Sequential Decision Making with Untrustworthy Service Providers

W. T. Luke Teacy, Georgios Chalkiadakis, Alex Rogers and Nicholas R. Jennings
Electronics and Computer Science,University of Southampton
Southampton, SO17 1BJ, United Kingdom
{wtlt,gc2,acr,nrj}@ecs.soton.ac.uk

ABSTRACT

In this paper, we deal with the sequential decision making problem of agents operating in computational economies, where there is uncertainty regarding the trustworthiness of service providers populating the environment. Specifically, we propose a generic Bayesian trust model, and formulate the optimal Bayesian solution to the exploration-exploitation problem facing the agents when repeatedly interacting with others in such environments. We then present a computationally tractable Bayesian reinforcement learning algorithm to approximate that solution by taking into account the expected *value of perfect information* of an agent's actions. Our algorithm is shown to dramatically outperform all previous finalists of the international Agent Reputation and Trust (ART) competition, including the winner from both years the competition has been run.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Design, Measurement, Experimentation

Keywords

Trust, Reputation, Uncertainty, Reinforcement Learning

1. INTRODUCTION

Trust constitutes an important facet of multi-agent systems research since it provides a form of distributed social control within highly dynamic and open systems whereby agents form opinions about others based on their own past interactions, as well as from the reports of other agents [11]. Now, in many dynamic open systems, such as e-marketplaces, agents have to interact with one another to achieve their goals—for example by purchasing services or information from each other. Here, agents may be self-interested, and when trusted to perform an action for (or provide information to) another, may betray that trust by not performing the action as required. In addition, due to the size of such systems, agents will often have to interact with agents with which they have little or no past experience. There is thus a need for models of trust and reputation that will ensure good interactions among software agents in large scale open systems.

Cite as: Title, Author(s), *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

To this end, a number of trust models and strategies have been proposed in order to deal with distinct aspects of the interactions between agents (e.g. to deal with lying agents, and to model and learn the behaviour of other agents [17, 13]). However, none of these approaches has so far explicitly dealt with the *sequential decision making problem* arising naturally in computational economies. Specifically, agents that repeatedly interact with each other, by either purchasing services or information from a pool of agents, have the opportunity to form opinions over time regarding the trustworthiness of some of their prospective service or information providers; however, they always face the dilemma of whether to keep interacting with the same "trusted" agents (i.e., *exploit* their experience) or to keep experimenting by trying other agents with whom they haven't had much interaction so far (i.e., *explore* in order to discover better providers). This is the classic exploration-exploitation problem in a (multi-agent) computational trust setting. Now, a number of approaches have been proposed to tackle this problem in non-trust related reinforcement learning (RL) settings. Here we adopt a principled *Bayesian approach* to resolve this dilemma.

In more detail, in addition to describing the theoretical aspects of our model, we also evaluate its performance (using a *tractable* approximation algorithm) against other trust strategies in a simulated e-market environment. Specifically, we apply our approach to the Agent Reputation and Trust (ART) International Competition Testbed [10], and show that it outperforms (by a huge margin) all the finalists of the two competition years. In so doing, this paper advances the state of the art in the following ways: First, we propose the first trust model that enables rational agents to take optimal sequential decisions in environments necessitating exploration when dealing with potentially untrustworthy service providers. Second, we provide a computationally tractable algorithm to approximate the optimal (but intractable) Bayesian solution. This algorithm extends the *value of perfect information* exploration ideas of [8, 7, 4] into a computational trust setting for the first time. Thus, it trades off the expected gains from exploration against the expected costs of choosing potentially suboptimal providers. Third, we demonstrate that this is the most effective strategy yet devised for the ART benchmark scenario.

The paper is organized as follows: Sec. 2 provides a background to computational trust and Bayesian RL; in Sec. 3 we describe our generic Bayesian trust model, detail the optimal solution to the agents' sequential decision making problem, and present our approximation algorithm to this solution. Then, in Sec. 4, we describe the specifics of the ART competition, and instantiate and evaluate our approach in this testbed. Finally, Sec. 5 concludes.

2. BACKGROUND

Here, we briefly review related work on computational trust and Bayesian RL.

2.1 Computational trust

The issue of trust in multi-agent systems is one that is widely recognized, and which has been addressed in a number of different ways (see [11] for a full review). In particular, much of this work focuses on estimating the future behaviour of an agent's peers, so it can decide how best to interact with them in the future. A common theme is the need to make assessments based on a variety of information sources, so that predictions are not sensitive to the absence or failure of any particular source. This is especially true of large systems, in which interactions regularly occur between entities that have no previous experience of each other. In this case, agents must base their decisions on third party experience, or use other information, such as general environmental trends.

Typically, the proposed mechanisms vary in how they represent agent behaviour, the sources of information they use, and the learning mechanism adopted. For example, with regard to information sources, we can use knowledge of social norms and rules [12], or the relationships that may exist between agents [1]. However, such evidence cannot be expected to exist in all domains. In contrast, in many domains, agents can observe the past behaviour of their peers, and so most models adopt past experience (direct or third party) as an indicator of preformance, including those discussed below.

With regard to learning, early mechanisms tend to adopt a heuristic approach, with improvised functions to account for different aspects of agent behaviour (for example, see [15]). However, such techniques have few theoretical properties showing how they should perform in different conditions, or how they compare to any notion of optimal performance. Consequently, a number of recent trust models have adopted a theoretical grounding based on probability theory. For example, models in which probability distributions are estimated over possible agent behaviour are presented in [17] and [13]. Probabilistic systems, such as these, allow an agent to combine its own experience with third party information, in a manner that is principled, and accounts for the possibility that third party information may be malicious, or otherwise unreliable. In addition to their theoretical benefits, such systems have also been shown to outperform other approaches in practice. Indeed, the winner for the ART competition for the past two consecutive years has been based on probability theory [16].

2.2 Bayesian reinforcement learning

Now, turning our discussion to Bayesian RL, assume we have an agent learning to control a stochastic environment modeled as a Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, \Pr, R \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ represent finite state and action sets; $\Pr$ are transition dynamics refering to a family of transition distributions $\Pr(s, a, \cdot)$, specifying the $\Pr(s, a, s')$ probability with which state s' is reached when action a is taken at s; and $R : \mathcal{S} \mapsto \mathbf{R}$ is a (stochastic) reward function specifying the probability with which the agent obtains reward r when state s is reached. Now, an RL agent does not know R and/or $\Pr$: so, it must *learn* a policy to maximize *sequential* performance (i.e., maximize the expected sum of future discounted rewards over an infinite horizon) based on the observed results of its interactions with the environment.

In *model-based RL*, the learner maintains an *estimated* MDP $\langle \mathcal{S}, \mathcal{A}, \widehat{\Pr}, \widehat{R} \rangle$, using standard methods to solve (or approximate the solution of) this MDP at each stage of the RL process. *Single-agent* Bayesian RL [7] allows agents to incorporate priors and explore optimally, assuming a prior density P over possible dynamics D and reward distributions R, updated with each experience tuple $\langle s, a, t, r \rangle$ observed. In a similar manner, agents using *multiagent* Bayesian RL [4] update priors over the space of possible *opponent strategies*, as well as over the space of possible MDPs. With sequential performance in mind, one can identify two components of the value of an agent's action at any particular belief state: an expected value with respect to the current belief state; and an expected value of the action's impact on that belief state. This second component, in particular, captures the *expected value of information* (EVOI) of an action in the following sense: each action triggers some "response" by the environment, which changes the agent's beliefs, influencing subsequent action choice and expected reward. Now, EVOI need not be computed directly, but it can be incorporated in Bellman equations describing the solution to the POMDP representing the exploration-exploitation problem (by conversion to a belief-state MDP). The optimal course of action for the agents is then to act greedily w.r.t. their actions' values (i.e., *without* a need for *explicit* exploration); this *Bayesian exploration* outperforms in expectation any other method that uses the same prior knowledge [3]. Thus, the Bayesian approach provides the *optimal* solution to the agents' exploration-exploitation problem. Furthermore, experiments with various tractable Bayesian algorithms in [8, 7, 4] demonstrate the practical value of Bayesian exploration.

3. A GENERIC BAYESIAN TRUST MODEL

Having outlined the background, we now describe a generic Bayesian trust model that can be used in a reinforcement learning framework to help an agent take sequentially optimal decisions while repeatedly interacting with service providers under uncertainty. We then propose an RL algorithm to tackle the problem in a computationally tractable manner.

3.1 The problem and its optimal solution

A *Bayesian trust sequential decision making problem* is characterized by a set of enquiring agents (or "trusters")[1], a set of service (or information) provider agents (or "trustees"), a set of types for the agents, a set of enquiring actions, a set of outcomes, a reward function, and agents' beliefs over types. We now describe each of these components in detail:

Assume a set of trusters $M = \{1, \ldots, m\}$ and a set of service providers $N = \{1, \ldots, n\}$. Nothing in our model prevents a truster from being a service provider itself—i.e., M and N may intersect. Each provider j has a specific *type* τ_j, which intuitively captures its "trustworthiness".[2] This, in the general case, can be considered to be drawn from a continuous space T_j (for example, a space of potential standard deviations defining a provider's accuracy or trustworthiness—as is the case in the ART framework). For any collection of service providers $S \subseteq N$, $\boldsymbol{\tau}_S = \langle \tau_{i;i \in S} \rangle$ is the vector of types of agents in the collection. Each agent i knows its own type τ_i, but not those of other agents: thus, a truster is unaware of the trustworthiness of the providers.

A truster i has available to it a finite set of *enquiring actions* A_i of size $2^{|N|}$: Specifically, a truster may (or may not) choose to contact and request a service (or information) from up to $|N|$ of the providers available. When an action is taken, it results in some *outcome* $o \in \mathcal{O}$. The odds with which an outcome is realized depend on the types (trustworthiness) of the contacted trustees: if, for example, they are art experts and their assessment on the value of an artwork is requested, the outcome of enquiring from them is the collective assessment error (calculated given the observed individual assessment errors, as we detail in Section 4). Whenever a

[1] For simplicity, henceforth in our paper we will refer to any such truster simply as an "agent"—unless it is clear from the context that this term refers to a trustee.

[2] Recently, Bayesian RL was used in the coalition formation setting, where agents maintain beliefs regarding potential partners' capabilities (types) [5, 6]. Our model bears certain resemblances (but also very distinct differences) with that work.

truster contacts collection S of service providers (by taking enquiring action $a_S \in A_i$), and trustee types are as in $\boldsymbol{\tau}_S$, an outcome o occurs with some probability drawn from a density $P(o|\boldsymbol{\tau}_S)$. Each outcome-action pair $\langle o, a_S \rangle$ then results in some *reward* $R_i(a_S, o)$ for truster i.

At each point in time, a truster has a *belief state* over the possible types of its potential trustees, which is defined as follows. For each potential trustee j, truster i maintains a prior belief state over its types, denoted b_i^j. One could think of the b_i^js as a set of hyperparameters describing the distributions over types. Thus, each b_i^j defines a probability density $P(\tau_j|b_i^j)$. Assuming that trustees provide information independently, we can assume that the belief state b_i of agent i consists of a collection of the b_i^j priors. In this way, a *joint density* $P(\boldsymbol{\tau}_S|b_i)$ can be defined for any collection S of agents, with $P(\boldsymbol{\tau}_S|b_i) = \prod_{j \in S} P(\tau_j|b_i^j)$.

Now we describe an RL process in which agents repeatedly contact subsets of potential trustees for service. In this case, a truster has to decide which subset S of trustees to enquire from (that is, which enquiring action a_S to take); then, through observation of the outcome of the service (or the accuracy of the received information), the truster is able to update his beliefs regarding the trustworthiness of the providers.

The process proceeds in stages: at each time step t, at which the truster has current beliefs b_i, it takes an action a_S and observes the outcome o. Then, i is able to update its beliefs about the providers' types. Denoting the *posterior* belief state of agent i, following experience $\langle o, a_S \rangle$, as b_i^{o,a_S}, we get the posterior $P(\boldsymbol{\tau}_S|b_i^{o,a_S})$ by Bayes rule:

$$P(\boldsymbol{\tau}_S|b_i^{o,a_S}) \propto P(o|\boldsymbol{\tau}_S)P(\boldsymbol{\tau}_S|b_i)$$

The process then repeats.

Given the model above, we now provide a set of Bellman equations [2] that provide the solution of the corresponding belief-state MDP that describes the sequential decision making problem facing a truster i. In these equations, $Q_i(a_S, b_i)$ denotes the quality of performing the enquiring action a_S of asking a subset of S service providers for their services while at belief state b_i; and $V_i(b_i)$ denotes the value of being at belief-state b_i:

$$Q_i(a_S, b_i) = \int_{\boldsymbol{\tau}_S} P(\boldsymbol{\tau}_S|b_i) \int_o P(o|\boldsymbol{\tau}_S)[r_i + \gamma V_i(b_i^{o,a_S})] do \, d\tau_S \tag{1}$$

$$V_i(b_i) = \max_{a_S} Q_i(a_S, b_i) \tag{2}$$

where r_i is the immediate reward for i, provided by the reward function $R_i(a_S, o)$; and $\gamma \in [0, 1)$ is a discount factor.

Notice that this formulation takes into account both the immediate value to i of performing an enquiring action, and its long-term (sequential) value deriving from the fact that i's beliefs will change as a result of the information it will receive by performing a_S. Thus, the agents enquire ("explore") in an informed way, asking for opinions from the providers they consider more reliable, taking into account the *value of information* implicit in these equations: this means the agents take into account the potential effect that new information will have on their future decisions, and explore in such a way that their anticipated costs (captured in the reward function) are outweighed by their anticipated (long-term) benefits. This belief-state MDP formulation enables us to resolve the exploration-exploitation tradeoff in this setting—in an optimal way, provided one can solve this set of equations.

Unfortunately, solving these equations exactly is, in the general case, impossible—the intractability of the solution arising from the well-known curse of dimensionality [2]. This forces us to consider computational approximations to tackle the problem. We now describe one such approximation to the optimal solution provided above; our algorithm avoids time-consuming lookahead calculations, but rather focuses on (myopically) estimating the value of information of any action.

3.2 The VPI exploration algorithm

The *value of perfect information (VPI) exploration method* we present here is based on a technique initially developed in [8, 7] for single-agent RL, and which was adapted to the multiagent RL context as described in [4]. Recasting the relevant ideas to the trust and reputation setting, we now propose a *VPI* exploration method that estimates the (myopic) value of obtaining perfect information about the types of service providers given current beliefs. This leads to the agents calculating an estimate of the sequential value of any action of selecting a set of providers. Though the basic idea of our algorithm is as in [8, 7, 4], it differs in that there is no sampling over a space of MDP models involved, but rather we require sampling over a space of types. Furthermore, applying the generic *VPI* ideas in the ART setting requires dealing with subtle technical issues in a principled, but also practical, manner. This will become apparent in Sec. 4, where we discuss the construction of the reward function for our setting, and ways to achieve action space reduction.[3] In addition, we note that this is the first time the *VPI* ideas have been applied in a realistic setting encompassing dozens of agents.

Now, let us consider what can be gained by learning the true value of *some* action a_S of choosing a subset S of providers to interact with. If a_S is executed, assume that it leads to specific *exact evidence* regarding the types of the agents in S. Thus, we assume that the real type vector $\boldsymbol{\tau}_S^*$ is revealed after a_S. In this way, the *true* value of a_S is also revealed for i; let it be denoted as $q_{a_S}^* = Q_i(a_S|\boldsymbol{\tau}_S^*)$. We calculate this myopically as:

$$Q_i(a_S|\boldsymbol{\tau}_S^*) = \int_o P(o|\boldsymbol{\tau}_S^*) R_i(a_S, o) do \tag{3}$$

employing sampling from the $P(o|\boldsymbol{\tau}_S)$ distribution, for computational efficiency reasons.[4]

This information is of interest only if it leads i to change its decision as to what strategy to follow. Specifically, there are two ways to take advantage of this new, "perfect" knowledge.

First, suppose that under its current belief state b_i, the value of i's current best action $a_1 = a_{S_1}$ (e.g., asking the S_1 subset for service) is $\bar{q}_1 = \bar{Q}_i(a_{S_1}, b_i)$, the expected value given this belief state (obtained through averaging over samples from b_i). Now, supposing that the new knowledge indicates that a_S is a better action (i.e., $q_{a_S}^* > \bar{q}_1$), i should perform a_S instead of a_{S_1}, gaining $q_{a_S}^* - \bar{q}_1$.

Second, say that the value of the current second best action $a_2 = a_{S_2}$ (e.g., enquiring of S_2) is $\bar{q}_2$. If action a_S coincides with the action considered best, $a_1 = a_{S_1}$, and the new knowledge indicates that the real value $q_{a_{S_1}}^* = q_{a_S}^*$ is less than the value of the action previously considered as second-best (that is, if $q_{a_{S_1}}^* < \bar{q}_2$), then the agent should ask S_2 instead of S_1 for service, gaining $\bar{q}_2 - q_{a_{S_1}}^*$.

Thus, the *gain* from learning the true value $q_{a_S}^*$ of a_S is:

$$gain_{a_S}(q_{a_S}^*|\boldsymbol{\tau}_S^*) = \begin{cases} \bar{q}_2 - q_{a_S}^*, & \text{if } a_S = a_1 \text{ and } q_{a_S}^* < \bar{q}_2 \\ q_{a_S}^* - \bar{q}_1, & \text{if } a_S \neq a_1 \text{ and } q_{a_S}^* > \bar{q}_1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

[3] Also, in reality, the distribution over types is a distribution over the *precision* of the providers' opinions, not readily provided but carefully constructed given the ART testbed specification (see Sec. 4).

[4] We note that $q_{a_S}^*$ could also be calculated by employing any n-step lookahead method—if one is willing to pay the additional computational price imposed.

However, i does not know in advance which types will be revealed for S; thus, he needs to take into account the *expected* gain given prior beliefs. Hence, i computes the *expected value of perfect information (EVPI)* about performing a_S as:

$$EVPI(a_S|b_i) = \int_{\boldsymbol{\tau}_S} gain_{a_S}(q^*_{a_S}|\boldsymbol{\tau}_S)P(\boldsymbol{\tau}_S|b_i)d\boldsymbol{\tau}_S \quad (5)$$

This expected value of perfect information represents the expected gain deriving from the exploration of enquiring action a_S. Thus, the value of a_S after taking *EVPI* into account is now defined as:

$$QV_i(a_S|b_i) = \bar{Q}_i(a_S, b_i) + EVPI(a_S|b_i) \quad (6)$$

The agents should then use these *QV* values, instead of using the usual Q-value quantities to select providers (in other words, *EVPI* is used as a way to boost the desirability of actions). Since our type space is continuous, we sample the joint type distribution to calculate the expected values and *EVPI* above.[5] In summary, *VPI exploration* proceeds as follows:

1. A number of type configurations are sampled from density P given current beliefs b_i.
2. The "true" q^*-values of any potential enquiring action a_S, w.r.t. each sample, are calculated using Eq. 3, an average $\bar{q}_{a_S}$ value is calculated for each a_S (given all samples drawn from b_i), and actions a_S are ranked given the $\bar{q}_{a_S}$ values.
3. The gain from choosing each a_S is then calculated via Eq. 4.
4. The *EVPI* for a_S is calculated via Eq. 5, the QV_i values for (any) a_S are calculated via Eq. 6.

The *QV*-values calculated by the *VPI* algorithm are subsequently used for action selection. Put simply, agent i should perform an action with maximal QV_i value (if there are more than one such actions, i randomly chooses one among them).

Finally, so far we have experimented with our method calculating the Q-values above myopically. This is done for computational efficiency reasons. Myopic as this calculation may be, our experiments clearly show the benefits of using this particular *VPI* algorithm. However, we note that our method is generic, in the sense that it can be used for the calculation of *VPI* over action quality values generated using any method deemed appropriate.

4. APPLICATION TO ART

We now present an empirical evaluation of the techniques described above by applying them to the ART testbed.[6] The aim of this testbed is to provide a standard problem scenario that can be used to compare different approaches to modelling and applying trust and reputation in multi-agent systems. Specifically, the testbed simulates a marketplace consisting of service providers (agents) that compete to provide information services. There is a fixed total number of clients who are apportioned between the agents according to the comparative quality of their service provision. Each of the providers needs to spend money in order to gain information. Furthermore, they can improve their quality of service by requesting (against a payment) information from their competitors. However, it is not necessary that the requested agents will provide good information. In fact, as a result of the competition between the agents, it is quite likely that they will provide bad information. Thus, within this competition, agents must not only decide how much of their income to spend to maximise their profit, but also which of their competitors they should purchase opinions from to obtain the most accurate information for the lowest investment.

For our purposes, the ART testbed provides a suitable scenario on which to evaluate our proposed methods for three reasons. First, there is a trade-off between the cost to an agent for acquiring information from its competitors, and the potential increase in the agent's future marketshare that such acquisition may bring about. Second, to determine which of its competitors provide the best information for a given amount of money, an agent must purchase information from different providers and compare their relative accuracy. Thus, each agent must make non-trivial decisions about how to trade-off immediate costs against potential future rewards. Finally, as the ART testbed provides a shared platform on which to evaluate agent decision mechanisms, its use facilitates comparisons between our methods and previous approaches in the literature to the same problem domain. In particular, we shall compare against previous work that provided the winning algorithm of the annual ART competition in the past two years [16]. The rest of this section is structured as follows: Sec. 4.1 describes the details of the testbed necessary to understand the application of our approach to the scenario; Sec. 4.2 describes the instantiation of our model to the testbed; and Sec. 4.3 presents our empirical results.

4.1 The ART testbed scenario

The main goal of an agent competing in the ART testbed is to maximise the profit it receives by the end of each game. To do so, each agent is assigned a number of artworks (paintings) for which it estimates the monetary value, in return for a fixed payment. Each painting is associated with exactly one client (one painting per client) drawn from a fixed set of clients, C, and is assigned to exactly one agent. Moreover, each game consists of a number of timesteps, and at the end of each timestep, each agent is assigned a proportion of clients according to the accuracy of its previous estimates, relative to its competitors. In this way, the agents that provide the most accurate estimates gain the largest share of available client revenue. However, in the interest of fair play, each agent initially receives an equal proportion of assignments, and to discourage end-game strategies, agents are not aware of the total number of timesteps in each game.

To make a profit, agents have three revenue streams: (1) payments received for appraisals of client paintings, (2) payments received from competitors for help in evaluating their painting assignments, and (3) payments received from competitors to help assess the reliability of opinions provided by other competitors. Each of these types of transaction have a fixed price. Specifically, an agent receives a fee of c_a for each painting assigned from a client; c_p for each painting evaluation requested by a competitor; and c_r for selling information about its competitors to other competitors. Each of these payment values are constant, and are chosen such that $c_a > c_p > c_r$.

Of these revenue streams, we shall focus our attention on payments received for painting appraisals, c_a, as these generally make up the majority of an agent's income. In particular, we shall consider how an agent can invest in both its own opinion and that of its competitors, so that it increases its marketshare. At the end of each timestep, each agent, i, is assigned a new marketshare, m_i, according to Eqs. 7 to 9.[7]

$$m_i = q \cdot m'_i + (1 - q) \cdot \tilde{m}_i \quad (7)$$

[5] Of course, the calculation of Q-values and *EVPI* can be done in a straightforward manner (using summation instead of integration, and no sampling) if the type space is discrete and the number of possible type configurations small.

[6] The ART testbed website can be found at `http://www.lips.utexas.edu/art-testbed/`.

[7] Equation 9 differs from its definition in [10], but matches the one actually in use in the ART competition.

$$\tilde{m}_i = \frac{1/e_i}{\sum_{b \in \mathcal{A}} 1/e_b} \tag{8}$$

$$e_i = \frac{1}{|C_i|} \sum_{c \in C_i} \frac{|p^*_{i,c} - v_c|}{v_c} \tag{9}$$

Here, m'_i, is the agent's previous marketshare; $q \in [0, 1)$ is a parameter determining the influence of previous marketshares over new assignments; $\tilde{m}_i$ is the agent's *provisional* marketshare, prior to adding the effect of the previous markshare; and e_i is the agent's average relative error for its client painting assignments in the most recent timestep. The latter term is calculated according to Eq. 9, in which C_i is the set of client paintings assigned to i in the previous timestep, v_c is the true value of painting c, and $p^*_{i,c}$ is the agent's overall estimate ("appraisal") of its value.

Each appraisal, $p^*_{i,c}$, is generated in the following way. First, the agent generates its own personal estimate (known as an opinion) of the painting's value. For agent j, its opinion of painting c is denoted $p_{j,c}$, and is generated by the testbed according to a normal distribution with mean v_c and standard deviation given by Eq. 10.

$$s = \left(s^* + \frac{\alpha}{c_g}\right) v_c \tag{10}$$

Here, s^* is a value randomly selected by the testbed from $\{0.1, 0.2, \ldots, 1.0\}$ that represents an agent's expertise in estimating a painting. Each painting is classified as belonging to one of a finite number of eras, and each agent has a different s^* value assigned for each era. The standard deviation is further determined by a constant parameter α known by all agents, and c_g, which is the monetary amount the agent chooses to spend on generating its opinion.

Clearly, an agent can increase the expected accuracy of its estimates (and thus its marketshare) by increasing its value of c_g. However, it must be careful to balance its investment against its need to make a profit, and s^* places a hard limit on how much it can increase its accuracy through personal assessment. To further increase its accuracy, an agent can purchase opinions from its competitors at a fixed price of c_p, with the constraint that only one opinion can be purchased from each competitor per painting. These third party opinions are generated in the same manner as an agent's personal opinion, but a competitor does not reveal its values for s^* or its policy for choosing c_g. Finally, an agent i's appraisal of a painting is calculated as a weighted sum of all purchased opinions, $p_{j,c}$, which can include the agent's own opinion (Eq. 11). The weights, w_j, assigned to each opinion, $p_{j,c}$, are set by the agent according to its beliefs about the s^* and c_g values used to generate the opinions. Optimal weights can be determined for an agent's own opinions given perfect knowledge of s^* and c_g, and can be estimated using Bayesian analysis for the unknown behaviour of its competitors [16].

$$p^*_{i,c} = \frac{\sum_j w_j \cdot p_{j,c}}{\sum_j w_j} \tag{11}$$

4.2 Model instantiation for the ART scenario

Based on the details of the ART testbed outlined in the previous section, we have devised a strategy for the scenario that uses the concepts described in Sec. 3. In particular, we concentrate on specifying how an agent decides the number of third party opinions to purchase to assess each painting, which competitors it should purchase opinions from, and how its beliefs about its competitors' opinion accuracy should be represented and updated.

In addition to this, a complete strategy for the ART testbed must specify policies for deciding how much to spend on personal opinions, deciding if and when to purchase reputation information, and how to respond to requests from competitors for reputation and opinions. For these aspects, we rely on the policies defined by the competition-winning strategy of [16]. The salient features are that an agent will always spend \$4 on generating its own opinion ($c_p = 4$), regardless of whether that opinion is for a directly assigned painting or requested by a competitor (we refer to [16] for the rationale for this); and due to the difficulty in its interpretation (as defined in the ART scenario), reputation is not used.

This allows us to concentrate on the problem of opinion selection for directly assigned client paintings, which we address by specifying three components of the generic model presented in Sec. 3. Specifically, we define (1) the outcome space, $\mathcal{O}$, for performing an enquiring action $a_S \in A_i$; (2) the representation of an agent's beliefs, b_i; and (3) the reward function, $R_i(a_S, o)$, which defines the immediate reward to agent i for performing a_S with outcome o. Together, these components fully instantiate the generic model, and allow an agent to use VPI for practical opinion provider selection.

To begin, we first specify the action outcome space, by defining the outcome of performing action a_S for painting c as a tuple $o = \langle v_c, \mathbf{p}_c \rangle$, where v_c is the true value of c and $\mathbf{p}_c =< p_{1,c}, \ldots, p_{l,c} >$ is the vector of estimates of v_c, generated by each of the opinion providers from which agent i requested an opinion. With this in mind, we now define the reward function and belief representation, and also discuss how the large number of actions ($2^{|N|}$ for the set of possible providers N) can be reduced for practical reasoning.

The Reward Function

In the interest of computational tractability, we define $R_i(a_S, o)$ under three simplifying assumptions. Specifically, that (1) each game lasts forever; (2) in each timestep, all paintings assigned to an agent are of the same era; and (3) the agent must consult the same collection of competitors for all paintings assigned in the current timestep. From the definition of the ART testbed, the immediate reward is then[8]:

$$R_i(a_S, o) = \tilde{m}_i \cdot c_a - m_i \cdot (|S| \cdot c_p + c_g) \tag{12}$$

where m_i is i's current marketshare and $\tilde{m}_i$ is i's provisional marketshare based on its current action. An interesting property of this equation is that the cost of generating an appraisal depends on the agent's current marketshare, while the increase in revenue (as a result of the current action) depends on the agent's provisional marketshare. The overall effect is that an agent will behave most competitively when its current marketshare is low and it believes it can significantly improve this by investing in its appraisal. For example, if the agent already controls most of the available marketshare then a significant increase in marketshare is impossible, and so a large investment in generating an appraisal may not be rational. However, if its marketshare is currently very low, then it only has to appraise a small number of paintings, and so even if it invests a lot to appraise them, the benefit in terms of increased marketshare may be significantly higher.

The influence of action outcome o on the reward emerges from its effect on the provisional marketshare. Specifically, from Eq. 8:

$$\tilde{m}_i = \frac{1/e_i}{1/e_i + \sum_{b \neq i} 1/e_b} = \frac{1/e_i}{1/e_i + \hat{e}} = \frac{1}{1 + e_i \cdot \hat{e}} \tag{13}$$

where $\hat{e}$ depends on the performance of i's competitors and e_i is i's average relative appraisal error for the specific timestep (calculated after observation of o using Eqs. 9 and 11). In general, e_i depends,

[8] From the scenario in [10], it is straightforward to prove Eq. 12, using mathematical induction over all timesteps: Eq. 12 actually incorporates the complete portion of the agent's future reward attributed to the specific provisional reward resulting from the current a_S (irrespective of the change in beliefs resulting from this action).

not only on the outcome for the current painting, but also on i's appraisal error for all other paintings it must assess in the current timestep. This means that, for this application, the rewards obtained for different paintings are dependent. However, as our experiments demonstrate, good performance can still be achieved in practice, even when assuming the independence of these rewards.

Modelling an Agent's Beliefs

For an agent to determine which actions it should perform, it must maintain beliefs about the behaviour of its competitors, which it updates in response to the observed outcomes of its actions. In particular, two aspects of competitor behaviour must be modelled: (a) the combined appraisal performance of all competitors, $\hat{e}$; and (b) the error distribution of each competitor's opinion estimates.

In general, $\hat{e}$, depends on the strategies employed by the agent's competitors, which may or may not adapt to the agent's own previous actions. Thus, in the absence of any specific model of behaviour, we assume only that the expected market performance changes slowly over time and estimate it using a moving average. That is, at the end of each timestep, an agent observes o and calculates its own relative appraisal error, along with its updated marketshare, and from this infers the value of $\hat{e}$ (Eqs. 14 and 15). The expected value of $\hat{e}$ in the next timestep is then estimated according to Eq. 16, where $\hat{e}'$ is the previous estimate, $h \in [0, 1)$ is the weight placed on the previous estimate, and $\hat{e}$ its most recent value. Specifically, from Eq. 13,

$$\hat{e} = \frac{1/\tilde{m}_i - 1}{e_i} \tag{14}$$

where, from Eq. 7, $\tilde{m}_i$ is given as:

$$\tilde{m}_i = \frac{m_i - q \cdot m_i'}{1 - q} \tag{15}$$

Then,

$$E[\hat{e}] \approx h \cdot \hat{e}' + (1 - h) \cdot \hat{e} \tag{16}$$

Given an adequate number of timesteps and stable market performance, Eq. 16 can provide a reasonable estimate of competitor behaviour, suitable for our purposes. In contrast, with regard to reliability of competitor opinions, a more detailed model of an agent's belief is key to choosing which competitors to acquire opinions from. This is due, not only to an agent's need to select competitors it believes will provide good opinions, but also to its need to model the uncertainty in its beliefs, so that it can explore the behaviour of competitors it knows little about.

For this reason, we adopt the following Bayesian model of an agent's beliefs about its competitor's opinions. From the scenario, we know that an agent's opinion is generated by the testbed according to a normal distribution with mean v_c and a standard deviation according to Eq. 10. As advocated in [16], with no loss of generality, agent j models the distribution of competitor i's opinion with less complexity by applying the following transformation:

$$\rho_i = \frac{(p_i - v_c)}{v_c} \tag{17}$$

The resulting variable, ρ_i, again follows a normal distribution, but is independent of the true painting value, with mean 0 and standard deviation $s = (s^* + \alpha/c_g)$, and so depends only on the opinion provider's expertise (s^*) for the painting era and its strategy for choosing c_g. Given these dependencies, an agent maintains separate beliefs about the distribution of ρ_i for each era and each competitor as follows.

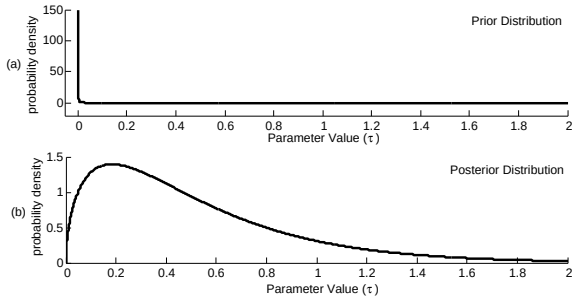

Figure 1: The τ_i Parameter Distribution Before (a) and After (b) Observing $\rho_i = 1, 2, 1$

For each competitor–era pair, an agent models the distribution of the precision:

$$\tau_i = 1/E[\rho_i^2] \tag{18}$$

which is defined as the reciprocal of the variance. This distribution over τ_i represents agent i's beliefs regarding the *type* ("trustworthiness" or "perceived expertise") of agent i in the specific era. If known, τ_i, fully determines the distribution of ρ_i for a given agent and era, and represents a state of complete information. In practice, an agent can only determine ρ_i for its own opinions, and not for those of its competitors. However, by modelling the distribution of τ_i, an agent can represent its beliefs about its competitors' opinions, given the evidence it currently has available. For example, if τ_i has a uniform distribution over all possible values of τ_i then this represents a state of no information, in which any value of τ_i is considered equally plausible. In contrast, if its distribution peaks sharply around one possible value then this implies a strong belief that the true value of τ_i lies close to that value.

As is standard practice, we initially assign a conjugate prior distribution to τ_i, so that beliefs can be easily represented and updated. Given that τ_i represents the precision of a normal distribution, this gives it a gamma distribution, with probability density function (p.d.f.) [9]:

$$P(\tau_i) = \frac{\beta^k}{\Gamma(k)} \tau_i^{k-1} \exp\left[-\beta\tau_i\right] \tag{19}$$

where $\beta > 0$ and $k > 0$ are *hyperparameters* specifying the shape of the distribution. Prior to observing any competitor behaviour, β and k are chosen to reflect the belief that it is equally plausible that a competitor's opinions may have high (good) or low (bad) precision relative to the agent's personal opinions, but that the true value is otherwise uncertain. Specifically, we choose $\beta = 10^{-6}$ and $k \approx 0.0591$, resulting in a 0.5 probability that ρ_i has a standard deviation greater than 0.4602, which is approximately the value one would expect in the ART competition from a competitor that is assigned better than average expertise and spends between 10% and 100% of its payment on generating its opinion. The resulting p.d.f. is illustrated in Figure 1 (part a), which encourages agents to explore competitor opinions given the high probability of good precision, while at the same time placing a low prior weight on such opinions due to the equally high probability of low precision.

When an agent observes the outcome of acquiring opinions from a collection of providers, it calculates the corresponding values of ρ_i by transforming the opinions using Eq. 17 and the observed true painting values. These are then used to update the hyperparameters according to Eq. 20, which provide the correct posterior distribution consistent with Bayes rule [9]. Figure 1 part (b) shows the posterior distribution formed after applying these equations recur-

sively for three observed values of ρ_i, $1, 2, 1$. The rapid change in shape after just a few observations demonstrates how an agent quickly updates its beliefs in light of gathered evidence.

$$k = k + \frac{1}{2}, \quad \beta = \beta + \frac{\rho_i^2}{2} \tag{20}$$

Action Space Reduction

When an agent decides which of its competitors to ask, it effectively chooses a subset of all its competitors. As there are 2^n subsets of any set of size n, an exhaustive search of all possible actions quickly becomes infeasible as the number of competitors increases. Thus, some method of reducing the search space is required, and so we currently adopt an algorithm consisting of the following three steps. First, in a game containing n competitors, we consider the set of n actions in which only one competitor is queried. These actions are then sorted in ascending order according to their expected utility and *EVPI*, so that the most desirable opinions are placed at the beginning of the list. Second, a new set of n actions is generated, by merging the previous actions in this order. Thus, in this new set, the kth action is to ask the k competitors considered best when enquired on their own. Finally, the new actions are assessed according to the *VPI* algorithm, and the best action is selected. As the reward for polling a collection of competitors is sub-additive, there is no guarantee that the selected action is optimal. However, as we shall demonstrate in the next section, this can achieve good performance in practice.

4.3 Empirical evaluation

In this section, we empirically evaluate the performance of our proposed strategy when run against previous competitors in the ART competition. In particular, we focus on three aspects of behaviour: (a) how *VPI* performs over time during each game, (b) how it performs in populations of agents that require varying amounts of exploration, and (c) how long our implementation takes to run in practice. In each of these experiments, we set the game parameters according to standard competition rules: each game ran for 100 time steps; paintings belonged to one of 10 eras; there were 20 clients in the system per competing agent; and the parameter values were $\alpha = 0.5$, $q = 0.1$, $c_a = 100$, $c_p = 10$, and $c_r = 1$. In all cases, each experiment was run for at least 30 runs for statistical significance, computed using t tests with 95% confidence intervals. With this in mind, we describe each of our three sets of results below.

Performance over time

To investigate *VPI*'s ability to explore over time, we ran a number of games involving populations of agents in which exploration is important to gain a competitive advantage. To achieve this, each game included three groups of "dummy" agents whose role was not to compete in the game, but to provide a source of opinions to the competitors. In each of these groups (denoted $G_{bad}, G_{med}, G_{good}$), "dummies" spent a consistent amount on opinion generation such that groups $G_{bad}, G_{med}, G_{good}$ provided a standard deviation of 5, 0.5 and 0.05 respectively.

To encourage exploration, G_{med} made up the largest proportion of the population at 20, G_{bad} consisted of 15 dummies, and G_{good} of only 5. The rationale for this is that the "good" dummies provide a strong competitive advantage, but the competitors must explore well to find them. The average bank balances for these games are illustrated in Figure 2, in which the competing strategies (including all finalists of the 2006 and 2007 competitions) appear in the legend in order of final score, and with 95% confidence intervals to indicate statistical significance. Specifically, *VPI* played against all

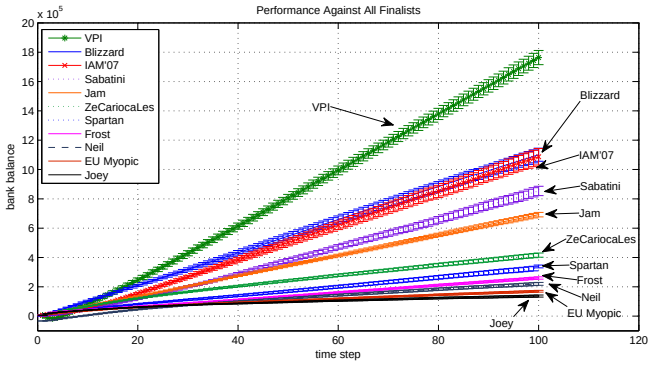

Figure 2: *VPI* versus ART Competition Finalists.

previous finalists, and a myopic expected utility-calculating algorithm (labeled *EU myopic*) that used the same reward function and model of agent behaviour, but did not account for the value of information. The results show that *VPI* significantly outperforms (by a factor larger than 1.6) the other strategies in this setting, including the previous winner of the 2006 and 2007 competitions (*IAM'07*)[9] and *EU myopic*. The poor performance of the latter shows the importance of exploration: despite having a good model of the environment, it cannot behave competitively without considering the impact of its actions on its changing beliefs. More generally, all agents quickly acquire a linear increase in bank balance over time, indicating a constant strategy for provider selection after an initial exploration of the environment. Interestingly, *Blizzard*, who came 3rd in the 2007 competition, appears to perform better exploration than *IAM'07*, on which it gains an early lead, but fails to maintain a significant advantage by the end of the game.[10] Despite this, no other agent comes close to *VPI*'s performance in this setting.

Performance in different populations

To assess the impact of varying populations on *VPI*'s performance, we ran more detailed experiments in which we varied the relative proportions of dummies providing high, medium and low standard deviations. Specifically, we kept the size of G_{bad} constant at 10, and the total size of G_{med} and G_{good} constant at 40, but varied the proportion of G_{good} relative to G_{med}. To reduce simulation time, we focused our attention on competitions against the previous winner, *IAM'07*, with all other finalists removed.

For these experiments, the average end of game bank balances are shown in Figure 3, plotted against the ratio of G_{med} to G_{good}. This shows that *VPI* remains competitive with *IAM'07* when there are equal numbers of G_{med} dummies and G_{good}. This is impressive given that *VPI* implements a generic model, while *IAM'07* uses specific prior distributions and heuristics tailored to the competition (see [16]). Moreover, as the proportion of G_{good} dummies decreases, exploration becomes more important, giving *VPI* a clear advantage: this highlights the strength of VPI exploration.

Runtime complexity

Although our results show that *VPI* vastly outperforms alternative strategies in terms of choosing better actions, it is important to consider the cost of such performance in terms of runtime complexity. Starting from our *myopic VPI* algorithm definition, it is easy to ver-

[9] *IAM'07*'s strategy was in fact identical to the strategy used by its 2006 "predecessor", *IAM* [16].

[10] We cannot comment further on *Blizzard*'s exploration behaviour, since we are unaware of the details of this competitor's strategy.

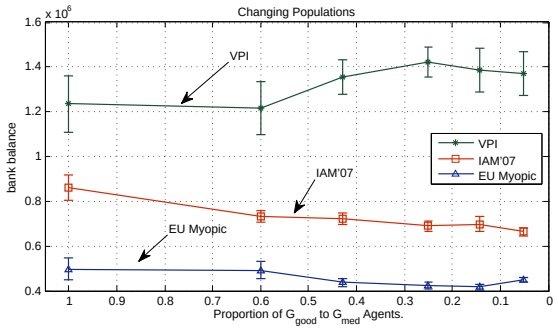

Figure 3: *VPI* versus *IAM'07* in Different Populations

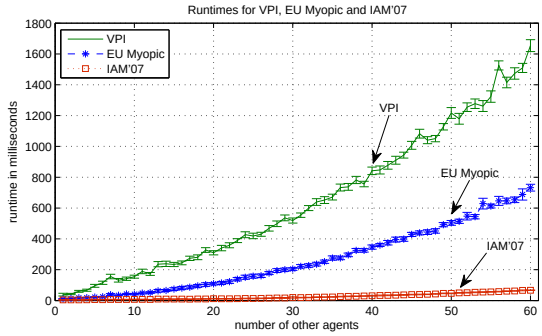

Figure 4: Runtime of *VPI*, *EU Myopic* and *IAM'07*.

ify that the algorithm has a theoretical runtime complexity linear to the number of opponents and the number of type samples used.[11]

To evaluate the VPI runtime performance in practice, we measure the total processing time for *VPI* to run per timestep (including action selection and belief updates) in games played against *IAM'07* and *EU Myopic*. Given that *VPI*, *EU Myopic* and *IAM'07* all maintain separate belief models for each agent and each era, and choose one set of opinions in each timestep for each era, these games present agents with paintings from only one era, to get a true indication of performance.

The average runtimes per timestep for these games are illustrated in Figure 4, plotted against the number of opponents, all of which implemented "dummy" strategies (i.e., G_{bad}, G_{med} and G_{good}). To obtain these results, simulations were run in Java, operating on cluster nodes with 2GB RAM, and dual AMD Opteron processors, and a type sample size of 50. The results show that all three algorithms operate in close to linear time, and although *VPI* has a larger runtime overhead, it is still reasonable, and, moreover, it scales well as the number of agents it must consider increases. This is significant because, although *VPI* is based on and approximates an optimal Bayesian formulation (which is otherwise intractable), it can be seen to produce good results in a practical time frame.

5. CONCLUSIONS

We presented a Bayesian approach for sequential decision making in multiagent environments requiring computational trust and reputation modeling. The Bayesian approach allows the agents to incorporate different trust priors and explore optimally with respect to their beliefs when choosing potential service or information providers (trustees) in such environments. We provided an algorithm that approximates the optimal Bayesian solution by taking into account the myopic value of perfect information entailed in an agent's actions, and demonstrated that it dramatically outperforms the (two years in a row) winning algorithm of the Agents Reputation and Trust international competition.

We believe that the value of this work is particularly apparent in e-marketplaces, where rational agents need to take trust-based decisions, without disregarding the impact of those decisions on their future welfare (as all previous existing work does). Building on this, for the future we want to integrate our *VPI* algorithm with algorithms performing explicit lookahead in belief space (to quantify expected gains in performance and costs in running time). Further, we would like to test our approach in even more dynamic environments, experimenting with opponents that change their behaviour over time. Both of the aforementioned tasks are readily allowed by our model. Last but not least, we are keen to recast and test these ideas in sensor networks [14], where the techniques described here may be used to learn and exploit correlations between sensors (due, for example, to their close physical proximity) in order to minimize redundant sampling and, thus, prolong sensor lifetime.

[11] By comparison, any lookahead algorithm would have a runtime that is exponential in the number of lookahead steps (proportional to s^n where n denotes lookahead steps and s is the sample size). In practice this means that even a 1-step lookahead method operating against 60 competitors would require roughly 3 times the allowed competition time to run (with $s = 50$).

Acknowledgments

This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Networks) project and the Data Information Fusion Defence Technology Centre (DIF DTC) Phase II Adaptive Energy-Aware Sensor Networks project. ALADDIN is jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1), and the DIF DTC project is joint funded by MoD and General Dynamics UK.

6. REFERENCES

[1] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In *Proc. of AAMAS'05*, 2005.

[2] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[3] R. Bellman. *Adaptive Control Processes: A guided tour*. Princeton University Press, 1961.

[4] G. Chalkiadakis and C. Boutilier. Coordination in Multiagent Reinforcement Learning: A Bayesian Approach. In *Proc. of AAMAS'03*, 2003.

[5] G. Chalkiadakis and C. Boutilier. Bayesian Reinforcement Learning for Coalition Formation Under Uncertainty. In *Proc. of AAMAS'04*, 2004.

[6] G. Chalkiadakis and C. Boutilier. Sequential decision making in repeated coalition formation under uncertainty. In *Proc. of AAMAS'08*, 2008.

[7] R. Dearden, N. Friedman, and D. Andre. Model based Bayesian Exploration. In *Proc. of UAI'99*, 1999.

[8] R. Dearden, N. Friedman, and S. Russell. Bayesian Q-Learning. In *Proc. of AAAI-98*, 1998.

[9] M. DeGroot and M. Schervish. *Probability & Statistics*. 2002.

[10] K. Fullam, T. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. Barber, J. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In *Proc. of AAMAS-05*. ACM Press, 2005.

[11] S. Ramchurn, T. D. Huynh, and N. R. Jennings. Trust in multiagent systems. *The Knowledge Engineering Review*, 19(1):1–25, March 2004.

[12] S. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. A computational trust model for multi-agent interactions based on confidence and reputation. In *Proceedings of the 6th International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69–75, Melbourne, Australia, July 2003. ACM Press.

[13] K. Regan, P. Poupart, and R. Cohen. Bayesian Reputation Modeling in E-Marketplaces Sensitive to Subjectivity, Deception and Change. In *Proc. of AAAI-06*, 2006.

[14] A. Rogers, E. David, and N. R. Jennings. Self-Organized Routing For Wireless Micro-Sensor Networks. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, 35(3):349–359, 2005.

[15] J. Sabater and C. Sierra. Social regret,a reputation model based on social relations. *SIGecom Exchanges*, 3(1):44–56, 2002.

[16] W. T. L. Teacy, T. D. Huynh, R. K. Dash, N. R. Jennings, M. Luck, and J. Patel. The art of iam: The winning strategy for the 2006 competition. In *Proc. of the 10th International Workshop on Trust in Agent Societies*, 2007.

[17] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *JAAMAS*, 12(2):183–198, 2006.