WHO CAN YOU TRUST: DEALING WITH DECEPTION

Michael Schillo[1], Petra Funk[2], Michael Rovatsos
Multi-Agent Systems Group, DFKI GmbH, Im Stadtwald,
D-66123 Saarbrücken, Germany
{schillo, funk, rovatsos}@dfki.de

Abstract

Trust is one of the most important concepts guiding decision making and contracting in human societies. In artificial societies, this concept has until recently been neglected. The inherent benevolence assumption implemented in many multi-agent systems can have hazardous consequences when dealing with deceit in open systems. Therefore we present a formalisation of trust, an algorithm for evaluating it and an application scenario. This approach can be applied to avoid the use of trusted third parties. It includes social behaviour for implementing punishment without empowered authorities. We argue that by adopting our ideas in open systems, agents can autonomously deal with deception and identify trustworthy parties. Our approach to this problem is twofold: Agents can observe the behaviour of others and thus collect information for establishing a trust model. Still, agents cannot rely exclusively on the information gathered. In order to adapt quickly to a new or rapidly changing environment and to ensure fast responses to new opportunities, we enable agents to make use of observations from other agents (*witnesses*).

However, testimonial evidence from such witnesses may be brittle, as witnesses may have differing motives and may try to deceive other agents about their true observations. Our solution is based on a stochastic process. For its evaluation, we chose a game theoretic abstraction of contracting between agents. The setting consists of a number of phases, which we consider also natural for contracting. The interaction of agents improves if they use communication about observations and apply the trust model to evaluating new contract partners. We demonstrate practical relevance of our ideas with a direct mapping between our scenario and electronic commerce.

1. Introduction

Trust is one of the most important social concepts to help agents (be they human or artificial) in identifying friendly partners within their social environment. In artificial societies, such as they are provided e. g. within multi-agent systems, the real-world assumptions about selfish, antisocial, or unreliable society members are currently not or only fragmentarily included. Instead, most multi-agent systems are founded on benevolence assumptions, including trustworthiness or reliability. With the growth of network services through the Internet, we find ourselves in hybrid artificial societies, where real-world assumptions about potential partners must be provided in terms of authentication mechanisms. Secure network protocols and cryptography do not guarantee reliable services by a service provider in the net. Therefore, a conceptualisation of trust and how it can be used in artificial societies is essential to dealing with deception in virtual worlds.

We propose an approach, which includes two important information-gathering activities for the agent: collecting information about the trustworthiness or reliability of potential partners by observation and interviewing other agents about their observations. They store this information in their knowledge base. Facts gathered by the agents through observation are included into the trust model directly. Testimonial evidence from interviews may be brittle, as witnesses may have diverging motives and may try to deceive other agents about their true observations. Thus, every

[1] The author is funded by the Studienstifung des deutschen Volkes.
[2] The author is funded by the European Commission (Platform project, contract number PL 97-2710).

agent is confronted with noise in the information and also with the possibility that the source of the information itself is trying to bias the recipient. Our formalisation of trust is able to deal with such not-so-trustworthy witnesses, enabling the agent to distinguish potential partners from undesired contracting parties.

Some of the ideas presented here have been inspired by the ideas of Bazzan and colleagues (Bazzan et al. 1997). They enriched the classical prisoner's dilemma ((Axelrod 1984), (Luce and Raiffa 1957)) with agents with social attitudes. Players are either rational egoists or generous altruists. The notion of moral sentiments was introduced to establish a method for altruistic agents to support each other. Their results demonstrate that pure rationality (attributed to the egoistic agents), in terms of never co-operating, does not pay off in the long run for neither the single egoist nor its social group. While Bazzan and colleagues provide their agents with knowledge about the structure of the society, we chose to adopt the paradigm "*The world does not come labelled*" (Edelmann 1987), i.e. our agents only know their own social attitude, they do not know that of other agents except for observed facts. In our scenarios the agents' activities are not strictly conform with their social attitudes; we introduce fuzziness into these social roles, by supplementing each agent with a probability factor for role conformity. We demonstrate how our agents use the formalisation of trust in order to deal with deceit by extending the prisoner's dilemma game with a partner selection phase before the actual game takes place. Based on testimonial evidence and the TrustNet model, agents negotiate about their potential partners for the game. During negotiation the agents are allowed to communicate with other agents to find out about their opinions on the altruism and the trustworthiness of other players. In (Schillo and Funk 1998) we discuss this negotiation protocol in more detail.

This research is under development in the context of the recently created interdisciplinary field of *socionics* (Malsch et al. 1996). Socionics is concerned with studying the relevance of sociological concepts and theories for computer science and vice versa. We have been strongly influenced by this endeavour and find the use of sociological concepts like trust, deceit, etc. considerably helpful to model artificial societies and automated contracting on behalf of human users. The results reported hear are part of one of the authors final thesis (Schillo 1999).

The paper is structured as follows: In Section 2 we discuss other approaches to modelling trust and why we reject them. The environment for experiments with the disclosed prisoner's dilemma game and the TrustNet as well as the extensions for partner selection are detailed in Section 3. Then, in Section 4 we present the algorithm to establish a TrustNet. We give experimental results in Section 5 and round up the presentation by concluding remarks and an outlook onto future work in Section 6.

2. On the use of Trust Values

The essence of our approach to modelling trust is the insight that trust is not an event in the sense of probability theory but rather a degree of how high some peer's honesty is estimated. This seemingly trivial observation is overlooked by many approaches employing probabilistic methods to model trust. The purpose of probabilities is to express the frequency with which events occur and not to provide for a qualitative model of someone's behaviour (although the two are obviously strongly connected). Starting from this observation, we briefly describe in this section why we oppose to direct combination of trust values.

2.1. Common Mistakes in Combining Information from Witnesses

If an agent A is to compute its trust $T(A,B)$ towards an agent B numerically, it is desirable for A to include any information that is communicated to A by witnesses W_1, W_2, …, W_n about the honesty of agent B. The honesty of B is the ratio r between the number of rounds, in which B enacted what it committed itself to and the number of total rounds. This means that A receives a

number of values r_1, …, r_n from witnesses and wishes to combine these to get an approximation r' of r. Many approaches at this stage compute r' as the average of r_1, ..., r_n, thereby neglecting the fact that the observations of B's moves are *not* independent, because the subsets of B's actions observed by W_1, W_2, ..., W_n are not necessarily disjoint. This is a phenomenon that Pearl calls „correlated evidence" and to which he ascribes the effect that „[extensional systems] will produce the same conclusions whether the weights originate from identical or independent sources of information." Figure 1 shows an example of such correlation that yields unreasonable results when evidence is combined: the elliptical nodes represent the agents A and W_1, W_2, .., W_n, whereas the rectangles on the right hand side represent the individual actions of agent B, which cause the witnesses to assign a certain honesty to B depending on the deceptive/truthful utterances of B that they have observed (those observations are shown by the unlabelled edges). The labelled edges to the left of the witnesses describe the trust values $T(W_i, B)$ communicated to A. It can clearly be seen that taking r' as the average of $T(W_1,B)$, $T(W_2, B)$ and $T(W_3, B)$ yields $r'=.75$, a value twice as high as the actual r=.375, while A would have been able to reconstruct a significantly more precise approximation if it only had used all the information that the witnesses had. Hence this way of combining the evidence seems highly unreasonable.

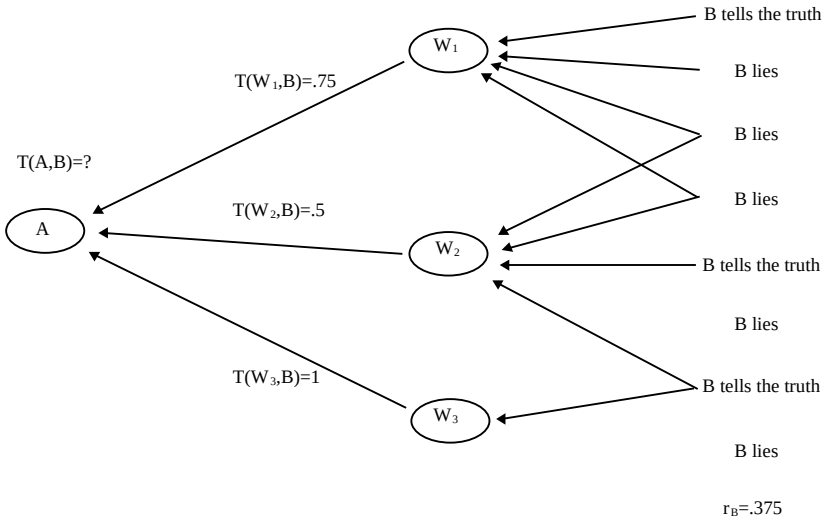

Figure 1 Combination of trust values from multiple witnesses

The solution to this problem is to uncover the actual observations that led to the individual $T(W_i, B)$ values and to propagate values along causally dependent as in Bayesian inference. Unfortunately, however, A has no means to access the observations concerning B's actions towards W_1, W_2, …, W_n other than by what is said about them by the possibly deceitful witnesses. Therefore the full net structure cannot be constructed by A in a reliable fashion and the correlations cannot be uncovered, so we show an alternative solution.

2.2. Trust and Transitivity

When using trust values obtained from a witness W it is reasonable to believe $T(W, B)$ only to the degree that we trust W i.e. $T(A, B)$ should depend on $T(A, W)$ as well as on $T(W, B)$. At first glance it seems straightforward to make use of this transitivity, since most people would agree that „if Frank trusts Jack and Jack tells Frank „Jim is trustworthy" then Frank should trust Jim" is a sensible line of reasoning. However, „being trustworthy" is not a binary proposition (event). If it were, we could combine $T(A, W)$ and $T(W, B)$ to compute $T(A, B)$ by using transitivity and Bayes' rule:

$$P(T(A,B)) = P(T(A,W) \cap T(W,B)) = P(T(W,B) \mid T(A,W))P(T(A,W)))$$

So if e.g. *T(A, W)=0.25* and *T(W, B)=0.5*, then all that A knows is that with a probability of *0.75* the actual trust value *W* has towards *B* is *different* from *0.5* but it is not obvious at all how this should affect *A*'s model of *B*. To illustrate this fact, Figure 2 shows the case in which *A* has made some observations about the behaviour of *B* itself (which yielded a *direct* trust value $T_d(A, B)=1.0$), and that it also has a model of how honest *W* is due to *W*'s previous behaviour ($T_d(A, W)=0.25$). In which way should *T(W,B)=0.5* be taken into account when it is received by *A*, given that the observations on which *T(W, B)* and $T_d(A,B)$ are based might overlap and given that W may have lied about the *real* value of *T(W,B)*?

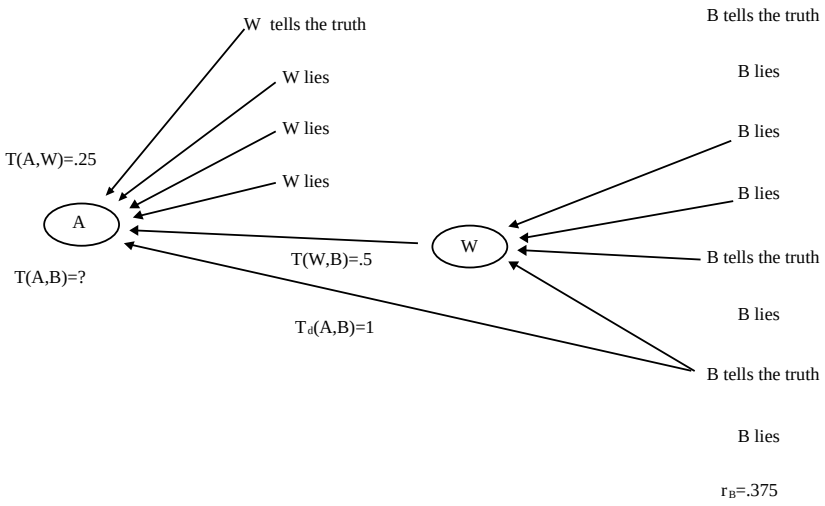

Figure 2 Transitive trust estimation

The reader is encouraged to try any reasonably simple combination of weight-combining functions (such as max, min and arithmetic operations) to compute *T(A,B)* from *T(A, W), T(W, B)* and $T_d(A, B)$. We are quite confident to say that no such function will yield satisfactory results, because (as mentioned in 2.1) changing and combining local weights in such a causal system cannot capture all the correlations that lead to the behaviour of the variables. In Section 4 we describe our solution to these problems, based on the analysis of „lying" as a stochastic process, a technique which we view as an adequate way of implicitly reconstructing witness observations in order to alleviate the problem of „correlated evidence".

3. The experimental environment

3.1. Disclosed Prisoner's Dilemma with Partner Selection

In order to evaluate our concept of trust, we chose an extension of the prisoner's dilemma game enhanced with a partner selection phase (Schillo and Funk 1998). Agents do not a priori know about the social attitude of potential partners, but they can learn from experience and observation. We model egoistic and altruistic personality profiles, but we provide them with a fuzzy factor: each agent plays according to its social attitude with a given probability. Isolation from future games can be a (punishing) result from constantly playing defect. Since we focus here on the trust model used to finding partners and evaluating observations from the game, we use a standard pay-off matrix and do not go into detail about the game itself. Due to space limitations we would like to refer readers unfamiliar with prisoner's dilemma games to the literature ((Axelrod 1984), (Luce and Raiffa 1957)).

The disclosed prisoner's dilemma with partner selection can briefly be described in 5 steps:

1. Each player pays a stake.

2. Pairs players are determined by negotiation. and agents can deceive others about their intentions.
3. The game is played.
4. The results are published. Due to limited perception, each agent gets only the results of a subset of all players.
5. The prizes are paid.

The *first step* should be clear, agents dispose of a limited amount of points; if an agent loses all its points, it has to retire from the game.

For the *second step*, we introduce a contract net-like protocol. The protocol is executed until each player had the chance to find a partner. For each new round, the agents are sorted by random to guarantee equal chances to all players. The first agent in this list is elected manager. The manager announces its intention for a game of prisoner's dilemma. All other agents answer whether they want to play with him and what their intention would be. The manager then may choose among them his partner for the game. If he has limited knowledge about the social attitudes of the bidders, he can make enquiries about them. Communication during this phase is strictly restricted in the following manner: the manager asks an agent Q, the contents is the name of a bidder S. Q's answer in turn is a list of observations that Q has made with S. Q does not have to be honest in this communication. Therefore the manager evaluates the information gathered by these short interviews. If he has chosen a game partner, we allow this agent to communicate in the same fashion. This way the bidders are not disadvantaged by not having been a manager in this round.

In the *third step* the agents play the game. This is a conventional game of prisoner's dilemma. The agents do not have to play according to their announcements in phase two.

The results of the games are published in *step 4*, enabling agents to observe other's behaviour in terms of frankness, reliability and trustworthiness. Each agent is only told the results of agents in its direct neighbourhood. With this information the agent can then update and calibrate their TrustNets in order to use it for the next partner selection.

In the final phase, *step 5*, the agent receive their prizes for their moves.

If an agent is not trustworthy in terms of sticking to its announced move, agents in a defined neighbourhood notice this. Therefore it can and will happen, that agents may at first gain from abusing the ignorant members of an agent society. In our experiments, we will demonstrate that after a number of rounds, such agents are excluded from playing, because they are no longer trusted.

3.2. Agent Profiles

The described setting allows for modelling a range of agent behaviours. This includes the traditional "benevolent agent" by using the configuration *1.0* for both altruism and honesty and a malicious agent with value *0.0* for both variables.

3.3. Practical Relevance

In this section we show the connection between the described setting and real world applications, were trust and finding out whom to trust is an essential goal. We chose the *electronic commerce* application to demonstrate this.

In electronic commerce as well as in our scenario, agents encounter other agents while interacting autonomously in order maximise their performance. They may achieve this by co-operating with their contract partners in the long run, or try to make "fast cash" and exploit others. Additionally, agents can offer contracts announce intentions that they will not exhibit at commitment

time. Electronic commerce is unbound by national frontiers and therefore free from national authorities. In traditional trading, however, these authorities guaranteed the fulfilment of agreements or punishment by their power. This means of stability is not available in electronic commerce (and for many business partners it will not be for some time). In our setting, the agents will not be punished if they deceive their game partners, but we enable them to track very fast which agent is behaving in a deceitful way so that they perform still very well. To model the conflict between behaving co-operative or deceitful and the respective outcomes, we use the prisoner's dilemma with the addition that agents need to pay before they are allowed to join a round of the game. This results in a loss of score, if they do not find some one that will join them in a game. Thus, there is an indirect punishment to malicious behaviour, which can be described as *virtual peer pressure*.

The idea of the peer pressure is based on the natural assumption that there is some communication in the world about the players in the market, e.g. press, personal communication, etc. Again this communication is in both settings not completely objective (to avoid the ambiguous term *trust*). In our experiments, agents will to some extend be *lying*. As they do not want to be found making up false data, they will try to bias the information they communicate. This can be achieved by leaving out the data they have observed that does not suit their intentions. We assume that they are motivated to make their competitors look *not* trustworthy and *not* suited for games with high pay-off in order to discourage players to choose other agents than themselves.

Electronic commerce and our setting have many features in common and they are a dangerous place for agents relying on benevolent contractors. Agents which succeed in our setting, will be interesting to investigate, when looking at the real world application. Additionally, we currently envision two other applications of the proposed mechanism. We believe that in the future mechanism like „Cookies" will be of more importance in the Internet world. The demonstrated solution will be an easy to implement way to find out say, which website you can trust to not abuse the information of the Cookie set on your machine (e. g. for using personal data). More serious is a similar application where our approach is just as applicable: the problem of differentiating between friendly and malicious hosts or migrating programs (e.g. (Beth et al. 1994), (Sander and Tschudin 1998)).

4. The TrustNet Implementation

In the described setting agents will naturally aim at collecting as much data on other agents as quickly as possible. An agent is aware of the fact that other agents may have a range of different behaviours of which only a few will be desirable to experience in a game of prisoner's dilemma. As a consequence counting on observations made by the agents do not suffice. Thus, the observations of other agents (*witnesses*) may be used. However, these witnesses may try to deceive by communicating false information.

4.1. Semantics of the TrustNet

During the game an agent has the chance to collect data of two types: the honesty with which a player announces what it will commit itself to and the way it chooses its options, altruistic or egoistic. Each agent stores this data in a graph where the nodes represent agents. Figure 3 is an example of a simple graph with only four nodes. More nodes can only be added when adding an edge from an existing node to it.

The nodes are annotated with two values, the trust, which the agent can put into the agent represented by the node and the altruism that can be expected from it. The edges carry information on the observations that the parent node agent told the owner of the net about the the child node agent. This data comprises which round was observed and which behaviour of the target agent was observed in terms of honesty and altruism. The owner of the net is represented in every

net as the root node (node *A* in Figure 3). All other nodes are descendants from the root node, as this is the node that is the source of all information. Its outgoing edges resemble the observations it made. The other edges constitute information that has reached the owner via communication.

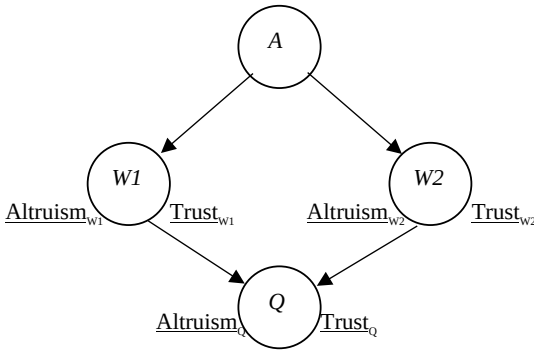

Figure 3 A simple TrustNet

4.2. Merging Information from Multiple Testimonies

There are two possible ways of collecting information on other players. First of all, every agent has the chance to observe a number of players directly. The smart agent will use a second option: interview other agents. In the second phase of the described game protocol every agent chooses another agent to play with. He can select a partner from a number of agents, which offer to play with him. If the agent does not have enough information about them, he can request information from all agents that he has met before.

Evaluating the information that a decision-maker observed by itself is rather easy. If it observed n games of an other agent (we will now call this agent the *target agent*) and the target agent behaved honest in e games, the natural approximation of the behaviour would be that the target agent will be honest in the next game with probability $p = e/n$. But how can the decision-maker combine information of a witness with the necessary weight of the trust in the witness? And, even more complicated, how can the decision-maker combine the information gathered by m agents and their corresponding m trust values? As we argued in Section 2 we cannot simply average over the weights. We will now introduce an algorithm for combining witness data that eliminates these probabilistic objections and has several advantages. We will look onto the problem from the perspective of an agent (*decision-maker*) that has to find out with which agent to play (*target agent*). The decision-maker asks other agents (*witnesses*) that have observed the behaviour of the target agent. These witnesses may have only partial information on the target agent, but our algorithm deals with this by combining many sets of partial data, like a jigsaw puzzle, to approximate the big picture. Additionally, the algorithm deals with overlapping data sets. Also the decision-maker has to deal with the fact that the witnesses may have the intention to lie about their observations and hide some information. The proposed algorithm deals with this by estimating how often the witnesses lied.

Before we present how this estimation is evaluated we will have to define *lying* in this context. First of all, what could be the motivation for a witness to lie? Any agent will want to be the one who plays with an agent of highly altruistic behaviour. There are two ways of influencing other agents in the proposed game. First, every agent can try to make other agents appear less altruistic when being asked for testimony on them. The decision-maker will then have a smaller tendency to play with them and chances are getting better that he may instead play with the witness. Second it can make look other agents less trustworthy, so that the decision-maker will tend more and more to ask it and not others about testimonies, increasing it's power.

Therefore, we find it reasonable to assume that there is a motivation to lie in the sense that lying about values of an agent consists of hiding a fact that would make the target agent look either altruistic or honest.

We denote information on the behaviour of an agent in a single game with *e* if it was honest (or *a* if it was altruistic) and *n* if it was not honest (the same for altruism). In the following we will only look at the calculations on the honesty data and come to the calculations on the altruism data later. The statement of a witness on the behaviour of a target agent is ε if it does not have information or claims to have no information. The variable *p* denotes the frequency of lying. According to this, lying is a function:

$$\text{Definition}: Lying: \{e\} \to \{e, \varepsilon\}$$

$$x \text{ a } \begin{cases} e & \text{with probability p} \\ \varepsilon & \text{with probability 1 - p} \end{cases}$$

So we expect witnesses to behave in the following way. When an agent requests information on a given target agent, it checks its own observations. It transmits all the data on dishonest and egoistic (non-altruistic) behaviour. It then looks up all data on honest behaviour. It applies the *Lying* function to every item and transmits only those where the result of the function is *e*. Thus a witness will neglect information but not tell something that is not the truth. It will not say that a target agent has played dishonest in game *x* if this was not the case. The reason for this is that the witness does not want to be observed at obviously lying. The agent that requests the information may have observed game *x* by itself. An alternative is to influence the judgement of others by providing biased information. This function is to some extend not intuitive to what humans understand by lying. The reason for still using this term is that we believe that the same mechanism can be used for more complex strategies in lying and that it covers many of the intentions expected by agents in settings like electronic commerce. We will look at other lying strategies in upcoming research.

Mathematically speaking what happens when this function is applied, is a *Bernoulli-experiment*. It is like tossing a coin that will show heads with probability *p*. In this case, showing heads corresponds with telling about the data item, tails is not telling about it. Repeating this experiment is a *Bernoulli chain*. The traditional use of a Bernoulli chain is to determine the number of heads or tails that will show on tossing a coin *n* times. We will do something different. After a witness has communicated some data, we know how many times the *lying* function has returned *e*, but we do not know how many times it returned ε. Or, in other words, we know the number of honest replies (which we also denote with *e*) but we do not know the total number of Bernoulli experiments *n*. An example of such a situation is given in Figure 4. The first two rows of data represent the information from two witnesses. As the information from the witnesses comprises also the game number, the information can be collated to a result tuple, which eliminates the problem that the data from the two witnesses might be overlapping (see Section 2).

The decision-maker can assume that this data is correct, as it assumes that the agents do not want to be caught obviously lying. But how can it find out if the witnesses have biased the reported data on game results? And if so, how much information has been hidden by them? We will now explain how exactly we estimate the hidden amount of information, judging from the trust of the decision-maker into the witnesses. As soon as we have established this, data can be collated and evaluated for any number of witnesses on target agents or different length of paths in the TrustNet. It can be used by applying the evaluation recursively from the target agent through all its ancestors up to the root node, whose honesty can be regarded as being 1).

Data on game result														Lies
Witness 1				¦		¦			X		X	X		X	?
Witness 2		¦		¦			X		X	X					?
Result		¦		¦		¦	X		X	X	X	X		X	?

Figure 4 Example for the reported data of two witnesses.

We use the following variables:

n is the amount of information the witness has on a target agent.

k is the estimation of the number of times the result of the *lying* function was ε i.e. the amount of positive data on a target agent that the witness intended to hide.

e is the number of reported *e* (where the result of the *lying* function was *e*).

p is an estimation by decision-maker of the parameter in the witnesses *lying* function. This is either the result of evaluation of its own observations or the result of a previous application of this algorithm.

By using the formula for binomial distributions it follows directly from this model that the probability for the event that the witness has lied *k* times (if we replace n by $k+e$) is:

$$P(T = k) = \frac{(k+e)!}{k!\,e!}(1-p)^k p^e$$

Now we can infer what the expectation value is, i.e. how often can we expect, did the witness lie. We need to distinguish two cases.

Case $e > 0$: If the witness reported on some *e*, it follows directly that the expectation value EX is:

$$EX = \sum_{k=0}^{n} k \binom{k+e}{k} (1-p)^k p^e = (k+e)\,p$$

Assuming that the witness has lied *k* times ($EX = k$), this determines that $k = \frac{ep}{1-p}$. Thus we have an approximation of the number of times the witness has betrayed by leaving out information in the case that it reports of at least one positive information. Now we take a look at the more difficult case that the witness reports on zero positive information. This will occur if the witness lies very often (its *p* is large) or if it has only very few positive information on the target agent. In any case we would like to infer something from its testimony.

Case $e = 0$: If we apply $e = 0$ to the binomial distribution equation, we get $P(T = k) = p^k$. Now we need to calculate the expectation value of this function. We will do this by determining the value E for *k*, where the area beneath the curve of the function left of E equals half the whole area beneath the function. In mathematical terms, this means that the expectation value E for *k* is

$$\frac{1}{2}\int_0^{\infty} p^k dk = \int_0^{E} p^k dk \Rightarrow k = \frac{\ln 1/2}{\ln p}.$$

As we know now how to determine the amount of hidden information, we can fill in the data in the last column of Figure 4, i. e. we can now complete the table by the number of *e* that are missing for each witness. However, we still do not know how to merge this data for a number of witnesses. We could make assumptions, e.g. assume that the unreported *e* have been observed in completely disjunct sets of games. Or assume that they have been assumed in all the same games.

Or we could take the average of the unreported *e*. As we have argued before it is not acceptable to neglect the possible, but not necessarily given, dependence of the data of the witnesses (see Section 2). We feel that the following heuristic is more reasonable.

We assume that if we look at all the data in the reported tuples, we find that it is distributed by random just as well as the data for the unreported part of the tuple. We conclude that the distribution of the unreported *e* will be similar to the distribution of the unreported *e*. In mathematical terms this means that the density of the data (the relation of the overlapping of the data in the tuples) is constant for the reported and the unreported data. The density of the reported data is only depending on known variables. The number of witnesses is known, as well as the total number of all reported entries in all tuples. Also we can determine the length of the result tuple. This will give us the density with the following equation:

$$density = \frac{matrixEntries / witnesse}{lengthoftuple}$$

Assuming that the density is the same in the unreported data, the density will determine the length of the tuple that has to be added to the already existing result tuple. We know judging from the motives of the witnesses that we can fill up this additional space with positive data on the target agent. We already know the number of agents and using the above equations we can determine the total number of the tuple entries that expected to be hidden by the witnesses.

$$density = \frac{hiddenEntries / witnesse}{additionalTupleLength}$$

Using the density equation for the reported data, this gives us an equation to determine the additional length of the result tuple:

$$additionalTupleLength = \frac{hiddenEntries / agents}{density}$$

We now have a reasonable approximation of what the witnesses tried to hide from the decision-maker (unfortunately we do not have the space in this paper to demonstrate an example of this calculation. We hope that the results will be convincing to the reader and create interest in further reading). This approximation will however, return only good values, if the estimation of the honesty of the witness is good. So the overall performance of this calculation will increase with the amount of data that is available to the decision-maker, which still is a reasonable behaviour for the algorithm.

One of the advantages of this procedure is that we do not try to propagate weights on the trustability of witnesses through the net, but reconstruct for every merge of information an approximation of what the witnesses would have said, *if* they had been completely honest about their information.

5. Preliminary Results

In this section we present one set of data that we have collected from a Java™ implementation of the TrustNet and the game protocol introduced in Section 3.1. The experiments for this research have been conducted by using the *Social Interaction Framework* (Schillo et al. 1999).

The example we briefly describe here, used the following parameters. We had agents with three different values for honesty (0.1, 0.5, and 0.99) and the same three different values for their altruism. All combinations of parameters were each represented by three agents with the ability to use the TrustNet and three with just using their own observations (the latter served as a *control group*). This adds up to 54 agents in total. All agents were allowed to observe 3 other agents during the game phase (this is about 5.7% of the population). The agents with the TrustNet were allowed

to request information from 15 agents (about one fourth of the population) in the partner selection phase.

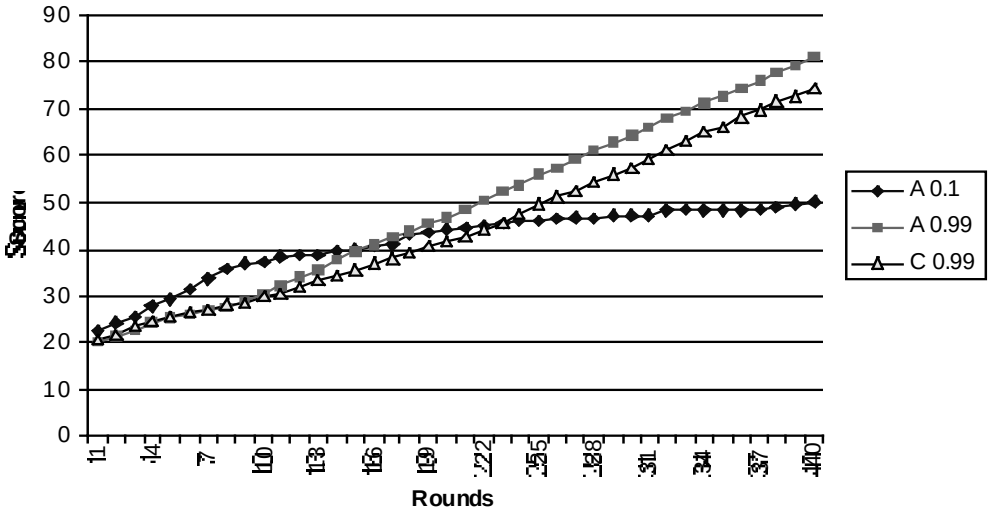

Figure 5 Example data from conducted experiments

For eligibility we have included in Figure 5 only the three most interesting groups of agents from the data. As the reader can see, the group of the agents with altruism value 0.1 (referred to as *group 3*) were most successful in the first few rounds. This is due to their constant intention to betray others and the altruists having not enough experience to be warned of their behaviour. The nine agents with altruism value 0.99 that used the TrustNet (group 1) needed approximately sixteen games to find out with whom they can play if they want to maximise their profit and reduce the risk of being betrayed. Although the control group (group 2) used all the observations they made to achieve the same knowledge, the lack of the TrustNet caused them to take about eight rounds more than group 1 to overtake the egoists. The interpretation of the performance of the groups is supported by data from other experiments.

We have also conducted experiments with a similar configuration of the agent society, but allowing the agents more observation (10% and 20% of the players). In this case, there is a tendency that the control groups perform better compared to the groups with TrustNet. We explain this by the bigger amount of information that can be observed directly, so that using the TrustNet has a smaller effect.

6. Conclusion

6.1. Summary

We presented an approach to make agents recognise beneficial contract partners by little observation and the use of witnesses. In our approach the agents can deal with lying witnesses and imperfect observation data by applying an algorithm that is based on probabilistic reasoning. This algorithm has been described and modelled with sociological and psychological terminology to improve correlations between simulation and real world application. It is applicable to *electronic commerce* as well as the problem of distinguishing between friendly and malicious hosts or migrating programs.

The preliminary results show that using the TrustNet is of advantage to the agents in a range of settings. In some experiments, agents using the TrustNet have performed after only 15 interactions in a population of 54 agents 10% better than agents from a control group. The results also

show that the use of our approach will be especially beneficial if the number of agents observed is small and / or the total number of agents is large.

6.2. Future Work

While this paper is being compiled, we are conducting experiments in the same settings with different configurations to collect more data. The upcoming results will help us to pin down under which circumstances using the TrustNet will increase performance and to what extent. We will extend our research to settings were the pay-off matrix for the game changes over time. We are also interested in researching the forming of coalitions, in changes of the behaviour over time and especially in more complex forms of lying that are guided by strategies.

Varying the ratio of egoists and altruists leads to different distribution curves on the agent society structure. Researching the effects of the TrustNet in such different society structures is another interesting challenge.

Acknowledgements

Michael Schillo would like to thank Jessica Seibert for her patience and support while this research has been carried out.

Bibliography

(Axelrod 1984) Axelrod, R. *The Evolution of Cooperation*. New York: Basic Books, 1984.

(Bazzan et al. 1997) Bazzan, A. L. C., Bordini, R. H. und Campbell, J. A. Agents with Moral Sentiments in an Iterated Prisoner's Dilemma Exercise. In: (Dautenhahn et al. 1997).

(Beth et al. 1994) Beth, T., Borcherding, M. und Klein, B. Valuation of Trust in Open Networks. In: *Computer Security-ESORICS 94 – Third European Symposium on research in Computer Security*. Brighton, UK, November, 1994.

(Dautenhahn et al. 1997) Dautenhahn, K., Masthoff, J. und Numaoka, C. *Socially Intelligent Agents*. Papers from the 1997 AAAI Fall Symposium, November 8-10, Cambridge, Massachusetts, Technical Report FS-97-02, 1997.

(Edelmann 1987) Edelmann, G. Neural Darwinism: The Theory of neural group selection. Basic Books, 1987.

(Luce and Raiffa 1957) Luce, R. and Raiffa, L. *Games and Decisions*. Wiley, New York, 1957.

(Malsch et al. 1996) Malsch, T. et al. *Expeditionen ins Grenzgebiet zwischen Soziologie und Künstlicher Intelligenz*. In: Künstliche Intelligenz, 2:96, S. 6-12, in german, 1996.

(Pearl 1988) Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Francisco, 1988.

(Sander and Tschudin 1998) Sander, T. and Tschudin, C. F. Fritz. *Protecting Mobile Agents against Malicious Hosts*. In: (Vigna 1998), S. 44 ff., 1998.

(Schillo and Funk 1998) Schillo, M. and Funk, P. Spontane Gruppenbildung in künstlichen Gesellschaften. In: *Proceedings des Workshops Sozionik der 22. Jahrestagung für Künstliche Intelligenz*. In German, 1998.

(Schillo 1999) Schillo, M. Vertrauen: Ein Mechanismus zur sicheren Koalitionsbildung in künstlichen Gesellschaften. Diploma Thesis, In German, To appear, 1999.

(Schillo et al. 1999) Schillo, M., Lind, J., Funk, P. Gerber, C. and Jung, C. *SIF - The Social Interaction Framework. System Description and User's Guide to a Multi-Agent System Testbed*. DFKI Research Report RR-99-02, Saarbrücken, Germany, 1999.

(Vigna 1998) Vigna, G. *Mobile Agents and Security*. Springer-Verlag, Berlin, 1998.