Trust-Based Mechanism Design *

Rajdeep K. Dash Sarvapali D. Ramchurn Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton, Southampton, UK.
{rkd02r,sdr01r,nrj}@ecs.soton.ac.uk

Abstract

*We define **trust-based mechanism design** as an augmentation of traditional mechanism design in which agents take into account the degree of trust that they have in their counterparts when determining their allocations. To this end, we develop an efficient, individually rational, and incentive compatible mechanism based on trust. This mechanism is embedded in a task allocation scenario in which the trust in an agent is derived from the reported performance success of that agent by all the other agents in the system. We also empirically study the evolution of our mechanism when iterated and show that, in the long run, it* always *chooses the most successful and cheapest agents to fulfill an allocation and chooses better allocations than other comparable models when faced with biased reporting.*

1. Introduction

Mechanism design (MD) is the field of microeconomics that studies how to devise systems such that the interactions between strategic, autonomous and rational agents lead to outcomes that have socially-desirable global properties. Given that the designer of a multi-agent system (MAS) typically has many of the same aims, there is a growing body of work that seeks to exploit the tools and concepts of MD to this end [3]. However, an important facet of MAS that is rarely considered in MD is that agents do not always complete their tasks as planned or promised (this means they are not always *sucessful*). Thus, for example, an agent may not always complete every task it starts or it may default on payment for a good. Furthermore, in traditional MD an agent chooses to interact with partners based on their costs or valuations only. However, cheapest is not always best and these agents may ultimately not be the most successful. Thus, in many practical situations the choice of interaction partners is motivated by an agent's individual model of its counterparts, as well as by information gathered from its environment about them. For example, on eBay buyers determine the credibility of particular sellers by considering their own interaction experiences with them (if they have any) and by referring to the historic evaluation information provided by other buyers. To capture this phenomenon, we exploit the notion of *trust* to represent an agent's perception of another agent's *probability of success* (POS) in completing a task [1] (as opposed to the agent's own belief about its own POS [9]). This, in turn, leads us to propose the area of *trust-based mechanism design* (TBMD) as an extension of traditional MD that adds trust as an additional factor to costs and valuations in decision making.

In more detail, the trust in an agent is generally defined as the expectation that it will fulfill what it agrees to do, given its observable actions and information gathered from other agents about it [1]. By their very nature, different agents are likely to hold different opinions about the trust of a particular agent depending on their experiences and the specifics of the trust model they use [10]. As a result, we cannot simply extend the conventional MD solution (e.g. the Vickrey-Clarke-Groves (VCG) mechanism) to encompass the notion of trust because such work is predicated on the fact that agents have *private and independent* information which determines their choice over outcomes. Trust, on the other hand, implies *public and interdependent information*.

In this work, we specifically consider MD in the context of task allocation (where it has often been applied [11]). In our scenario, agents may have different probabilities of success in completing a task assigned to them (e.g. it may be believed that a particular builder has a 95% chance of making a roof in five days, while another builder may be believed to have a 75% chance of doing so). Moreover, an agent may assign different weights to the reports of other agents depending on the similarity of their types. For example, consider a "repair engine" task assigned to a garage. In this case, two agents owning a Ferrari would assign higher weights to each other's report about the POS of the garage than they would to the report of another agent which owns a Robin Reliant.

Against this background, this paper develops and eval-

* The first author gratefully acknowledges funding from a BAE studentship and ORS scholarship. The authors thank Dr. R.Mason for helpful initial comments.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

uates the notion of trust-based mechanism design. We also define the general properties that trust models must exhibit to allow a trust-based mechanism to generate an optimal allocation of tasks. In particular, we advance the state of the art in the following ways:

1. We specify the properties that trust models must satisfy to be incorporated in mechanisms that permit efficient allocations.
2. We generalise the standard VCG mechanism to incorporate the notion of trust.
3. We prove that the trust-based mechanism we develop is efficient, individually rational, and incentive compatible.[1]
4. We empirically show that our trust-based mechanism leads to the most successful and cheapest agents being selected to fulfill an allocation in the long run and that it performs better than comparable mechanisms when agents' reports of POS are biased.

The remainder of the paper is structured as follows. Section 2 discusses related work in the areas of trust and mechanism design. Section 3 describes the basic task allocation problem with a standard VCG solution. Section 4 presents our trust-based mechanism design and develops an appropriate mechanism for trust-based task allocation. In section 5 we show the generality of our mechanism by reducing it to various known instances. Section 6 empirically evaluates the mechanism with respect to other comparable mechanisms. Finally, section 7 concludes and outlines future work.

2. Related Work

In associating trust to mechanism design, we build upon work in both areas. In the area of trust and reputation, a number of computational models have been developed (see [10] for a review). While these models can help in choosing the most successful agents, they are not shown to generate efficient outcomes in any given mechanism. An exception to this is the work on reputation mechanisms [4, 6]. However, these mechanisms only produce efficient outcomes in very constrained scenarios and under strict assumptions (e.g. in [4] sellers are monopolists and each buyer interacts at most once with a seller and in [6] the majority of agents must already be truthful for the mechanism to work).

In the case of MD, there has been comparatively little work on achieving efficient, incentive-compatible and individually-rational mechanisms that take into account *uncertainty* in general. An exception to this rule is the dAGVA mechanism [7] which considers the case when the types of agents are unknown to themselves but are drawn from a probability distribution of types which is common knowledge to all agents. However, in our case, the agents know their types and these incorporates their uncertainty related to fulfilling a task. Porter et al. [9] have also considered this case and their mechanism is the one that is most closely related to ours. However, they limit themselves to the case where agents can only report on their own POS. This is a drawback because it assumes the agents can measure their own POS accurately and it does not consider the case where this measure may be biased (i.e. different agents perceive the success of the same event differently). Thus our mechanism is a generalisation of theirs (see section 5 for the formal proof).

Finally, our work may also seem to be a case of interdependent, multidimensional allocation schemes [2] where there is an impossibility result of not being able to achieve efficiency when considering interdependent, multidimensional signals [5]. However, we circumvent this by relating the trust values to a probability that an allocation is completed, rather than to an absolute valuation or cost signal.

3. A Standard VCG Task Allocation Scheme

We consider a set of agents $\mathcal{I}$, where $\mathcal{I} = \{1, \ldots, I\}$, and a set of possible tasks $\mathcal{T}$. Each agent $i \in \mathcal{I}$ has a particular value, $v_i(\tau, \theta_i)$, for having a task (completed by another agent), $\tau \in \mathcal{T}$, which is dependent on its type θ_i drawn from a possible set of types, Θ_i. An agent i also has a cost, $c_i(\tau, \theta_i)$, of attempting to complete a task. Given a vector of values, $\boldsymbol{v}(\tau, \theta)$, and costs, $\boldsymbol{c}(\tau, \theta)$, from the set of agents, we can determine the value of an allocation $K \in \mathcal{K}$ where $\mathcal{K}$ is the set of all possible mappings of $\mathcal{T}$ to $\mathcal{I}$. Once a certain allocation K is implemented, an agent i is then asked to pay for the task(s) it requested or receive payment for the task(s) it performed. The overall transfer of money to a particular agent i is denoted by r_i. As is common in this domain, we assume that an agent is rational (expected utility maximiser) and has a quasi-linear utility function [7].

Definition 1. *A **quasi-linear utility function** is one that can be expressed as:*

$$u_i(K, r_i, \theta_i) = v_i(K, \theta_i) - c_i(K, \theta_i) + r_i \qquad (1)$$

In devising a mechanism for task allocation, we focus on *incentive compatible direct revelation* mechanisms (DRMs) by invoking the *revelation principle* which states that any mechanism can be transformed into a DRM. In this context, "direct revelation" means the strategy space (i.e. all possible actions) of the agents is restricted to reporting their types and "incentive compatible" means the equilibrium strategy (i.e best strategy under a certain equilibrium concept) is truth-telling. Hence, in our allocation scheme, the agents report their types to a centre which then decides on the allocation K and the reward vector $\boldsymbol{r}$ and reports these back to

1 The mechanism we develop also forms the only class of mechanisms that have these properties under a Nash equilibrium strategy when factoring trust into the decision making process. Intuitively, this follows from the uniqueness of the VCG which charges agents their marginal contribution to the system. Since we use a similar technique to develop our mechanism we believe the same result will ensue (the formal proof of this assertion is beyond the scope of this paper).

AAMAS'04, July 19-23, 2004, New York, New York, USA.

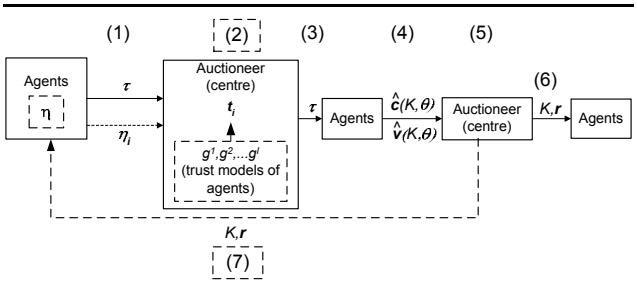

Figure 1. Simple task allocation model. The dotted lines represent the modifications we make to the mechanism when using trust in the feedback loop. The $g^{\mathcal{I}}$ functions represent the trust functions that are used to aggregate all POS values from other agents into a common measure of trust.

the agents. The problem at hand is then to find a mechanism $M(\boldsymbol{v}(\tau,\boldsymbol{\theta}),\boldsymbol{c}(\tau,\boldsymbol{\theta}))=\{K,\boldsymbol{r}\}$ that fulfills the following commonly sought objectives in MD:

- *Efficiency*: an allocation that maximises the total utility of all the agents in the system.
- *Individual Rationality*: an allocation scheme is individually rational if agents are willing to participate in the scheme rather than opting out of it. It is commonly assumed that the utility of an agent choosing to opt out of a scheme, $\underline{u_i}(.)$, is 0. Hence, it is sufficient to ensure that the agents derive a utility $u_i \geq 0$ by being in the system.
- *Incentive Compatibility*: an incentive compatible system is one in which the agents will find no better option than to reveal their true type.

Amongst the class of mechanisms that satisfy the above properties, the VCG mechanism implements an efficient allocation under dominant strategies (i.e. each agent has a best strategy no matter what other agents' strategies are) [7]. Using the VCG mechanism, our task allocation problem is then reduced to the following protocol which is shown in figure 1:

1. The centre receives the set of tasks $\boldsymbol{\tau}$ to be allocated from the agents (step 1).
2. The centre then posts these tasks in the vector $\boldsymbol{\tau}$ (step 3). Each agent i then reports its cost $\widehat{c}_i(K,\theta_i)$ (in the vector $\widehat{\boldsymbol{c}}(K,\boldsymbol{\theta})$) for completing a set of tasks in the set of allocations K along with the *reported* valuation $\widehat{v}_i(K,\theta_i)$ (in the vector $\widehat{\boldsymbol{v}}(K,\boldsymbol{\theta})$) it derives from having a set of tasks completed (step 4). In the rest of the paper, we will superscript with '$\widehat{\ }$' those variables and functions that are *reported* to the centre (auctioneer) to differentiate from those that are privately known. Of course, the reported values and costs can be different from the actual values and costs.
3. The centre then solves the following standard VCG auction equation (step 5): $\widehat{K}^* = \arg\max_{K\in\mathcal{K}} \sum_{i\in\mathcal{I}} [\widehat{v}_i(K,\theta_i) - \widehat{c}_i(K,\theta_i)]$ and computes each transfer r_i in the vector $\boldsymbol{r}$ as: $r_i = \left[\sum_{j\in -i}\left[\widehat{v}_j(\widehat{K}^*,\theta_j) - \widehat{c}_j(\widehat{K}^*,\theta_j)\right]\right] - \left[\max_{K\in\mathcal{K}} \sum_{j\in -i}[\widehat{v}_j(K,\theta_j) - \widehat{c}_j(K,\theta_j)]\right]$ where $-i \equiv \mathcal{I}\setminus i$.
4. The centre allocates the tasks according to the optimal allocation K^* and implements the transfers r_i (step 6).

The VCG mechanism results in an alignment of the goal of each agent with that of the mechanism designer via the use of the transfer part of the mechanism. Basically, each agent has as its best strategy the social optimum goal, which can only be achieved via a truthful revelation. That is, for each agent i, $\widehat{c}_i(K,\theta_i) = c_i(K,\theta_i)$ and $\widehat{v}_i(K,\theta_i) = v_i(K,\theta_i)$. Since the agents find it optimal to report their true valuations and costs, the centre thus finds the efficient allocation in step 3 (i.e. $\widehat{K}^* = K^*$). The second part of the transfer ensures that agents have $u_j \geq 0$ and thereby makes the mechanism incentive compatible.

We have thus presented a standard DRM for our task-allocation problem that achieves efficiency, incentive compatibility, and individual rationality under dominant strategy equilibrium. However, this mechanism only considers the cost and value of the tasks. In the next section we introduce trust as another dimension to be used in the computation of the efficient allocation and show why the standard VCG is neither incentive compatible nor efficient when trust is taken into account.

4. Trust-Based Mechanism Design

To incorporate trust, a further dimension needs to be added to the utility function in equation 1 which, in turn, requires both the allocation and payment schemes in the VCG mechanism to be modified to take this additional dimension into account. Having defined our mechanism (see section 4.4), we prove that it is incentive-compatible, efficient and individually-rational (in section 4.5). Before doing this, however, we first need to specify the generic properties that allow trust to be defined as a measure that can be used in computing efficient allocations.

4.1. Properties of the Trust Model

Many computational trust models have been developed to allow agents to choose their most trustworthy interaction partners (as discussed in section 2). However, at their most fundamental level, these models can be viewed as alternative approaches for achieving the following properties [2]:

2 Note that we do not focus on a particular trust model. This is because trust models implement the above properties in their own ways and in different contexts. Therefore, we concentrate on these abstract properties to keep the focus on the relationship between trust and the design of an efficient mechanism. In so doing, we ensure that the prop-

AAMAS'04, July 19-23, 2004, New York, New York, USA.

1. The trust measure of an agent i in an agent j depends both on i's perception of j's POS and on the perception of other agents on j's POS. This latter point encapsulates the concept of *reputation* whereby the society of agents generally attributes some characteristic to one of its members by aggregating some/all the opinions of its other members about that member. Thus, each agent considers this societal view on other members when building up its own measure of trust in its counterparts [1]. The trust of agent i in its counterpart j, $t_i^j \in [0,1]$, is given by a function, $g : [0,1]^{|\mathcal{I}|} \rightarrow [0,1]$, (which, in the simplest case, is a weighted sum) of all POS measures sent by other agents to agent i about agent j as shown below:

$$t_i^j = g(\{\eta_1^j, \ldots, \eta_i^j, \ldots, \eta_N^j\}) \quad (2)$$

where $\eta_i^j \in [0,1]$ is the POS of agent j as perceived by agent i and g is the function that combines both personal measures of POS and other agents' measures. In general, trust models compute the POS measures over multiple interactions. Thus, the level of success recorded in each interaction is normally averaged to give a representative value (see [10] for a general discussion on trust metrics).

2. Trust results from an analysis of an agent's POS in performing a given task. The more successful, the more trustworthy the agent is. Thus, the models assume that trust is monotonic increasing with POS. Therefore, the relationship between trust and POS is expressed as: $\frac{\partial t_i^j}{\partial \eta_i^j} > 0$, where t_i^j is the trust of i in agent j and η_i^j is the actual POS of agent j as perceived by i.

Given the above, agents can update their trust rating for another agent each time they interact (both by recording their view of the success of their counterpart and by gathering new reports from other agents about it). Thus, if an agent's POS does not change, the trust measure in it should become more precise as more observations are made and received from other agents. Moreover, having the trust monotonic increasing with POS ensures Mirrlees's condition regarding fixed points in allocation schemes [8] (which is a necessary condition for the mechanism to be efficient) is satisfied.

4.2. Augmenting the Task Allocation Scenario

In this section we show how trust is to be calculated and taken into account in the task allocation example we described in section 3. Here, any trust model satisfying the properties discussed in section 4.1 can be used when actually building the system. The following changes are made (as shown in figure 1):

- Each agent i reports to the centre their POS vector $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}} = [\widehat{\eta}_i^1 \ldots \widehat{\eta}_i^I]$ (step 1). This is the POS that an agent has observed about the other agents. This vector may not be complete if agents have not experienced any past interactions with other agents. However, this does not affect the properties of the mechanism since the centre will only pick those POSs that are relevant (and calculate trust according to these).
- The agents must also submit their respective trust calculation function (equation 2) that applies over the vector of all (or part of) other agents' reported POSs (i.e. $\widehat{\boldsymbol{\eta}}$), $\boldsymbol{t_i} = g(\widehat{\boldsymbol{\eta}})$, to the centre before the allocation of tasks (step 2). This allows the centre to compute the trust of agent i in all other agents (given i's own perception, as well as other agents' perceptions of the task performer's POS). Given that the trust t_i only affects the allocation of tasks originating from agent i, the latter has no incentive to lie about its trust function to the centre (otherwise it could result in i's task not being allocated to the agent deemed most trustworthy by i).

The trust function $g(.)$ may assign different weights to the reports of different agents depending on the level of similarity between the types of agents i and $-i$ (where $-i \equiv \mathcal{I} \setminus i$). Thus, given the trust functions and reports of POS of each agent, we now require the centre to maximise the overall *expected* valuation of the allocation (in step 5), as opposed to the valuation of the allocation independent of trust (i.e. which the standard VCG does). This is because an agent has a certain probability of completing the task to a degree of success which may be less than one. We denote as γ the *completion vector* of an allocation K which measures the level to which each task in an allocation is deemed completed. Thus, the expected value of an allocation is then $\left(E_{[\gamma|K,\boldsymbol{t_i}]}\left[\sum_{i\in\mathcal{I}} \widehat{v}_i(K,\theta_i)\right] - \sum_{i\in\mathcal{I}} \widehat{c}_i(K,\theta_i)\right)$ given the trust vector $\boldsymbol{t_i}$. This captures the fact that the agent i, that allocated the task, determines the value of γ. Moreover, agent j, to which the task has been allocated, incurs a cost independent of how agent i evaluates the task. This effectively means that the valuations are non-deterministic while the costs are deterministic. The centre thus determines the efficient allocation K^* (step 7) such that the value of the efficient allocation is maximised.

Having shown how to fit trust into the process of determining the value of allocations, in the next subsection we provide a simple example to show why the standard VCG solution of section 3 is not incentive compatible (and thus not efficient). This then motivates the search for a mechanism that is.

4.3. Failure of the VCG Solution

Consider a system of four agents where agent 4 has asked for a task τ to be allocated and its valuation of this task is $v_4(\tau,\theta_4) = 210$. Each agent i has a cost c_i to perform the task proposed by 4 (agent 4 has infinite cost to perform the

erties of our mechanism are independent of any specific trust model.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

Agent i	c_i	η_i^1	η_i^2	η_i^3	t_4^i
1	40	0.4	1.0	0.8	0.5
2	80	0.6	1.0	0.8	1.0
3	50	0.5	1.0	0.9	0.86
4	∞	0.525	1.0	0.95	na

Table 1. A set of four agents in which agent 4 has proposed a task.

task by itself) and does not derive any value from the task being performed. Now, suppose that the trust function of agent 4 is a weighed sum of the POS reports by the agents (i.e. $t_4^i = \alpha \cdot \widehat{\boldsymbol{\eta}}^{\boldsymbol{i}}$ where $\alpha = [0.3 \;\; 0.2 \;\; 0.1 \;\; 0.4]$). Note that we do not concern ourselves with the reports η_i^4 since the task is proposed by agent 4 itself. Table 1 shows the cost c_i of attempting the task, and the observed POS value of each agent, $\boldsymbol{\eta_i}$, as well as the trust computed by agent 4, t_4^i, if each agent reports truthfully on its $\boldsymbol{\eta_i}$.

The VCG solution of section 3 determines the allocation and payments based only on cost and valuations. However, this would clearly fail to find an efficient allocation since agent 1 would be allocated the task despite being the least trusted and hence most likely to fail. If we instead implemented the VCG mechanism with the *expected* valuations (taking into account the trust and POS reports), we then have $K^* = [0010]$ (i.e agent 3 is allocated the task), $r_1 = r_2 = 0$ and $r_3 = 210\gamma - 130$. Thus, agent 3 will then derive an average payment of $0.87 \times 210 - 130 = 52.7$. However, this scheme is not incentive-compatible because agent 2 can lie about η_2^3 by reporting $\widehat{\eta}_2^3 \leq 0.7357$ which will then lead to agent 2 being allocated the task and deriving a positive utility from this allocation. Note that this scheme is exactly that of [9] for a single-task scenario (with the modification that we use γ as a level of success rather than a binary indicator function of success or failure).

As can be seen, the VCG mechanism needs to be extended to circumvent this problem. Specifically, we require a mechanism that is efficient given the reports of the agents on their costs and valuations of allocations, as well as their observed POS vector (since the VCG is affected by false reports of POS). In effect, we need to change the payment scheme so as to make the truthful-reporting of POSs an optimal strategy for the agent again. Once this is achieved, the centre can then choose the efficient allocation based on expected utilities. The difficulty with designing such a mechanism is that the centre cannot check on the validity of POS reports of agents because it is based on a private observation carried out by the agent. Thus two agents may legitimately differ in their observed POS of another agent due to their different interaction histories with that agent.

4.4. The Trust-Based Mechanism

Before presenting our trust-based mechanism (TBM), we first introduce some notation. Let the sum of utilities of all agents in a system given an allocation K and a completion vector γ be denoted as $U(K, \boldsymbol{\theta}, \gamma) = \sum_{i \in \mathcal{I}} v_i(K, \theta_i, \gamma) - \sum_{i \in \mathcal{I}} c_i(K, \theta_i)$. Then the expected utility $\overline{U}(K, \boldsymbol{\theta}, \gamma)$ before the allocation is carried out is $E_{[\gamma|K, \boldsymbol{t_i}]}[U(K, \boldsymbol{\theta}, \gamma)]$ where $\boldsymbol{\theta}$ is the vector containing all agent types. We also denote the marginal contribution of the agent i to the system given an efficient allocation $\widehat{K}^*$ as $mc_i = \overline{U}_{-i}(\widehat{K}^*, \boldsymbol{\theta}, \gamma) - \max_{K \in \mathcal{K}} \left[\overline{U}_{-i}(K, \boldsymbol{\theta_{-i}}, \gamma)\right]$ where $\max_{K \in \mathcal{K}} \left[\overline{U}_{-i}(K, \boldsymbol{\theta_{-i}}, \gamma)\right]$ is the overall expected utility of the efficient allocation that would have resulted if agent i were not present in the system. Now, we can detail TBM:

1. Find the efficient allocation $\widehat{K}^*$ such that:

$$\widehat{K}^* = \arg\max_{K \in \mathcal{K}} \overline{U}(K, \boldsymbol{\theta}, \gamma) \tag{3}$$

   This finds the best allocation; that is, the one that maximises the sum of *expected utilities* of the agents, conditional on the reports of the agents. We note here that we do not take into consideration the reward functions of the agents when calculating the overall utility since these rewards are from one agent to another and therefore do not make a difference when calculating the overall utility of the agents.

2. We now calculate the efficient allocation that would have resulted if agent i's report is taken out: $K^*_{-i} = \arg\max_{K \in \mathcal{K}} E_{[\gamma|K, \boldsymbol{t'_i}]}[U(K, \boldsymbol{\theta}, \gamma)]$ where $\boldsymbol{t'_i} = g(\widehat{\boldsymbol{\eta}} \backslash \widehat{\boldsymbol{\eta}}_{\boldsymbol{i}})$. This computes how $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ affects which allocation is deemed efficient.

3. We now find the effect that an agent's $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ has had on its marginal contribution. Thus, find $D_i = \overline{U}(\widehat{K}^*, .) - \overline{U}(K^*_{-i}, .)$. This distils the effect of an agent's $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ report on its marginal contribution.

4. Given K^*, the payment r_i made to the agent i is then:

$$r_i = mc_i - D_i \tag{4}$$

   Naturally, if r_i is negative it implies that i makes a payment to the centre. The first part of the payment scheme, mc_i, calculates the effect that an agent's *presence* has had on overall expected utility of the system. We also subtract D_i to take into account the effect that an agent's POS report has on the chosen allocation. This is in line with the intuition behind VCG mechanisms in which an agent's report affects the allocation but not the payment it receives or gives.

We will now prove each of the properties of TBM in turn whilst intuitively explaining why the mechanism has the aforementioned properties.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

4.5. Properties of TBM

Proposition 1. *TBM is incentive-compatible in ex-ante Nash Equilibrium.*

Proof. We first need to calculate the expected utility, $E_{[\gamma|K,\boldsymbol{t_i}]}\left[u_i(K,\theta_i,\gamma)\right]$, that an agent derives from TBM because the goal of a rational agent is to maximise its expected utility. We note here that we are assuming that the agent is myopic in that it is only concerned with its current expected utility given the cost vector, $\boldsymbol{c}(K,\boldsymbol{\theta})$, the value vector, $\boldsymbol{v}(K,\boldsymbol{\theta})$, and the trust vector $\boldsymbol{t}$. The expected utility that an agent, $\overline{u}_i(\widehat{K}^*,\theta_i,\gamma)$, derives from an efficient allocation, as calculated from equation 3, given the reports of all agents in the system is:

$$\begin{aligned}\overline{u}_i(\widehat{K}^*,\theta_i,\gamma) &= E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma)\right] - c_i(\widehat{K}^*,\theta_i) \\ &\quad + mc_i(\widehat{K}^*,\theta_i,\gamma) - D_i \\ &= E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma) - \widehat{v}_i(\widehat{K}^*,\theta_i,\gamma)\right] \\ &\quad - \Big(c_i(\widehat{K}^*,\theta_i) - \widehat{c}_i(\widehat{K}^*,\theta_i)\Big) + \\ &\quad \overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma) - \max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right] \quad (5)\end{aligned}$$

¿From 5 we will firstly prove the following lemma:

Lemma 1. *An agent has an equilibrium strategy to reveal its observed POS values.*

Proof. We consider how $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ affects $\overline{u}_i(\widehat{K}^*,\theta_i,\gamma)$. From equation 5 we observe that $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ cannot affect $\overline{U}(K_{-i},\boldsymbol{\theta},\gamma) - \max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right]$. Thus, an agent only has an incentive to lie so that $\widehat{K}^*$ is selected such that $E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma) - \widehat{v}_i(\widehat{K}^*,\theta_i,\gamma)\right] - \Big(c_i(\widehat{K}^*,\theta_i) - \widehat{c}_i(\widehat{K}^*,\theta_i)\Big)$ is maximised. If an agent reveals its cost and valuation truthfully i.e. $\widehat{v}(.) = v(.)$ and $\widehat{c}(.) = c$, we then have the term as zero. Then an agent cannot gain from an untruthful reporting of $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$. If, however, an agent is to gain from such an untruthful reporting, it needs to set either $\widehat{v}(.) < v(.)$ or $\widehat{c}(.) > c$ or both. However, doing so would decrease the chance of i successfully allocating a task or winning an allocation. Therefore, i would not reveal untruthful values for $\widehat{c}(.)$ and $\widehat{v}(.)$. Moreover, i will actually report truthfully its $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}}$ since this allows the centre to choose those agents that i deems to have a high POS (as well as helping other agents choose i as having a perception close to theirs). Thus, reporting $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}} = \boldsymbol{\eta}_{\boldsymbol{i}}$ is an ex-ante Nash equilibrium strategy. □

Given lemma 1, we can now show that TBM is incentive compatible. Suppose an agent is truthful about $\widehat{v}(.)$ and $\widehat{c}(.)$. Then it derives as utility $\overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma) - \max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right]$. Now assume that the agent lies about $\widehat{v}(.)$ and $\widehat{c}(.)$ so as to increase its utility. This then means that $E_{[\gamma|\widehat{K}^*,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma) - \widehat{v}_i(\widehat{K}^*,\theta_i,\gamma)\right] - \Big(c_i(\widehat{K}^*,\theta_i) - \widehat{c}_i(\widehat{K}^*,\theta_i)\Big) + \overline{U}(K'_{-i},\boldsymbol{\theta},\gamma) > \overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma)$ where K'_{-i} is the efficient allocation found with $\widehat{c}(.)$ and $\widehat{v}(.)$ without the report of $\boldsymbol{\eta}_{\boldsymbol{i}}$. However, as argued earlier, an agent would not report a lower value or a higher cost. Thus $E_{[\gamma|K,\boldsymbol{t_i}]}\left[v_i(\widehat{K}^*,\theta_i,\gamma) - \widehat{v}_i(\widehat{K}^*,\theta_i,\gamma)\right] - \Big(c_i(\widehat{K}^*,\theta_i) - \widehat{c}_i(\widehat{K}^*,\theta_i)\Big) \leq 0$. Furthermore, by the maximisation of step 2 of TBM, $\overline{U}(K'_{-i},\boldsymbol{\theta},\gamma) < \overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma)$ if all other agents report truthfully. Thus, TBM is incentive-compatible in an ex-ante Nash equilibrium. □

Proposition 2. *TBM is efficient.*

Proof. Given that the agents are incentivised to report truthfully (proposition 1), the centre will calculate the efficient allocation according to equation 3 (i.e. $\widehat{K}^* = K^*$). □

Proposition 3. *TBM is individually-rational (in expected utility).*

Proof. We need to show that the expected utility of any agent from an efficient allocation K^* is greater than if the agent were not in the scheme (i.e. $\overline{u}_i(K^*,\theta_i,\gamma) \geq 0$). As a result of the inherent uncertainty in the completion of tasks, we cannot guarantee that the mechanism will be ex-post individually-rational for an agent. Rather, we prove that the mechanism is individually-rational for an agent if we consider expected utility. Given truthful reports, the utility of an agent from equation 5 is $\overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma) - \max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right]$. The first maximisation is carried out without the reports $\boldsymbol{\eta_i^{-i}}$,whereas the second maximisation is carried out over the set of agents $\mathcal{I}\setminus i$. Thus, the second maximisation is carried out over a smaller set than the first one. As a result $\max_{K\in\mathcal{K}}\left[\overline{U}_{-i}(K,\boldsymbol{\theta_{-i}},\gamma)\right] \geq \overline{U}(K^*_{-i},\boldsymbol{\theta},\gamma)$ such that $\overline{u}_i(K^*,\theta_i,\gamma) \geq 0$. □

5. Instances of TBM

TBM can be viewed as a generalised version of the VCG mechanism in which there exist uncertainties about whether a set of agents will carry out an allocation and about the relevance of reports of POS by agents. In this section, we demonstrate its generality by analysing two specific instances of the mechanism.

5.1. Self-POS Reports Only

The non-combinatorial mechanism developed in [9] is a special case of TBM. Specifically, agents only report on their own POS (i.e. $\widehat{\boldsymbol{\eta}}_{\boldsymbol{i}} = \widehat{\eta}^i_i$) and agents assign a relevance of 1 to reports by all other agents. However, since in their model there is no notion of varying perceptions of success, we need to introduce the notion of a *report agent* that has $v(K,.) = 0$ and $c(K,.) = \infty$. This acts as a proxy to agents reporting the ex-post POS to the centre. This also caters for the problem of single POS reports as there is then no measure of t^j_i once j's report is removed (and hence $\overline{U}(K^*_{-i},.)$

AAMAS'04, July 19-23, 2004, New York, New York, USA.

is undefined). The centre then calculates the efficient allocation as: $K^* = \arg\max_{K \in \mathcal{K}} \left[\overline{U}(\widehat{K}^*, \boldsymbol{\theta}, \gamma)\right]$ and the payment to agent i is $r_i = mc_i - D_i = mc_i$. The term $D_i = 0$ since, as a result of the report agent, $\overline{U}(\widehat{K}^*, .) = \overline{U}(K^*_{-i}, .)$ (because $\boldsymbol{t}$ is equal in both cases).

5.2. Single-Task Scenario

Consider the single task scenario (as presented in table 1) where an agent k proposes a single task τ_k. Using equation 3, the efficient allocation is then simplified to:
$K^* = \arg\max_{K \in \mathcal{K}} \left[\overline{U}(\widehat{K}^*, \boldsymbol{\theta}, \gamma)\right]$.

The payment to agent i, from equation 4, is then:
$r_i = E_{[\gamma|K^*_{-i}, \boldsymbol{t_k}]}\left[v_k(K^*_{-i}, \theta_k, \gamma)\right] + \widehat{c}_i(\widehat{K}^*, \theta_i, \gamma) - \sum_{i \in \mathcal{I}} \widehat{c}_i(K^*_{-i}, \theta_i, \gamma) - \max_{K \in \mathcal{K}} \left[E_{[\gamma|K, \boldsymbol{t_k}]}\left[v_k(K, \theta_k, \gamma)\right] - \sum_{j \in -i} \widehat{c}_j(K, \boldsymbol{\theta_{-i}}, \gamma)\right]$.

Since the above single-task scenario is an instance of the TBM, it is still incentive compatible. Therefore, when applying the above allocation scheme to the example, we can take the reported values of the agents as being truthful. Given this, the efficient allocation is agent 3 getting to do the task. Then, we need to check whether agent 3's report has made itself more attractive. To do so, we remove the report of agent 3 and end up with agent 4 having a trust vector $\boldsymbol{t_4^i} = [0.5 \;\; 1.0 \;\; 0.9]$ which again leads to agent 3 being allocated the task. Thus agent 3 will get an expected utility of $210 * 0.8667 - \;\; 50 + \;\; 50 \;\; - 130 - 50 = \;\; 2$. Agent 1 and 2 no longer have an incentive to lie about the POSs since this would not increase their utility. However, suppose that, after the allocation, every type becomes common knowledge. Then agent 2 can deduce that lying about its costs *and* reported POS would allow its utility to increase. This would have been maximised when agent 2 reports $\widehat{c}_2(.) = 110$ and $\widehat{\eta}_2^3 = 0$. However, before the allocation is carried out and payments are made, agent 2 would not know about the private types of other agents and may reduce its chance of deriving a positive utility by reporting $\widehat{c}_2(.) > c_2(.)$. Furthermore, agent 2 does not report $\widehat{\boldsymbol{\eta}}_{\mathbf{2}}^{\mathbf{-2}} < \boldsymbol{\eta}_{\mathbf{2}}^{\mathbf{-2}}$ since then $u_2(.) = 0$ even if it wins the allocation. A similar argument applies to agent 1. Thus, the mechanism has an exante Nash Equilibrium of truthful reporting.

6. Experimental Evaluation

Here we empirically evaluate TBM by comparing it with the fault tolerant mechanism (FTM) of [9] (this is chosen because it also deals with the POS of agents as discussed in sections 2 and 5.1) and the standard VCG. We refer to task performing agents as contractors in what follows. In our experiments we perform 500 successive allocations, in the scenario described in section 4, with six agents each given one task to complete. After each allocation, contractors perform tasks and the level of success is measured and reported to all agents. Each agent can then update its measure of the contractors' POSs as well as the contractors' trustworthiness as discussed in section 4.1. The valuations and POS of each agent are obtained from a uniform distribution and the costs are the same for all tasks. We iterate the process and average the results (here for 200 iterations). Given the properties of TBM and FTM we postulate the following hypotheses and validate them as shown below:

Hypothesis 1. *TBM always chooses the efficient allocation (K^*) in the long run.*

This hypothesis reflects the fact that we expect agents in TBM to take a number of interactions to model the true POS of their counterparts, using their individual trust models. After this time, however, the mechanism can choose those contractors that are most successful at completing a given task. As can be seen in figure 2, the optimal allocation chosen by TBM, K^*TBM, reaches the efficient allocation K^* (given *real* POSs) after 116 interactions.[3] After 116 interactions, the POS of each contractor is accurately modelled, as is the trust of agents in their contractors. Thus, the most trusted and utility maximising allocation is found by the TBM. This result is observed for all cases where the POSs of contractors are varied.

Hypothesis 2. *TBM finds better allocations than FTM when contractors' own reported POS are biased.*

While FTM only takes into account a contractor's own reports, TBM uses the trust model of the various individual agents (which take into account reports not only from the contractor) to make an allocation. In the particular trust model we use in TBM, an agent can give different weights to reports from different agents (as shown in section 4.3). We therefore varied the weight w, assigned to a contractor's report of its own POS in the trust model of an agent. Here we exemplify the cases where $w = 0.5$ (i.e. the contractor's report is given equal weighting to the agent's perceived POS), $w = 0.25$ and $w = 0$ (i.e. no importance is given to the contractor's report).

As can be seen, our hypothesis is validated by the results given in figure 2 (with normalised expected values). Note here that K^*VCG is the allocation independent of POSs or if POSs of agents are all equal. We note as K^*TBM_w the allocation chosen by TBM with a weight w. In more detail, TBM_0 (i.e. TBM) reaches the optimal allocation K^* (i.e. equivalent to zero bias from the seller) after 116 iterations, while $TBM_{0.25}$ and $TBM_{0.5}$ settle around a suboptimal allocation (the expected value of which decreases with increasing w). Moreover, FTM is seen to settle at $K^*FTM = 0.8$ after 82 iterations. In general, it is noted that FTM always settles at $K^*FTM < K^*$ (and some-

3 The results were validated using a student's t-test with two samples of 100 and 200 iterations assuming equal variances with means $\mu_1 = 0.99999$ and $\mu_2 = 1.0$ and p-value $p = 0.778528$. This means that the difference between the means is not significant.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

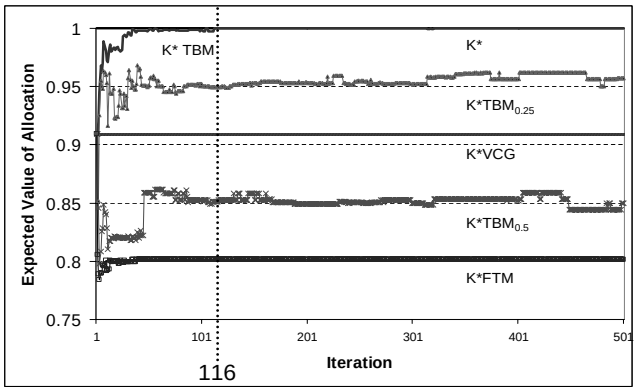

Figure 2. Expected value of chosen allocations for TBM and FTM where $K^* = 1$, $K^*VCG = 0.909$, and at equilibrium, $K^*TBM = 1$, $0.97 > K^*TBM_{0.25} > 0.94$, $0.86 > K^*TBM_{0.5} > 0.84$, and $K^*FTM = 0.8$.

times even $K^*FTM < K^*VCG$ as in figure 2) depending on the valuations agents have for the tasks. This result is explained by the fact that the biased reports cause biased trust values to be obtained by the centre which then chooses a sub-optimal allocation (i.e. less than K^* which chooses agents according to their 'real POSs'). $TBM_{0.25}$ and $TBM_{0.5}$ are less affected by biased reports since the weighted trust model reduces the effect of bias on the overall trust values (but still affects the mechanism). In most trust models, however, $w \geq 0.5$ is never given to the contractors' POS report and here it only represents an extreme case [10]. Moreover, if the bias is removed, then FTM and the weighted TBMs behave the same as TBM since the agents then perceive the same POS and all achieve K^*. It was also observed that the speed with which TBM and FTM achieve K^* also depends on the difference between the optimum allocation and other allocations. This is because the smaller the differences, the harder it becomes to differentiate these allocations given imperfect estimations of POSs (i.e. the larger the samples, the more accurate the POSs are, hence the longer the learning rate).

7. Conclusions ansd Future Work

In this paper we have introduced the notion of trust-based mechanism design (TBMD) which generalises the VCG mechanism by using the trust model of individual agents in order to generate efficient allocations. We have developed a trust-based mechanism (TBM) and proved that it is efficient, individually rational, and incentive compatible. Moreover, we have empirically evaluated TBM and shown that it *always* achieves the optimum allocation in the long run and achieves better allocations than its closest comparison when contractors provide biased reports of POS.

Future empirical work will develop trust models that learn the similarity between the different types of agents to achieve more efficient allocations faster in a context where agents may be of different types (i.e. where different groups of agents sense different *degrees* of success). Theoretical work will focus on showing that our TBM is the only class of mechanism that can result in efficient allocations based on trust reports. Moreover we will remove the assumption of myopicity of agents and develop a mechanism which considers how agents might strategise over rounds of allocation. We also aim to show instances where the comparatively brittle Nash equilibrium can be strengthened as a result of ex-post actions (such as checking by the centre [9], or rewarding for past performance). Finally we will attempt to make the mechanism group incentive compatible (i.e. being robust to collusion) by developing a cross-monotonic payment scheme.

References

[1] P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Blackwell, 1998.

[2] P. Dasgupta and E. Maskin. Efficient auctions. *Quarterly Journal of Economics*, 115:341–388, 2000.

[3] R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

[4] C. Dellarocas. Goodwill hunting: An economically efficient online feedback mechanism for environments with variable product quality. In *Proc. of the Workshop on Agent-Mediated Electronic Commerce*, pages 238–252, 2002.

[5] P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69(5):1237–59, 2001.

[6] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *Proc. of the IEEE Conf. on E-Commerce*, pages 285–292, 2003.

[7] A. MasColell, M. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[8] R. Mirrlees. An exploration in the theory of optimum income taxation. *Review of Economic Studies*, 38:175–208, 1971.

[9] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *Proc. of the Int. Conf. on Uncertainty in AI*, pages 414–421, 2002.

[10] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 2004.

[11] T. Sandholm. Making markets and democracy work: A story of incentives and computing. In *Proceedings of the Int. Joint Conf. on AI*, pages 1649–1671, 2003.

AAMAS'04, July 19-23, 2004, New York, New York, USA.