518, New York, NY, USA, 2005. ACM.
4. Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
5. Sirkka L. Jarvenpaa, Noam Tractinsky, and Michael Vitale. Consumer trust in an internet store. *Inf. Technol. and Management*, 1(1-2):45–71, 2000.
6. S. Marsh. Formalising trust as a computational concept, 1994.
7. Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *Knowl. Eng. Rev.*, 19(1):1–25, 2004.
8. Jordi Sabater and Carles Sierra. Regret: A reputation model for gregarious societies. In *Fourth Workshop on Deception Fraud and Trust in Agent Societies*, pages 61–70, 2001.
9. Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.
10. Amirali Salehi-Abari and Tony White. Towards con-resistant trust models for distributed agent systems. In *IJCAI '09: Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, pages 272–277, 2009.
11. Michael Schillo, Petra Funk, and Michael Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*, 14(8):825–848, September 2000.
12. W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model. In *AAMAS '05*, pages 997–1004, New York, NY, USA, 2005. ACM.
13. Bin Yu, M.P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on*, pages 1–10, 30-31 Aug. 2004.
14. Bin Yu and Munindar P. Singh. A social mechanism of reputation management in electronic communities. In *CIA '00*, pages 154–165. Springer-Verlag, 2000.
15. Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *AAMAS '03*, pages 73–80, New York, NY, USA, 2003. ACM.