Strategic Deliberation and Truthful Revelation: An Impossibility Result*

Kate Larson and Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
{klarson,sandholm}@cs.cmu.edu

Abstract

In many settings, agents participating in mechanisms do not know their preferences *a priori*. Instead, they must actively determine them through deliberation (e.g., information processing or information gathering). Agents are faced not only with the problem of deciding how to reveal their preferences to the mechanism but also how to deliberate in order to determine their preferences. For such settings, we have introduced the *deliberation equilibrium* as the game-theoretic solution concept where the agents' deliberation actions are modeled as part of their strategies. In this paper, we lay out mechanism design principles for such deliberative agents.

We also derive the first impossibility results for such settings - specifically for private-value auctions where the agents' utility functions are quasilinear, but the agents can only determine their valuations through deliberation. We propose a set of intuitive properties which are desirable in mechanisms used among deliberative agents. First, mechanisms should be *non-deliberative*: the mechanism should not be solving the deliberation problems for the agents. Secondly, mechanisms should be *deliberation-proof*: agents should not deliberate on others' valuations in equilibrium. Third, the mechanism should be *non-deceiving*: agents do not strategically misrepresent. Finally, the mechanism should be *sensitive*: the agents' actions should affect the outcome. We show that no direct-revelation mechanism satisfies these four properties. Moving beyond direct-revelation mechanisms, we show that no *value-based* mechanism (that is, mechanism where the agents are only asked to report valuations - either partially or fully determined ones) satisfies these four properties.

1 Introduction

Game theory, and mechanism design in particular, have long been successfully used in economics and have recently drawn a lot of research interest from computer scientists (e.g., [13] [14]). In most of this work it is assumed that the participants, or agents, know their preferences and the goal of the mechanism is to extract this information to a sufficient extent, and select an outcome such that desirable properties are achieved. However, there are many settings where agents do not know their preferences *a priori*. Instead they may, for example, have to solve computationally complex optimization problems, query databases, or perform complicated searches in order to determine the worth of an outcome. We call the actions taken to determine preferences *deliberation*.

If there are no restrictions placed on the deliberation capabilities of agents, they could optimally determine their preferences and act as fully rational agents. However, in many settings there are costs associated with deliberation. Agents are not able to optimally determine their preferences, but instead must trade off quality of valuations against deliberation cost. Decision making under costly deliberation resources is challenging even in single-agent settings. Having to interact with other agents complicates the problem further. Agents must take into account the other agents' actions in determining both how to act in the mechanism and also how to use deliberation resources.

We have proposed explicitly including the deliberation actions of agents into their strategies, and then analyzing games for *deliberation equilibria* which are fixed points in the space of strategy profiles from this enlarged strategy space [7] [8]. Using this approach, we have studied common auction mechanisms such as the first-price auction, Vickrey auction, ascending

*Submitted, under review.

auction, descending auction, and generalized Vickrey auction. We discovered the existence of interesting strategic behavior. In each auction mechanism studied, there existed instances where, in equilibrium, agents would use their deliberation resources to determine *other agents' valuations* of the item(s) being auctioned. We coined this phenomenon *strategic deliberation*.

In this paper, we build on this body of work. Instead of looking at the properties of mechanisms that were designed for fully rational agents, we ask the question: "Is it possible to *design* mechanisms that have desirable properties for such deliberative agents?" We propose a set of weak, intuitive properties that are desirable for mechanisms designed for such agents. In particular, we propose that mechanisms should not solve the deliberation problems for the agents, that strategic deliberation should not occur in equilibrium, that agents should not have incentive to misreport, and that the agents' actions affect the outcome. We show that no direct-revelation mechanism satisfies these four properties. Moving beyond direct-revelation mechanisms, we show that no *value-based* mechanism (that is, mechanism where the agents are only asked to report valuations - either partially or fully determined ones) satisfies these four properties.

The rest of the paper is as follows. We first provide an example application where our approach is needed (Section 2). We then given an overview of pertinent mechanism design concepts, and describe the model for deliberative agents (Sections 3 and 4). We show that there is a parallel to the revelation principle for our setting, but argue that the direct mechanism produced in the proof has highly impractical properties. We then propose a set of mechanism properties which are important when the mechanism is to be used among deliberative agents. Our main results show that it is impossible to design a direct-revelation mechanism that satisfies those desirable properties, and furthermore, it is impossible to design any value-based mechanism that satisfies them (Section 5).

2 An Example Application

To make the presentation more concrete, we now discuss an example domain where our methods are needed. Consider a distributed vehicle routing problem with two geographically dispersed dispatch centers that are self-interested companies (Figure 1) [19] [21]. Each center is responsible for certain tasks (deliveries) and has a certain set of resources (vehicles) to take care of them. So each agent—representing a dispatch center—has its own vehicles and delivery tasks.

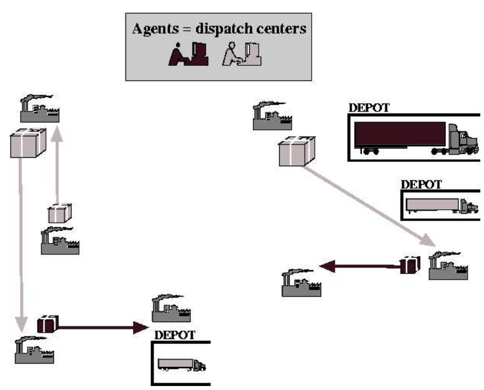

Figure 1: *Small example problem instance of the distributed vehicle routing problem. This instance has two dispatch centers represented in the figure by computer operators. They receive the delivery orders and route the vehicles. The light dispatch center has light tasks and trucks while the dark dispatch center has darker tasks and trucks. The dispatch centers receive all of their delivery orders at once, and then have some time to compute a routing solution before the trucks need to be dispatched. For example, in some practical settings, the delivery tasks are known by Friday evening and the route plan for the next week has to be ready by Monday morning when the trucks need to be dispatched.*

Each agent's problem is to minimize transportation costs (driven mileage) while still making all of its deliveries while honoring the following constraints: 1) Each vehicle has to begin and end its tour at the depot of its center (but neither the pickup nor the drop–off locations of the orders need to be at the depot), 2) Each vehicle has a maximum load weight and maximum load volume constraint, 3) Each vehicle has a maximum route length (prescribed by law), and 5) Each delivery has to be included in the route of some vehicle. This problem is $\mathcal{NP}$-complete.

Assume that an additional task is to be allocated to a dispatch center via some auction mechanism. Before agents' can formulate and submit bids, they must first know how they value the new task. This requires determining the cost of incorporating the new task into the current delivery schedule which potentially requires solving two $\mathcal{NP}$-complete problems (one without the new task, and one with the new task). The resources available to the agents to solve these problems may be limited. For example, the agents may have deadlines by which they require a solution or computing may be costly. Each agent must carefully consider the tradeoff they are willing to make on solution quality given the restrictions on their computing resources. It may also be in an agent's best-interest to use some of its deliberating resources to (partially) determine the values of the other agents. By doing some initial deliberating on a competitor's problem, an agent can gather information that may be useful in formulating its own deliberating and bidding strategies. By carefully designing the auction rules, it may be possible to reduce the strategic burden placed on agents.

3 Mechanism Design

In this section we present an overview of pertinent mechanism design concepts. We assume that the reader has a basic background in game theory. The mechanism design problem is to implement an optimal system-wide solution to a decentralized optimization problem with self-interested agents with private information about their preferences for different outcomes. In recent years mechanism design has been used in many important applications such as electronic market design and resource allocation problems.

We assume that there is a set of agents, I, $|I| = n$. Each agent, i, has a *type*, $\theta_i \in \Theta_i$, which represents the private information of the agent that is relevant to the agent's decision making. In particular, an agent's type determines its preferences over different outcomes. We use the notation $u_i(o, \theta_i)$ to denote the utility of agent i with type θ_i for outcome $o \in \mathcal{O}$ ($\mathcal{O}$ is the space of possible outcomes). As mentioned in the first paragraph, the goal of mechanism design is to implement some system-wide solution. This is defined in terms of a *social choice function*.

Definition 1 (Social Choice Function). *A* social choice function *is a function $f : \Theta_1 \times \ldots \times \Theta_n \mapsto \mathcal{O}$, such that, for each possible profile of agents' types $\theta = (\theta_1, \ldots, \theta_n)$ assigns an outcome $f(\theta) \in \mathcal{O}$.*

The mechanism design problem is to implement a set of "rules" so that the solution to the social choice function is implemented despite agents' acting in their own self-interest.

Definition 2 (Mechanism). *A mechanism $M = (S_1, \ldots, S_n, g(\cdot))$ defines the set of strategies S_i available to each agent and an outcome rule $g : S_1 \times \ldots \times S_n :\mapsto \mathcal{O}$, such that $g(s)$ is the outcome implemented by the mechanism for strategy profile $s = (s_1, \ldots, s_n)$.*

A mechanism *implements* a social choice function $f(\cdot)$ if there is an equilibrium of the game induced by the mechanism which results in the same outcomes as $f(\cdot)$ for every profile of types, θ.

Definition 3 (Implementation). *A mechanism $M = (S_1, \ldots, S_n, g(\cdot))$* implements *social choice function $f(\cdot)$ if there is an equilibrium strategy profile $s^* = (s_1^*, \ldots, s_n^*)$ such that $g(s^*(\theta)) = f(\theta)$ for all θ.*

An important class of mechanisms are *direct revelation mechanisms*.

Definition 4 (Direct revelation mechanism). *A* direct revelation mechanism *is a mechanism in which $s_i = \Theta_i$ for all i and has outcomes rule $g(\hat{\theta})$ based on reported types $\hat{\theta} = (\hat{\theta_1}, \ldots, \hat{\theta_n})$.*

One of the most important results in mechanism design is the *Revelation Principle*. It states that any mechanism can be transformed into an equivalent direct mechanism where, in equilibrium, all agents truthfully reveal their types (that is, the mechanism is *incentive compatible*).

Theorem 1 (Revelation Principle). *Suppose there exists a mechanism M that implements the social choice function $f(\cdot)$ in dominant strategies (Bayesian-Nash equilibrium). Then $f(\cdot)$ is truthfully implementable in a dominant strategies (Bayesian-Nash) incentive compatible direct-revelation mechanism.*

The Revelation Principle suggests that mechanism designers need only be concerned with direct-revelation mechanisms. Later in the paper we will discuss the Revelation Principle in more detail.

In this paper we restrict ourselves to settings where agents have *quasilinear preferences*.

Definition 5 (Quasilinear Preferences). *A quasilinear utility function for agent i with type θ_i is of the form:*

$$u_i(o, \theta_i) = v_i(x, \theta_i) + t_i$$

where outcome o defines a choice $x \in \mathcal{K}$ from a discrete choice set $\mathcal{K}$ and a transfer t_i by the agent. The notation $v_i(x, \theta_i)$ represents the valuation function of agent i, that is, the value the agent places on choice $x \in \mathcal{K}$.

In general, mechanisms for quasilinear preferences take a certain form.

Definition 6 (Mechanisms for quasilinear environments). *A mechanism for quasilinear environments is a mechanism* $M = (S_1, \ldots, S_n, (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot)))$ *such that the outcome function* $g(\cdot) = (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot))$ *where* $k : S_1 \times \ldots \times S_n :\mapsto \mathcal{K}$ *is a choice rule which selects some choice from choice set* $\mathcal{K}$*, and transfer rules* $t_i : S_1 \times \ldots \times S_n :\mapsto \mathbb{R}$*. one for each agent, compute the payment* $t_i(s)$ *made by agent* i*.*

In addition to quasilinear environments, we also assume that agents' have private values. This means that an agent's utility depends only on its own type. Many ecommerce applications are set in the quasilinear private-value environment. For example, many auctions belong to this set of mechanisms. The choice rule $k(\cdot)$ specifies which agents are allocated which items, and the transfers, $t_i(\cdot)$, specify the amount each agent must pay.

4 Deliberative Agents

To participate in a mechanism, agents need to know their preferences over different outcomes. In quasilinear environments, like those studied in this paper, the preferences of agents are represented by valuation functions which assign a numerical value to each outcome, given the agent's type.[1] In this paper we study settings where agents do not simply know their valuation functions. Instead agents are deliberative; they must deliberate in order to determine their valuations. In this section we present a model for deliberative agents.

4.1 Deliberative Agent Model

We assume that agents do not know their valuation functions *a priori*. Instead, each agent i has some amount of deliberation resources R_i that it can use to determine valuation functions for different problems. For each valuation problem j, agent i has some algorithm A_i^j that it uses to determine valuations. It may use the same algorithm for multiple problems and has algorithms for other agents' valuation problems.

We assume that all algorithms are *anytime algorithms*. The defining property of an anytime algorithm is that it can be stopped at any point and it will return a solution. If the algorithm is given additional resources then the quality of the solution improves. Many algorithms have anytime properties. For example, most iterative refinement algorithms are anytime since they always return a solution, and improve it if allowed to run longer [11]. Similarly, many search and information gathering applications can be viewed as anytime algorithms. As more information about an item is obtained, the knowledge about the true valuation improves.

Deliberation can either *improve* or *refine* agents' valuations. Let $v_i^j(r_j)$ be the value obtained by agent i for problem j by supplying r_j deliberation resources to algorithm A_i^j. If an agent deliberates to improve its valuations, then for all amounts of deliberation resources r, r' such that $r \leq r'$, $v_i^j(r) \leq v_i^j(r')$. In particular, deliberating changes the actual valuation for the problem. The agent's true valuation for problem j after devoting r_j resources to the problem is $v_i^j(r_j)$. If an agent refines its value, its deliberating does not change the valuation. Instead, there is a preexisting valuation which the agent learns through deliberating. The agent starts with a probability distribution over the space of valuations. By deliberating, the probability distribution is refined. The agent determines the true valuation of the outcome only when the distribution becomes a point mass or the outcome of the mechanism is revealed.

To highlight the difference between improving and refining we provide examples. In the dispatch center example in Section 2 the valuation of the item up for auction was the difference between the computed cost of including it into a delivery schedule and not including it. This fits into an improving model since if the agent did not compute a delivery solution, the item would have no value. An art auction where there is some question as to whether the artwork is a forgery or not is an example where agents refine their valuations. By researching (deliberating) the agent may be better able to determine the likelihood of whether the painting is forged. The deliberating does not change the painting, but it does change the agent's estimation of the value of the painting. The results in this paper apply to both improving and refining scenarios. We included a discussion of these two models to illustrate the generality of our approach.

If agents had unlimited deliberation resources then they would be able to optimally determine their valuations and act as though they were fully rational as assumed in most of the game theory and mechanism design literature. However, in many situations agents have restrictions on their resource usage (for example, deadlines, cost associated with making queries, etc.). We model the restrictions of agents' deliberation resources by using *cost functions*. Assume there are m possible problems that

[1] An example of an outcome is an allocation of goods in an auction.

agent i can deliberate on. Define R_i^m to be the set of vectors $(r_1, \ldots, r_m)$ such that r_k is the amount of deliberating resources used on problem k. A cost function maps resource vectors to non-negative real numbers,

$$\mathrm{cost}_i : R^m \mapsto \mathbb{R}^+ \cup \{0\}.$$

The only restrictions we place on cost functions is that they must be non-decreasing and additive. That is, given resource vectors $\overline{r}, \overline{r}' \in R^m$, $\mathrm{cost}_i(\overline{r} + \overline{r}') = \mathrm{cost}_i(\overline{r}) + \mathrm{cost}_i(\overline{r}')$ and if $\overline{r} \leq \overline{r}'$ then $\mathrm{cost}_i(\overline{r}) \leq \mathrm{cost}_i(\overline{r}')$.

4.2 Controlling Deliberation

As mentioned earlier, if agents had unlimited resources for deliberating, then they would be able to optimally determine their valuations and thus act like fully rational agents. However, restrictions on the deliberating resources of agents force them to make deliberation tradeoffs between valuation quality and cost. Anytime algorithms allow for this tradeoff to occur, but do not provide a complete solution since they do not help the agents determine how the tradeoff should be made. Instead, anytime algorithms are paired with a meta-level control procedure that aids in determining how much resources to devote to an algorithm and when to stop deliberating and act with the valuation obtained. There are two components of the procedure; the *performance profile* which describes how deliberating affects the output of the algorithm, and a process for using the information in the performance profile to make decisions. For the rest of the paper we will use the term performance profile to refer to both the descriptive and procedural aspects.

In order to make decisions about how to use deliberating resources, an agent must know what results deliberating has produced. Agents keep this information in a *state of deliberation*. Let r_j be the amount of resources spent on problem j. Let $V_j(r_j)$ be the set of valuations for problem j after r_j resources have been allocated. If agents are improving their valuations then $|V_j(r_j)| = 1$, if agents are refining their valuations then $|V_j(r_j)| \geq 1$. Let $\Delta V_j(r_j)$ be the probability distribution over $V_j(r_j)$. Define $\mathrm{path}(r_j)$ to be a list of all deliberating results obtained for problem j for $r < r_j$.

Definition 7 (State of Deliberation). *The* state of deliberation *for agent i at resource vector $\overline{r}_i = (r_1, \ldots, r_m)$ is*

$$\gamma_i(\overline{r}) = \langle(\mathrm{path}(r_1), \Delta V_1(r_1)), \ldots, (\mathrm{path}(r_m), \Delta V_m(r_m))\rangle$$

We provide an example to help in clarifying the definition. Assume that an agent can improve its valuations on two problems, 1 and 2. Assume that $V_1(1) = \{4\}, V_1(2) = \{7\}$ and $V_2(1) = \{3\}$. The state of deliberation for the agent with resource vector $\overline{r} = (2, 1)$ is $\langle((\emptyset, \{4\}, \{7\}), (\emptyset, \{3\})\rangle$ where we assume that $\{x\}$ means that with probability 1 the value was x.

A performance profile is a procedure that takes information about current results (state of deliberation) and maps it into a probability distribution over potential future results.

Definition 8 (Performance Profile). *Let $\gamma_i(\overline{r})$ be the current state of deliberation for agent i with resource vector $\overline{r} = (r_1, \ldots, r_m)$. Assume that the agent wishes to apply an additional r'_j resources to problem j. A* performance profile *for agent i and problem j is a mapping from the current deliberation state and resource vector to a probability distribution over the set of all deliberation states given the new resource vector. That is*

$$PP_i^j(\gamma_i(\overline{r}, r'_j)) = \Delta\Gamma_i(\overline{r}')$$

where $\overline{r}' = (r_1, \ldots, r_j + r'_j, \ldots, r_m)$.

The performance profile is used by an agent to determine its expected valuation for problem j after devoting an additional r'_j resources to it. Assume that an agent is in current deliberative state $\gamma_i(\overline{r})$. The expected valuation for problem j in this state is

$$E[v_i^j(r_j)] = \int_{V_j(r_j)} \omega_j pr_j(\omega_j) d\omega_j$$

where pr_j is the probability density function over $V_j(r_j)$.

By investing an additional r'_j resources to the problem, the expected valuation for problem j is

$$E[v_i^j(r_j + r'_j)] = \int_{\Gamma_i(\overline{r}')} g(\gamma_i(\overline{r}')) \int_{V_j(r_j + r'_j)} pr'_j(\omega_j)\omega_j d\omega_j d\gamma_i(\overline{r}')$$

where g is the density function provided by the performance profile over the states of deliberation, and pr' is the density function over $V_j(r_j + r'_j)$ for each state of deliberation.

If the agent is only concerned about obtaining the best valuations, and ignored the other agents in the mechanism, the agent can determine the value of deliberating for an additional r'_j steps on problem j. The *value of deliberating* is defined to be

$$VD = E[v_i^j(r_j + r'_j)] - E[v_i^j(r_j)] + \text{cost}_i(\overline{r}) - \text{cost}_i(\overline{r} + (0, \ldots, 0, r'_j, 0, \ldots, 0)).$$

If the value of deliberating is positive then the agent should deliberate for an additional step. If it is 0 or negative the agent should stop deliberating and act with the current solution.[2]

4.3 Strategic Behavior of Deliberative Agents

Deliberation control in single agent settings is difficult. Having to interact with other agents complicates the problem further. Agents must take into account the other agents' actions in determining both how to act in the mechanism and also how to use deliberation resources. We have proposed explicitly including the deliberation actions of agents into their strategies and then

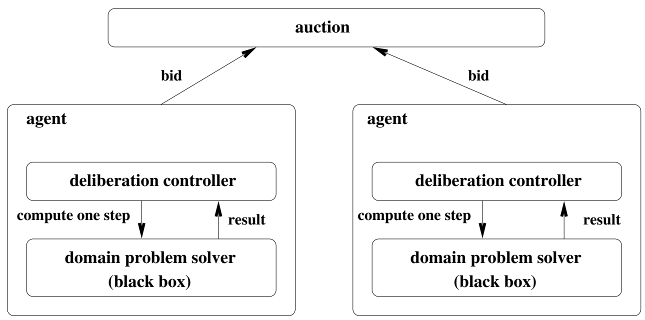

Figure 2: *An auction mechanism with two deliberative agents. In order to submit a reasonable bid, each agent needs to first (approximately) compute its valuations for the item that is up for auction The deliberation strategies of the agents may include deliberating on each others' valuation problems. We call this strategic deliberation.*

analyzing games for deliberation equilibria [7, 8]. Let D_i be the set of deliberative actions for agent i. A deliberative action $d_j \in D_i$ is the act of taking one deliberation step on problem j where a step is defined exogenously. Since it is similar to a time step, we will use the notation t to represent the $t'th$ step. The set A_i is the set of non-deliberative actions that an agent can take. The set is defined by the mechanism. For example, in a sealed-bid auction, the set A_i is simply the set of bids that the agent may submit to the auctioneer, while in an ascending auction the set A_i is the set of messages that an agent can send to the mechanism whenever the price increases (i.e. $A_i = \{\text{in}, \text{out}\}$). A strategy for a deliberative agent is a plan which specifies which actions to execute (deliberative and other) for all histories, where a history at step t, $H(t) \in \mathcal{H}(t)$ is a tuple consisting of the current deliberation state of the agent, all non-deliberative actions it has taken, and all actions it has observed other agents taking.

Definition 9 (Strategy for a Deliberative Agent). *A* strategy *for deliberative agent i is*

$$s_i = (\sigma_i^t)_{t=0}^{\infty}$$

such that

$$\sigma_i^t : \mathcal{H}(t) \mapsto D \times A.$$

In this new strategy space, we look for equilibria. We call these equilibria *Deliberation equilibria* [7]. In earlier work, we studied common auction mechanisms such as the first-price auction, Vickrey auction, ascending auction, descending auction, and generalized Vickrey auction [8]. We discovered the existence of interesting strategic behavior. In each auction mechanism studied, there existed instances where, in equilibrium, agents would use their deliberation resources to determine *other agents' valuations* of the item(s) being auctioned. We coined this phenomenon *strategic deliberation.*

[2]Performance profile deliberation control has been well studied by AI researchers [18] [2] [22] [5] [7]. There are many techniques for doing this form of deliberation control, including curve-based, table-based, and tree-based performance profiles.

Definition 10 (Strategic Deliberation). *If an agent i uses any of its deliberation resources to determine another agent's value, then agent i is strategically deliberating. That is, strategy $s_i = (\sigma_i^t)_{t=0}^{\infty}$ has strong deliberation if at time t there exists a history $H_i(t)$ such that*

$$\sigma_i^t(H_i(t)) = (d_j, x)$$

where x is any action in A_i and d_j is the action of devoting deliberation resources to a problem of agent j where $j \neq i$.

5 Mechanism Design for Deliberative Agents

In this section we tackle the problem of designing mechanisms for deliberative agents. We start with the obvious solution of defining an agent's type to be its entire set of deliberation tools and derive a revelation principle. We then argue that this results in unsatisfactory mechanisms. We propose a list of properties which we believe mechanisms for deliberative agents should have. We then study value-based mechanisms and show that it is not possible to design mechanisms which have all these properties.

5.1 Revelation Principles for Resource Bounded Agents

The type of an agent is all its private information that is relevant to the agent's decision making. In the deliberative agent model there are several interpretations of type. An agent's type may be its entire deliberating apparatus (performance profiles, algorithms, cost functions etc.) or it may only be the most recent valuations obtained by the agent. How types are defined changes the mechanisms.

We start this section by defining a type of an agent to be its entire deliberating apparatus. Let $\{\mathcal{A}_i^j\}_{j=1}^m$ be the set of algorithms available to agent i (this includes random number generators and other required technology), let $\{PP_i^j\}_{j=1}^m$ be the set of performance profiles for agent i, let $\mathrm{cost}_i(\cdot)$ be the cost function of agent i and let $\mathcal{I}$ be the set of problem instances on which agent i may have to deliberate. Define the type of agent i to be

$$\theta_i(x) = \langle \{\mathcal{A}_i^j\}_{j=1}^m, \{PP_i^j\}_{j=1}^m, \mathrm{cost}_i(\cdot), \mathrm{inst}\rangle$$

where $\mathrm{inst} \in \mathcal{I}$. That is, the type of an agent is defined by its algorithms, performance profiles, cost function and specific problem instances.[3] It is now straightforward to derive a Revelation Principle for deliberative agents.

Theorem 2. *Suppose there exists a mechanism $M = (S_1, \ldots, S_I, g())$ that implements the social choice function $f(\cdot)$ in dominant strategies. Then $f(\cdot)$ is truthfully implementable in dominant strategies.*

Theorem 3. *Suppose there exists a mechanism $M = (S_1, \ldots, S_I, g())$ that implements the social choice function $f(\cdot)$ in Bayesian Nash equilibrium. Then $f(\cdot)$ is truthfully implemented in Bayesian Nash equilibrium.*

Appendix A contains the proofs. Theorems 2 and 3 imply that mechanism design for deliberative agents can be reduced to the problem of mechanism design for rational agents. In particular, if a social choice function is implementable then it is implementable by a mechanism where agents truthfully reveal their types in a single step. However, this completely ignores the deliberative properties of the agents. First, Theorems 2 and 3 assume that agents are able to reveal all their deliberative properties including algorithms, performance profiles, cost functions, and any other technology that could be used by the agents. Secondly, it assumes that the mechanism has enough deliberating resources of its own to do all the deliberating for all the agents.

5.2 New Properties for Mechanisms

We believe that mechanisms for deliberative agents should have good deliberative properties in addition to good economic properties. In this section we propose a set of properties which we argue are desirable.

Property 1 (Non-Deliberative). *A mechanism should be* non-deliberative*. This is, the mechanism should not solve the agents' individual deliberation problems.*

[3] In example of a problem instance is the set of delivery jobs in a travelling-salesman application.

If a mechanism is non-deliberative the agents' are responsible for solving their own deliberation problems while the mechanism's primary concern is determing an outcome given the agents' strategies. We believe that in many settings it is unreasonable to assume that the mechanism is capable of both determining optimal deliberation policies for agents and computing the outcome given the policies. The problem with the mechanisms presented in Theorems 2 and 3 is that they were not non-deliberative.

Property 2 (Deliberation-Proof). *A mechanism should be* deliberation-proof. *That is, agents should have no incentive to strategically deliberarate.*

Recall that strategic deliberation is defined to be the act of actively using deliberation resources to determine the values of other agents (Definition 10). A well designed mechanism reduces the amount of strategizing required by agents to act optimally. A deliberation-proof mechanism eliminates the need for agents to gather information on competitors problems. In earlier work, Larson and Sandholm showed that the Vickrey and ascending price auctions are not deliberation-proof [8].

Property 3 (Non-Deceiving). *A mechanism should be* non-deceiving. *Assume that v_i is the true computed value of agent i. A mechanism is non-deceiving if the agent never has incentive to send a report to the mechanism such that if any agent had seen the report, their belief that agent i's value is v_i would be 0.*

A non-deceiving mechanism does not require that an agent directly reveal its true values. It just ensures that an agent does not lead all participants to believe that its true value is not possible. For example, assume that an agent had secretly computed a value v. A mechanism would be deceiving if the agent had incentive to report that its value was strictly greater than v. The mechanism would not be deceiving if the agent had incentive to report that its value was greater than some y, $y < v$.

Since we desire that mechanisms be non-deliberative, we restrict our attention to *value-based mechanisms* for the rest of the paper.

Definition 11 (Value-based Mechanism). *A value-based mechanism, $M = (S_1, \ldots, S_i, g())$, is a mechanism where each agent i's strategies are restricted so that the only allowable messages are functions of (partially) determined valuation functions and where $g()$ is a function only of the agents declared values. That is $g : V_1 \times \ldots \times V_I \mapsto \mathcal{O}$. Agents do not reveal information about performance profiles, algorithms, cost functions etc.*

Value-based mechanisms are non-deliberative. The mechanism is not given any of the tools required for it to actively deliberate on agents' problems. Examples of value-based mechanisms include sealed-bid auctions where agents submit numerical bids, and ascending and posted-price auctions where agents answer yes or no to the query of whether they would be willing to buy an item at a specified price.

Mechanisms can be broadly classified into two groups, sensitive and non-sensitive.

Definition 12 (Sensitive). *A mechanism is* sensitive *to agents' strategies if for each agent i there exist strategies s'_i, s''_i, $s'_i \neq s''_i$ such that for strategy profiles $s' = (s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_I)$, $s'' = (s_1 \ldots, s_{i-1}, s''_i, s_{i+1}, \ldots, s_I)$, $g(s') \neq g(s'')$.*

Non-sensitive mechanisms are ones in which agents' strategies do not influence the outcome. Dictatorial mechanisms which always choose the preferred outcome of a specific agent and totally random mechanisms which choose outcomes in a completely random fashion are examples of non-sensitive mechanisms.

Theorem 4. *Assume mechanism $M = (S_1, \ldots, S_n, g(\cdot))$ is value-based and non-sensitive. Then M is non-deliberative, non-deceiving and deliberation-proof.*

The proof is found in appendix B. While this is a positive existence result (there do exist mechanisms with the set of desirable properties) it is not particularly useful. Most interesting mechanisms (efficient mechanisms, for example) are sensitive. In the rest of the paper we will focus on sensitive mechanisms.

5.3 Sensitive Mechanisms

We first look at direct mechanisms to see how deliberative agents behave.

Theorem 5. *There exists no value-based sensitive direct mechanism that is deliberation-proof across all problem instances. (An instance is defined by the agents performance profiles, cost functions, and current deliberation problems.)*

The proof is found in Appendix C. While direct mechanisms are not deliberation-proof in general, they do allow us to study why strategic deliberation occurs. The first reason is that by knowing the valuations of others, an agent may be able to tailor the announcements in such a way so as to increase their chances of being included in the final allocation. However, by designing mechanisms which are incentive-compatible this issue disappears. A more insidious problem is the following. Agents may have asymmetric cost functions. It may be more difficult to deliberate on some problems as opposed to others. If an agent finds itself in a situation where it is easier to deliberate on a competitor's problem compared to its own problems, then it may strategically deliberate in order to determine if it is worth while to actually do (or continue doing) its own deliberation. For example, in a Vickrey auction, if agent 1 can learn for a low cost that agent 2 has achieved a very high valuation after deliberating then it may be in agent 1's best interest to not waste additional deliberation resources on determining its own problems.

To get rid of strategic computing the deliberative agents must be provided enough information by the mechanism so that they can determine whether to devote resources to their own problems or not. Many indirect mechanisms reveal information. This information can be used by the agents to help determine which are the best deliberative and non-deliberative actions to take. For example, in a single item ascending auction, as the price rises the mechanism may reveal to all agents the number of agents remaining in the auction. The agents can use this information to deduce valuation information about the remaining agents.

To explicitly model the information that is revealed to the agents by the mechanism we introduce a *feedback game*. A feedback game (M, F) is the extensive form game induced by mechanism M (this includes the deliberation actions of the agents) coupled with a *feedback function*. At each step of the game, the feedback function maps all messages that are sent to the mechanism at that step to the information that would have been revealed to the agents through the mechanism. To illustrate, we provide two examples. Let M be a direct auction. At each step agents have the choice of taking a deliberation action, and at some step t' agents submit bids to the auction. The feedback function for this example is

$$F(t, \omega) = \begin{cases} \emptyset & \text{if } t \neq t' \\ (x, p) & \text{if } t = t' \end{cases}$$

where ω is the vector of agents' messages. In an ascending auction more information is revealed by the mechanism. The feedback function is

$$F(t, \omega, p) = \{i | i \in I \text{ and } i \text{ is still in the auction at price } p\}.$$

Introducing an explicit feedback function does not change the original mechanism. However, it provides a tool to the mechanism designer for reasoning about what information is available to agents and how the agents are doing belief revision given the information reported by the feedback function.

Lemma 1. *Given any mechanism M it is possible to construct a feedback function such that the equilibria in the feedback game (M, F) are the same equilibria in the original mechanism M.*

Using the feedback function as a tool to help in the analysis, we are able to show our final impossibility result. In general, sensitive value-based mechanisms do not satisfy the proposed desirable properties.

Theorem 6. *There exists no sensitive value-based mechanism that is*

- *non-deliberative,*
- *deliberation-proof, and*
- *non-deceiving*

across all problem instances. (An instance is defined by agents' performance profiles, cost functions).

6 Related Research

Both the the game theory and the computer science research communities are interested in mechanism design issues where computation and information gathering are issues. In this section we review the literature which is most closely related to the work presented in this paper.

From the computer science research community there has been work on both bounded–rational bidding agents and mechanisms. Sandholm noted that under a model of costly computing, the dominant strategy property of Vickrey auctions fails to hold, even with simple agent models [20]. Instead, an agent's best computing action can depend on the other agents. In recent work, auction settings where agents have hard valuation problems have been studied [15]. Auction design is presented as a way to simplify the meta–deliberation problems of the agents', with the goal of providing incentives for the "right" agents to compute for the "right" amount of time. A costly computing model, with simple computing control where agents compute to refine their valuations, is used. However, situations where agents may compute on each others' problems in order to refine their bids are not considered. There has also been recent work on computationally limited mechanisms. In particular, research has focused on the generalized Vickrey auction and investigates ways of introducing approximate algorithms to compute outcomes without loosing the incentive compatibility property [12, 6, 9]. These methods still require that the bidding agents compute and submit their valuations.

In the economics and game theory literature there has been some recent work on information acquisition and mechanism design. This work has mainly focused on studying the incentives to acquire information in different auction mechanisms. Most work assumes that an agent can only gather information about its own valuation. Perisco compares first-price and second-price auctions and shows that if agents have affiliated valuations then agents choose to acquire more information about their own valuation in a first price auction [16]. Compte and Jehiel compare the ascending price auction with the second price auction and show that there exist situations where an agent will pay to acquire information in an ascending auction but not in the second price auction [3, 4]. Rezende also studies an ascending price auction where agents are allowed to pay to acquire information about their own valuation. He shows that in such a setting, in equilibrium a bidder's best response function has a simple characterization which is independent of other agents' strategies. Bergemann and Välimäki study general mechanisms [1]. They are interested in mechanisms where agents acquire the efficient amount of information. They show that in the private value model then VCG mechanism is efficient in the sense that agents have incentive to acquire enough information so as to maximize the social welfare. However, in a purely common value setting agents either over-acquire or under-acquire information.

Rasmusen's work is the most similar to ours [17]. It assumes that agents do not know their valuations but must invest to learn them and are also able to invest in competitors value problems. Like us, he shows that in a second price auction, an agent may base its decision on whether to learn its true valuation on another agent's valuation, but his focus is on understanding behavior such as sniping that is commonly seen in different online auctions like eBay.

7 Conclusion

In many settings, agents participating in mechanisms do not know their preferences (valuations) *a priori*. Instead, they must actively determine them through deliberation (e.g., information processing or information gathering). Agents are faced not only with the problem of deciding how to reveal their preferences to the mechanism but also how to deliberate in order to determine their preferences. For such settings, we have introduced the *deliberation equilibrium* as the game-theoretic solution concept where the agents' deliberation actions are modeled as part of their strategies.

In this paper, we laid out mechanism design principles for such deliberative agents. We first showed that the revelation principle applies to such settings in a trivial sense by having the mechanism carry out all the deliberation for the agents. This is impractical, and we propose that mechanisms should be *non-deliberative*: the mechanism should not be solving the deliberation problems for the agents. Second, mechanisms should be *deliberation-proof*: agents should not deliberate on others' valuations in equilibrium. Third, the mechanism should be *non-deceiving*: agents do not strategically misrepresent. Finally, the mechanism should be *sensitive*: the agents' actions should affect the outcome. We showed that no direct-revelation mechanism satisfies these four intuitively desirable weak properties. This is the first impossibility result in mechanism design for deliberative agents. Moving beyond direct-revelation mechanisms, we showed that no *value-based* mechanism (that is, mechanism where the agents are only asked to report valuations - either partially or fully determined ones) satisfies these four properties.

This result is negative. It states that either we must have mechanisms which do the deliberating for the agents, or complex strategic (and costly) counterspeculation can occur in equilibrium. However, there is some hope. It may be possible to weaken one of the properties slightly, while still achieving the others. For example, it may be possible to design multi-stage mechanisms that are not value based; the mechanism could help each agent decide when to hold off on deliberating during the auction (and when to deliberate on one's own valuation on different bundles of items in a combinatorial auction). In another direction, by relaxing strategic deliberation and compensating agents appropriately, it may be possible to design mechanisms where agents who can deliberate cheaply and efficiently deliberate for all agents. These are areas which we plan to pursue in future work.

References

[1] Dirk Bergemann and Juuso Välimäki. Information acquisition and efficient mechanism design. *Econometrica*, 70:1007–1034, 2002.

[2] Mark Boddy and Thomas Dean. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67:245–285, 1994.

[3] Olivier Compte and Philippe Jehiel. On the virtues of the ascending price auction: New insights in the private value setting. working paper, December 2000.

[4] Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or dynamic formats?, 2001. Working Paper.

[5] Eric Hansen and Shlomo Zilberstein. Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence*, 126:139–157, 2001.

[6] Noa Kfir-Dahav, Dov Monderer, and Moshe Tennenholtz. Mechanism design for resource bounded agents. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 309–315, Boston, MA, 2000.

[7] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001. An early version appeared in AAAI 2000, pp. 48–55, Austin, TX, 2000.

[8] Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 169–182, Sienna, Italy, July 2001.

[9] Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002. Early version appeared in ACMEC-99.

[10] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[11] S Minton, M D Johnston, A B Philips, and P Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58(1–3):161–205, 1992.

[12] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 242–252, Minneapolis, MN, 2000.

[13] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001. Early version in STOC-99.

[14] Christos Papadimitriou. Algorithms, games and the Internet. In *STOC*, pages 749–753, 2001.

[15] David C Parkes. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

[16] Nicola Perisco. Information acquisition in auctions. *Econometrica*, 68(1):135–148, January 2000.

[17] Eric Rasmusen. Strategic implications of uncertainty over one's own private value in auctions. working paper, January 2003.

[18] Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361–395, 1991.

[19] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., July 1993.

[20] Tuomas Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000. An early version appeared ICMAS'96, pages 299–306, 1996.

[21] Tuomas Sandholm and Victor R Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94(1):99–137, 1997. Early version appeared in IJCAI'95, pages 662–669, 1995.

[22] Shlomo Zilberstein and Stuart Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82(1–2):181–213, 1996.

A Proof of Theorem 2

In this section we prove Theorem 2. The proof for Theorem 3 is similar.

Proof. The proof follows an argument similar to that of the original Revelation Principle. Assume agent i has m problems it can deliberate on. Let $\{A_i^j\}_{j=1}^m$ be the set of algorithms and all required tools for running the algorithms for agent i, $\{PP_i^j\}_{j=1}^m$ be the set of performance profiles, $\mathrm{cost}_i(\cdot)$ be the cost function of agent i, and let $x \in \mathcal{I}$ be a specific problem instance.

Recall the definition of the type of agent i. The type of the agent,

$$\theta_i(x) = \langle \{\mathcal{A}_i^j\}_{j=1}^m, \{PP_i^j\}_{j=1}^m, \mathrm{cost}_i(\cdot), x \rangle$$

Suppose an indirect mechanism $M = (S_1, \ldots, S_I, g(\cdot))$ implements social choice function $f(\cdot)$ in dominant strategies. Then, there exists a strategy profile $s^* = (s_1^*, \ldots, s_I^*)$ such that

$$u_i(g(s_i^*(\theta_i(x)), s_{-i}(\theta_{-i}))) - \mathrm{cost}_i(s_i^*(\theta_i(x)) \geq u_i(g(s_i'(\theta_i(x)), s_{-i}(\theta_{-i}))) - \mathrm{cost}_i(s_i'(\theta_i(x)))$$

for all s_i' and s_{-i}.

Alter the mechanism in the following way. Introduce a mediator who says to each agent i: "You tell me your type, and when you say your type is $\theta_i(x)$, I will play $s_i^*(\theta_i(x))$ for you. This includes deliberating on the problems as specified by s_i^*. I will charge an amount $\mathrm{cost}_i(s_i^*(\theta_i(x))$ for this service." If $s_i^*(\theta_i(x))$ is the optimal strategy for agent i for each $\theta_i(x) \in \Theta_i$ in the initial mechanism M for any strategy chosen by the other agents, then agent i will find telling the truth to be a dominant strategy in this new scheme. This means we have a mechanism which truthfully implements $f()$. □

B Proof for Theorem 4

Proof. Suppose a mechanism $M = (S_1, \ldots, S_I, g(\cdot))$ is value-based and non-sensitive. Clearly, the mechanism is non-deliberative. Since it is value-based, the mechanism is not provided by the set of tools it would require to solve the agents' deliberation problems. Let s_i' and s_i'' be two strategies for agent i such that all deliberation actions of specified by the two strategies are the same, but s_i' is non-deceiving and s_i'' is deceiving. Therefore, $\mathrm{cost}_i(s_i') = \mathrm{cost}_i(s_i'')$. Since the mechanism is non-sensitive, $g(s_i', s_{-i}) = g(s_i'', s_{-i})$ and so

$$u_i(g((s_i', s_{-i})) - \mathrm{cost}_i(s_i') = u_i(g(s_i'', s_{-i})) - \mathrm{cost}_i(s_i'')$$

and so following a non-deceiving strategy is weakly dominant for agent i.

A similar argument holds when showing that the mechanism is deliberation-proof. Let s_i be a strategy such that agent i deliberated only on its own problems. Further assume that it devotes some amount r_i on deliberating. Let s_i' be any strategy where agent i deliberates r_i on its own problems and r_{-i} on other agents' problems. Since the mechanism is not sensitive,

$$g(s_i, s_{-i}) = g(s_i', s_{-i})$$

for all s_{-i}. Therefore,

$$u_i(g(s_i, s_{-i})) - \mathrm{cost}_i(s_i) \geq u_i(g(s_i', s_{-i})) - \mathrm{cost}_i(s_i')$$

since the cost function is additive and nondecreasing. This holds for any r_i. Therefore, agent i is always better off by not deliberating on other agents' problems. □

C Proof of Theorem 5

Proof. To prove the theorem, we need only show that there exist instances, defined by the performance profiles, where in equilibrium agents will strategically deliberate. We will restrict ourselves to mechanisms which are individually rational. If a mechanism does not satisfy participation constraints then agents have no incentive to deliberate in the first place. Let $M = (S_1, \ldots, S_n, g())$ be any incentive compatible mechanism. Since our agents have quasilinear utilities, it must be the case the $g() = (k(), t())$ where $k : S_1 \times \ldots \times S_n \mapsto X$ and $t : S_1 \times \ldots S_n \mapsto \mathbb{R}^n$.

Assume that $|I| = 2$, that is there are two agents. The agents have access to the same algorithms. Each agent i has performance profiles $PP_i^{\{i,j\}}$. The performance profiles are common knowledge. Each agent has a cost function cost_i where

$$\mathrm{cost}_1(\mathbf{r}) = \epsilon \sum_{i=1}^{2} r_i$$

for some small $\epsilon > 0$, and

$$\mathrm{cost}_2(\mathbf{r}) = r_1 + Kr_2$$

for some constant $K > 1$.

Since we are free to choose the problem instance, we define the performance profiles as follows. The performance profile for agent 1 is

$$PP_1(\mathbf{r}) = \begin{cases} 0 & \text{if } \sum r_i = 0 \\ v_1^h \text{ with probability } p & \forall \mathbf{r} \text{ such that } r_1 > 0 \\ v_1^l \text{ with probability } 1-p & \forall \mathbf{r} \text{ such that } r_1 > 0 \end{cases}$$

The performance profile for agent 2 is

$$PP_2(\mathbf{r}) = \begin{cases} 0 & \text{if } \sum r_i = 0 \\ v_2^h \text{ with probability } q & \forall \mathbf{r} \text{ such that } r_2 > 0 \\ v_2^l \text{ with probability } 1-q & \forall \mathbf{r} \text{ such that } r_2 > 0 \end{cases}$$

where v_1^h, v_1^l, v_2^h and v_2^l are chosen such that $k(v_1^h, v_2^*) = 1$, $k(v_1^*, v_2^l) = 1$ and $k(v_1^l, v_2^h) = 2$. The transfers $t_i(v_1, v_2)$ are determined so that the mechanism is incentive compatible. In particular, the transfer function of agent i can not be a function of its own declaration. That is, $t_i = t_i(v_j), j \neq i$.

For small ϵ agent 1 has a dominant strategy which is to deliberate for one step on its own problem. For example if agent 2 decides not to deliberate then clearly agent 1 is best off deliberating on its own problem since it does not matter what value agent 2 could have obtained. If agent 2 deliberates on its own problem, then agent 1 is still best off deliberating on its own, since

$$-\epsilon + p(v_1^h + qt_i(v_2^h) + (1-q)t_1(v_2^l)) + (1-p)(1-q)(v_1^l + t_1(v_2^l)) \geq 0$$

where 0 is the utility if agent 1 does not deliberate and

$$-\epsilon + p(v_1^h + qt_i(v_2^h) + (1-q)t_1(v_2^l)) + (1-p)(1-q)(v_1^l + t_1(v_2^l)) \geq -\epsilon q + (1-q)(-2\epsilon + pv_1^h + (1-p)v_1^l + t_1(v_2^l))$$

where the left hand side is the utility if agent 1 deliberates on agent 2's problem and then deliberates on its own problem only if agent 2's valuation is v_2^l. In a similar way it is easy to show that for small ϵ agent 1 is best off deliberating on its own problem when agent 2 deliberates on agent 1's problem first.

Given that agent 1 will deliberate on its own problem, agent 2 must determine its best strategy. It can determine its own true valuation for a cost of $K > 1$ but it can also determine the valuation of agent 1 at a cost of 1. Agent 2 will first determine whether agent 1 has high or low type under the following conditions;

$$-p + (1-p)(q(v_2^h + t_2(v_1^l)) - (1+K)) \geq 0,$$

that is agent 2 is better off deliberating on agent 1's problem than not deliberating at all, and

$$-p + (1-p)(q(v_2^h + t_2(v_1^l)) - (1+K)) \geq -K + (1-p)q(v_2^h + t_2(v_1^l)),$$

that is agent 2 is better off deliberating on agent 1's problem than deliberating on its own problem. This means that agent 2 will strategically deliberate if K is set such that

$$\frac{p + (1-p)q}{1-(1-p)q} \leq K \leq q(v_2^h + t_2(v_1^l)) - \frac{1+p}{1-p}$$

where we are free to choose p, q, v_2^h, v_1^l. $\square$

D Proof of Theorem 6

The feedback function is a tool used in the main proof to formally model the information exchanged between agents. Since we aim to prove a theorem over all mechanisms we need to first show that introducing a feedback function is possible.
Proof of Lemma 1

Proof. Let $M = (S_1, \ldots, S_n, g())$ be any mechanism and let G^M be the extensive form game that agents play when participating in mechanism M. For each stage t in the game let $\omega(t)$ represent the messages sent to the mechanism by the agents. Let $\text{info}(\omega(t))$ be the information revealed at stage t in the game G^M after agents send $\omega(t)$. Define the feedback function $F()$ as follows

$$F(t, \omega(t)) = \text{info}(\omega(\text{t})).$$

By introducing the feedback function, nothing in the game G^M has been changed. Therefore, the feedback game (M, F) is equivalent to the game G^M. ☐

In Theorem 6 we study agents' strategic behavior in feedback mechanisms. To do so, we do a perfect Bayesian equilibrium (PBE) analysis. Before starting the proof, we review the definition of a PBE. We assume that the reader is familiar with extensive form games [10].

Definition 13 (System of Beliefs). *A system of beliefs μ in an extensive form game Λ is a specification of a probability $\mu(x) \in [0, 1]$ for each decision node x in Λ such that*

$$\sum_{x \in H} \mu(x) = 1$$

for all information sets H.

Definition 14 (Sequentially Rational). *A strategy profile $s = (s_1, \ldots, s_I)$ is* sequentially rational *given belief system μ if whenever it is agent i's turn to move*

$$E[u_i|H, \mu, s_i, s_{-i}] \geq E[u_i|H, \mu, s_i', s_{-i}]$$

for all s_i' and for all information sets H.

Definition 15 (Perfect Bayesian Equilibrium). *A profile of strategies and system of beliefs (s, μ) is a* perfect Bayesian equilibrium *if it has the following properties:*

- *The strategy profile s is sequentially rational given belief system μ.*
- *The system of beliefs μ is derived from strategy profile s using Baye's Rule when ever possible.*

Proof. Theorem 6

Let mechanism M be an incentive compatible (non-deceiving) direct mechanism which implements social choice function $f(\cdot)$. Mechanism M is not deliberation-proof (Theorem 5). Therefore, it is possible to construct problem instances such that in equilibrium, agents will strategically deliberate. Using the technique in the proof for Theorem 5 construct performance profiles PP_1, PP_2 (such that in one deliberation step agents learn their true valuation) and cost functions, cost_1 cost_2, for 2 agents such that in the mechanism M, one agent (agent 2) has incentive to strategically deliberate while the other (agent 1) has a dominant strategy to deliberate only on its own problem. We will need an additional assumption on the equilibrium strategies and transfer functions. If an agent decides not to deliberate and thus have valuation 0, then we assume that its strategy indicates to the mechanism that $v_i = 0$. Second, we assume that $t_i(0) \leq t_i(v)\ \forall v \neq 0$. These assumptions do not change the equilibrium behavior of the agents since if an agent, for all intents and purposes, resigns from the game, declaring its valuation to be anything else but the default valuation of 0 is weakly dominant.

Since strategic deliberating occurs in equilibrium in the direct mechanism M, there exists an information set H_2 such that

$$E[u_2|H_2, \mu, ((a_1, H_2)s_2, s_1] \geq E[u_2|H_2, \mu, (a_2, H_2)s_2, s_1]$$

or

$$E[u_2|H, \mu, ((a_1, H_2)s_2, s_1] \geq E[u_2|H_2, \mu, (\emptyset, H_2)s_2, s_1]$$

where $(a, H)s$ denotes the strategy which specifies taking deliberation action a at information set H. In particular in the direct mechanism this information set occurs at the start of deliberating for agent 2.

Let $M' = (S_1, \ldots, S_n, k(), t_1(), \ldots, t_n())$ be an indirect mechanism which also implements the social choice function $f(\cdot)$. Let s^* be the equilibrium strategy profile for mechanism M'. Since the mechanism implements the same social function as M the outcomes of the two mechanisms must be the same. Create an appropriate feedback function $F(\cdot)$ to produce feedback game $(M', F(\cdot))$ (Lemma 1).

Let v represent agent j's actual valuation that it could achieve if it deliberated on its own problem (recall that in this example instance, agents need only deliberate for one step on a problem to determine the valuation. With no deliberation their valuation is 0). At each stage t of the game induced by the mechanism M' define $\overline{V_i}(t)$ to be a partition that agent i makes of agent j's possible deliberatable valuations. In particular, $\overline{V_i}(t) = \{V, V'\}$ where if $v \in V$ then agent i has incentive to deliberate on its own problem and if $v \in V'$ then agent i is best off stopping all deliberation on any problem. Given the construction of the problem instance, there exists stages such that $V' \neq \emptyset$.

The feedback function F can exhibit different properties at each stage t of the game.

Definition 16 (Separating). *Let $\overline{V_i}(t) = \{V, V'\}$ be a partition for agent i at stage t. A feedback function F is* separating at stage t *if for any messages $m(v_j)$, $m(v'_j)$ sent by agent $j \neq i$ such that $v_j \in V$ and $v'_j \in V'$,*

$$F(t, (m(v_i), m(v_j))) \neq F(t, (m(v_i), m(v'_j))).$$

Definition 17 (Pooling). *A feedback function $F(\cdot)$ is* pooling at stage t *if it is not separating at stage t.*

Since agent 1 has a dominant strategy we assume that it determines its own valuation and then sends signals to the mechanism. Assume that at stage $t-1$ agent 2 has partition $\overline{V}_i(t-1) = \{V, V'\}$ where $V' \neq \emptyset$. Upon observing in stage t $F(t, s_1(v_1))$ agent 2 updates its beliefs about what set the actual valuation of agent 1 is in.

Assume the feedback function F is pooling at stage t. Let $m(v^*)$ be the message that agent 1 sends to the mechanism in equilibrium and let $a^*(F(t, m(v^*))$ be the deliberation action that agent 2 takes upon observing $F(t, (v^*))$. Since F is pooling at stage t, agent 2 must base its beliefs on whether $v_1 \in V$ or $v_1 \in V'$ solely on the probability given by the performance profiles. However, this is the same beliefs that agent 2 would have in a situation where there was a direct mechanism. Therefore, agent 2 is best off strategically deliberating (since the performance profiles were chosen so as to have strategic deliberation in the direct mechanism) or doing nothing (not even sending a message) and waiting for the next stage of the game.[4]

Assume that at stage t the feedback function is separating. Again, let $m(v_1^*)$ be the signal that agent 1 sends to the feedback mechanism in equilibrium and let $a^*(F(t, m(v_1^*)))$ be the action that agent 2 takes in equilibrium upon observing $F(t, m(v_1^*))$. In perfect Bayesian equilibrium, the beliefs on the equilibrium path must be correctly derived from the equilibrium strategies using Bayes Rule. This implies that upon seeing $F(t, m(v_1^h))$ agent 2 must assign probability one to agent 1 having high type. Similarly it assigns probability one to agent 1 having low type if it observes $F(t, m(v_1^l))$. Upon observing $F(t, m(v_1^h))$ agent 2 believes that agent 1 has high type, it also believes that the mechanism will choose an outcome preferable to agent 1 (since the mechanism is implementing social choice function $f()$ which would select an outcome favorable to agent 1 when it has high type). Therefore, agent 2 is best off not computing at all since otherwise it would just incur a cost without being able to recoup the cost from getting a preferred outcome. Therefore, $a^*(F(t, m(v_1^h))$ is to do nothing, agent 1 gets its preferred outcome and pays $t_1(0)$. That is, the utility of agent 1 is

$$v_1^h + t_1(s_2(0)) - \epsilon = v_1^h + t_1(0) - \epsilon$$

If agent 2 witnessed $F(t, m(v_1^l))$ then it believes that agent 1 is low type and it has incentive to deliberate. The expected utility for agent 1 in this situation is

$$(1-q)(v_1^l + t_1(s_2(v_2^l))) - \epsilon$$

However, if agent 1 has a low type but sends a message to the mechanism $m(v_1^h)$ then agent 2 would not deliberate and the utility of agent 1 would be

$$v_1^l + t_1(s_2(0)) - \epsilon \geq (1-q)(v_1^l + t_1(s(v_2^l))) - \epsilon$$

Therefore, if the feedback function is separating, agent 1 has incentive to misreport its deliberated value to the feedback mechanism. Therefore, the mechanism is deceiving. □

[4]To get rid of situations where agents wait forever, one can introduce a liveness condition which will force an agent to eventually take some action.