Regret-based Incremental Partial Revelation Mechanisms

Nathanaël Hyafil
Department of Computer Science
University of Toronto
Toronto, ON, M5S3H5, CANADA
{nhyafil}@cs.toronto.edu

Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, ON, M5S3H5, CANADA
{cebly}@cs.toronto.edu

Abstract

Classic direct mechanisms suffer from the drawback of requiring full type (or utility function) revelation from participating agents. In complex settings with multi-attribute utility, assessing utility functions can be very difficult, a problem addressed by recent work on preference elicitation. In this work we propose a framework for incremental, partial revelation mechanisms and study the use of *minimax regret* as an optimization criterion for allocation determination with type uncertainty. We examine the incentive properties of incremental mechanisms when minimax regret is used to determine allocations with no additional elicitation of payment information, and when additional payment information is obtained. We argue that elicitation effort can be focused simultaneously on reducing allocation and payment uncertainty.

Introduction

Mechanism design [13] studies the design of protocols through which self-interested agents can interact to achieve some (e.g., socially desirable) objective. The *revelation principle* states that mechanisms can be restricted to those in which agents reveal their type, that is, their utility function over outcomes. However, as agents increasingly interact in powerful computational settings, outcome spaces are becoming more complex, combinatorial auctions (CAs) being a now standard example [6]. Thus eliciting complete type information is unlikely to be successful.

This limitation of direct revelation mechanisms is starting to be addressed. Recent research has examined methods involving limited or incremental elicitation of types to circumvent some of these difficulties (see, e.g., [5; 1; 16]), much of this in the context of (single-good or combinatorial) auctions. We continue along these lines by proposing *regret-based elicitation* in partial revelation mechanisms. Specifically, we use *minimax regret* to define the quality of an outcome in the presence of type uncertainty. Techniques for elicitation based on regret can quickly determine relevant type information for optimal choices, or provide bounds on error if optimality is not possible.

While we draw on single-agent regret-based elicitation frameworks [2; 4; 3], our key contributions are the investigation of incentive properties when we adopt these models for the design of incremental, partial-revelation mechanisms. We first elaborate a general model for incremental mechanisms that admit partial type revelation. Focusing on efficiency in quasi-linear environments and *ex post* equilibria, we then define a simple payment scheme that can be used with any partial type information and describe the incentive properties that result from a direct application of regret-based elicitation to multi-agent settings. We then improve incentives by adding a second "payment elicitation" phase, itself a common mechanism design approach [9; 12]), that exploits the notion of regret and allows *a priori* bounds on manipulability to be provided. Finally, we argue that if one is going to allow for both allocation and payment uncertainty, one should not break elicitation into the two phases. Instead, we define a "global regret" over both measures simultaneously and describe a regret-based elicitation process that quickly reduces *both* loss in efficiency and manipulability. A key feature of our framework is that *a priori* bounds on manipulability can be provided, allowing a tradeoff between the quality and "incentive properties" of the mechanism and the amount of elicitation required.

Related Work

Much work has been devoted to the partial elicitation of preferences. Apart from a considerable body of work on ascending-price mechanisms, recent work has focused on more direct elicitation of types. Although theoretical results show savings in information revelation are impossible in the worst-case [15], a number of approaches have been empirically successful by exploiting the (inherent or assumed) structure of the valuation space, particularly in CAs [5; 9; 18; 11]. In our work, we also exploit a known, compact representation of valuations, but the type of structure we consider is completely general (and applies beyond CAs).

When designing incremental mechanisms, it is not only necessary to elicit enough information to determine an optimal allocation, but also to determine suitable payments to ensure that appropriate incentive properties are met. A famous result by Green and Laffont (1979) shows that, in general, choosing the efficient allocation in dominant strategy (or ex-post) equilibrium requires using the Groves payment scheme. To ensure individual rationality, it is sufficient to use VCG payments (a special case of Groves). Hudson and Sandholm [9], for instance, consider a two-phase approach

for CAs in which elicitation is first directed toward determining an efficient allocation, and is then focused on optimal allocations in the $(n-1)$-agent sub-economies to determine VCG payments, thus ensuring truth telling is an *ex post* equilibrium in the sequential mechanism.[1] In [11], the two phases are run simultaneously by using universal demand queries, but the elicitation of the allocation and the payments is still done independently. In [17], incremental, partial revelation mechanisms are designed by converting a one-shot dominant strategy mechanism into a multistage one.

While we focus on incremental analogs of VCG mechanisms with partial type revelation, Nisan and Ronen [14] discuss the *computational approximation* of VCG mechanisms. They show that, for two important classes of problems, any "reasonable" approximation scheme for determining allocations will destroy truth-telling of the corresponding VCG scheme. This form of approximation is quite different from ours in that we assume optimization is possible, but rely on regret to bound the amount of information that is actually required to do the (approximate) optimization.

Fadel and Segal [7] tackle a related problem to ours, examining the communication cost incurred by insisting enough information be revealed to ensure (ex post) incentive properties are respected by a mechanism designed to implement some social choice function. They show that ex post incentive compatibility can incur significant cost (over the information required to implement the function). Our focus is somewhat different, allowing the (bounded) relaxation of ex post incentive compatibility in order to alleviate the burden of revelation (in practice).

Mechanism Design: Background

We adopt a standard quasi-linear environment with n agents in which the aim is to choose an *outcome* or *allocation* $\mathbf{x}$ from the set $\mathbf{X}$. Each agent $i \leq n$ has *type* t_i drawn from set T_i, and valuation function $v_i : \mathbf{X} \times T_i \rightarrow \mathbb{R}$, with $v_i(\mathbf{x}; t_i)$ denoting the value of allocation $\mathbf{x}$ if i has type t_i. In many cases, we can view t_i as encoding i's utility function over $\mathbf{X}$. Let $T = \prod_i T_i$ be the set of full type vectors. The *social welfare* of $\mathbf{x}$ given $t \in T$ is $SW(\mathbf{x}; t) = \sum_i v_i(\mathbf{x}; t_i)$. Let t_{-i} denote a type vector over all agents but i, and $SW_{-i}(\mathbf{x}; t) = \sum_{j \neq i} v_i(\mathbf{x}; t_i)$.

A *mechanism* consists of a set of actions $A = \prod_i A_i$, an allocation function $x^* : A \rightarrow \mathbf{X}$ and n payment functions $p_i : A \rightarrow \mathbb{R}$. Intuitively, the mechanism offers the action set A_i to i, and chooses an allocation based on the actions taken by each agent. We assume *quasi-linear utility*; that is, an agent i's utility for an allocation $\mathbf{x}$ and payment ρ_i is $u_i(\mathbf{x}, \rho_i, t_i) = v_i(\mathbf{x}; t_i) - \rho_i$. It is clear that mechanism m induces a (Bayesian) game for the participating players, assuming that each agent possesses probabilistic beliefs about the types of the others. In this game, each agent i adopts a strategy $\pi_i : T_i \rightarrow A_i$ associating an action with its type.

[1] In some elicitation strategies, once an efficient allocation is determined, enough information has been elicited to determine VCG payments [5]. Unfortunately, these rigid schemes tend to elicit considerably more information than some others, which typically perform better [9].

The goal of mechanism design is to design m to implement some social choice function $f : T \rightarrow \mathbf{X}$. For instance, f may be social welfare maximization (i.e., $f(t) = \arg\max SW(\mathbf{x}; t)$). In this work, we focus on social welfare maximization or *efficient allocation*. Implementation then depends on the equilibrium concept used; specifically, if m induces strategies π_i for each agent in equilibrium such that $x^*(\pi(t)) = f(t)$ for all $t \in T$, we say that m *implements* f. Standard equilibrium concepts lead to dominant strategy, ex post, and Bayes-Nash implementation. We will be concerned largely with implementation in *ex-post equilibrium*: a collection of strategies π such that π_i is optimal for i even when the types of all other agents are known to i (assuming they adopt their strategies in π).

The *revelation principle* allows one to focus attention on direct, incentive compatible mechanisms in which $A_i = T_i$ and each agent will reveal its type truthfully in equilibrium. A direct mechanism is *incentive compatible* if (in equilibrium) each agent reports its type truthfully. A mechanism is *ex-post individually rational* if no agent i is better off not participating in the mechanism (in equilibrium) even when the types of others are known to i. The VCG scheme (elaborated below) is a famous class of mechanisms that induces truth telling in dominant strategies in quasi-linear settings with social welfare maximization as the goal.

Incremental Partial Revelation Mechanisms

Because of the difficulties of full type revelation discussed above, we focus on the incremental revelation of partial types. We define a *partial type* $\theta_i \subseteq T_i$ for agent i to be any subset of i's types. A partial type vector θ includes a partial type for each agent. We now elaborate on the definition of a mechanism to draw out the structure of the iterative querying process involved in incremental elicitation.

Let Q_i be the set of queries the mechanism can pose to i, and let $R_i(q_i)$ be the responses i can offer to $q_i \in Q_i$. We interpret each query as asking i about its type; thus each response $r \in R_i(q_i)$ is equated with a partial type $\theta_i(r) \subset T_i$. For example, standard direct mechanisms would ask i "What is your true type?" with response set T_i. A simpler query "Is your valuation for outcome $\mathbf{x}$ greater than v?" admits two responses (yes and no) corresponding to the obvious subsets of T_i. Standard queries (e.g., value, rank, demand queries) can all be represented in this way. Let $Q = \cup Q_i$.

A *nonterminal history* is any finite sequence of queries and responses (including the empty sequence), and a *terminal history* is any nonterminal history followed by an outcome $\mathbf{x} \in \mathbf{X}$. Let $\mathcal{H} = \mathcal{H}_t \cup \mathcal{H}_n$ be the set of (terminal or nonterminal) *histories*, and for any $h \in \mathcal{H}$, let h_i denote the restriction of h to queries/responses for agent i. For any h, let $h^{\leq k}$ denote the initial k-step history. We use h^k to denote the kth query-response pair or outcome in sequence h and $a(h^k)$ refers to the "action" (i.e., query asked or outcome chosen) at stage k of this history. An *incremental mechanism* $M = \langle m, (p_i)_{i \leq n} \rangle$ consists of: (a) a mapping $m : \mathcal{H}_n \rightarrow Q \cup \mathbf{X}$, that for each nonterminal history chooses a query for some agent or selects an outcome; and (b) a collection of payment functions $p_i : \mathcal{H}_t \rightarrow \mathbb{R}$ that associates a payment for agent i with each terminal history. The set

of *realizable histories* induced by m is simply that subset of histories h for which $a(h^{k+1}) = m(h^{\leq k})$.[2]

An agent *strategy* π_i associates a response $\pi_i(h_i, q_i; t_i) \in R_i(q_i)$ with every query, conditioned on its local history and its type. [3] Strategies π_i and types t_i together with m induce a specific (possibly unbounded) history: $h(m, \pi, t)$. Since each response is associated with a partial type, for any length k local history h_i we say that $\theta_i(h_i) = \cap_{j \leq k} \theta_i(r^j)$ is the *revealed partial type* of agent i (that is, i has represented his type to lie within the partial types associated with each response). We say π_i is *truthful* iff $t_i \in \theta_i(\pi_i(h_i, q_i; t_i))$ for all $t_i \in T_i$, $q_i \in Q_i$ and $h_i \in \mathcal{H}_i$. A truthful strategy is necessarily *history independent* if responses correspond to disjoint partial types.

We say an incremental mechanism M is *direct* iff m and p_i depend only on the partial types that are revealed and not on the precise history: that is, $m(h) = m(h')$ if $\theta_i(h) = \theta_i(h')$ for each $i \leq n$, and similarly for payments p_i. We restrict attention to direct mechanisms, and write $m(\theta)$ and $p_i(\theta)$ to emphasize the dependence of the mechanism decisions only on the partial type vector revealed so far. If $m(\theta) \in \mathbf{X}$, we write $x^*(\theta)$ to denote the outcome chosen. An incremental mechanism is a *partial revelation mechanism* if there exists a realizable terminal history h and agent i such that $\theta_i(h_i)$ admits more than one possible type t_i. In other words, it is possible for the mechanism to terminate without full knowledge of the types of all agents.

Given a mechanism m and response policies π_{-i} for agents other than i, the utility of agent i of type t_i for using strategy π_i is defined as:

$$u_i(\pi_i, \pi_{-i}; t_i) = v_i(x^*(\theta(h)); t_i) - p_i(\theta(h))$$

if the history h induced by $\langle \pi_i, \pi_{-i} \rangle$ is terminal. Otherwise, we set $u_i = 0$.

Classic direct mechanisms can be viewed as a special case of incremental mechanisms in which each agent is asked directly "What is your type?" and the mechanism then terminates with the appropriate outcome and payment functions.

We continue by describing some general properties of direct, incremental, partial revelation mechanisms.

Definition 1. *A direct mechanism $M = \langle m, p \rangle$ satisfies* δ-allocation certainty *iff for all realizable terminal histories h, $\mathbf{x}^*(\theta(h))$ is such that*

$$\forall t \in \theta(h), \forall \mathbf{x} \in \mathbf{X}, SW(x^*(\theta(h)); t) \geq SW(\mathbf{x}; t) - \delta$$

M satisfies allocation certainty *if this holds for $\delta = 0$.*

That is, M is δ-allocation certain if, whenever it terminates, it has enough information about agent types to determine a δ-efficient allocation.

Definition 2. *A mechanism $M = \langle m, p \rangle$ is* δ-efficient *iff it is* terminating *(i.e., all realizable histories are terminal) and δ-allocation certain. M is* efficient *if it is terminal and is allocation certain.*

Allocation certainty (or its approximation) does not imply that the mechanism "knows" the social welfare of the chosen outcome, only that it is (within δ of) optimal.

[2] M need only specify queries, etc. for realizable histories.

[3] We assume i knows only its own history; this can be relaxed to admit (partial) revelation of other agent queries/responses.

Regret-based Allocation Elicitation

In contrast to standard direct mechanisms, partial revelation mechanisms do not generally allow for the exact optimization of social welfare. Allocation decisions made in the face of type uncertainty run the risk of being suboptimal unless enough type information is obtained to admit allocation certainty. However, we will often be interested in mechanisms that do not reach allocation certainty in order to relieve elicitation burden. In such a case, some means of making decisions in the presence of type uncertainty is required.

The concept of *minimax regret* has recently been proposed and studied as a means of optimization in (single-agent) decision problems in the face of (non-probabilistic) utility function uncertainty and for driving utility elicitation [2; 4; 3]. We describe the minimax regret notion as well as one elicitation strategy in the context of incremental, partial revelation mechanisms.

Suppose a partial revelation mechanism must choose some allocation $\mathbf{x} \in \mathbf{X}$ with access to only incomplete information about agent utility functions via a partial type vector θ. We define the minimax regret of $\mathbf{x}$ as follows:

Definition 3. *The* pairwise regret *of decision $\mathbf{x}$ with respect to decision $\hat{\mathbf{x}}$ over feasible type set θ is*

$$R(\mathbf{x}, \hat{\mathbf{x}}, \theta) = \max_{t \in \theta} SW(\hat{\mathbf{x}}; t) - SW(\mathbf{x}; t), \quad (1)$$

This is the most one could regret choosing $\mathbf{x}$ instead of $\hat{\mathbf{x}}$ (e.g., if an adversary could impose any type in θ). The maximum regret *of decision $\mathbf{x}$ and the* minimax regret *of feasible type set θ are, respectively:*

$$MR(\mathbf{x}, \theta) = \max_{\hat{\mathbf{x}}} R(\mathbf{x}, \hat{\mathbf{x}}, \theta) \quad (2)$$

$$MMR(\theta) = \min_{\mathbf{x}} MR(\mathbf{x}, \theta) \quad (3)$$

A *minimax-optimal* decision is any $\mathbf{x}^*$ that minimizes Eq. 3. Without distributional information over the set of possible utility functions, choosing a minimax-optimal decision $\mathbf{x}^*$ minimizes the worst case loss with respect to possible realizations of the types $t \in \theta$. We refer to the regret maximizing $\hat{\mathbf{x}}$ in Eq. (3) as the *witness* for $\mathbf{x}$.

Minimax regret optimization can be difficult in general, but recent approaches show how it can be made practical when utility models are *factored* into a convenient functional form such as *generalized additive independence (GAI)* [8], and utility uncertainty is expressed in the form of linear constraints on such factored models [2; 4]. In this setting, minimax regret optimization can be formulated as a linear, mixed-integer program with exponentially many constraints, but can be solved using an iterative constraint generation procedure that, in practice, enumerates only a small number of (active) constraints [2; 3].

Several elicitation strategies have been proposed that attempt to reduce minimax regret quickly. We describe one strategy here, called the *current solution strategy (CSS)*, which has proven quite effective in both constraint-optimization problems with GAI models [3] and winner determination in CAs with linear utility models [4]. CSS works as follows: given the current feasible type region θ, let $\mathbf{x}^*$ and $\hat{\mathbf{x}}$ be the minimax optimal and witness allocations, respectively. Each of these allocations involves a specific

instantiation of the GAI factors of the agents' valuations, hence regret can be reduced only by imposing additional constraints on θ that tighten our knowledge of at least some of these parameters. Direct *bound* queries can be posed that ask the user to tighten the bounds on one of these parameters. CSS queries the parameter with the loosest bounds, among those instantiated in $\mathbf{x}^*$ and $\hat{\mathbf{x}}$, and has been shown to be extremely effective in practice in reducing regret.

Applying regret-based elicitation to the design of incremental mechanisms works as follows: after each query the mechanism (assuming truthful revelation) knows that agent types lie within some θ and computes $MMR(\theta)$ and the minimax optimal $\mathbf{x}^*$. If $MMR(\theta) \leq \delta$, the mechanism terminates with $\mathbf{x}^*$; otherwise, the current solution $\mathbf{x}^*$ and witness $\hat{\mathbf{x}}$ are used to determine the next query. The termination condition ensures such a mechanism satisfies δ-allocation certainty. If we set $\delta = 0$, then full allocation certainty is achieved and the mechanism is efficient if we can guarantee termination. While these elicitation techniques generally converge to zero-regret in practice, in the worst case no elicitation technique that imposes linear constraints on the space of valuations can be guaranteed to terminate finitely; however, regret can be made arbitrarily small [2].

Incentive Properties

The elicitation strategy defined in the previous section satisfies (exact or approximate) allocation certainty and therefore provides an (exact or approximate) efficient allocation function. To fully define a mechanism, we require a payment scheme that induces reasonable incentive properties. To this end, we propose a partial revelation analog of VCG payments. We describe its incentive properties when applied directly after the allocation elicitation phase described above (with no *additional* information elicited to determine payments); and then propose a strategy for payment elicitation in a subsequent phase.

Allocation Certainty

We first define a class of mechanisms that is a partial revelation generalization of VCG. Let $\mathcal{M}$ have a δ-efficient allocation function m that terminates with δ-allocation certainty and a partial type vector θ. An agent's payment is simply the maximum VCG payment over all the possible types of other agents. More precisely, let $\mathcal{M} = \langle m, (p_i^\top)_{i \leq n} \rangle$ where:

- m is δ-efficient
- $p_i^\top(\theta) = \max_{t_{-i} \in \theta_{-i}} p_i^v(\mathbf{x}^*(\theta), t_{-i})$

where p_i^v is the VCG payment scheme:

$$p_i^v(\mathbf{x}, t_{-i}) = \max_{\mathbf{x}_{-i}} SW_{-i}(\mathbf{x}_{-i}; t_{-i}) - SW_{-i}(\mathbf{x}; t_{-i})$$

We refer to this payment scheme under partial types as *partial VCG payment*.

Thm 1. *Let $\mathcal{M}$ have a δ-efficient allocation function and use partial VCG payments. Then $\mathcal{M}$ is a δ-efficient,* δ-ex post *individually rational,* $(\delta + \varepsilon(x^*(\theta)))$-ex post *incentive compatible mechanism, where $\varepsilon(\mathbf{x}) = \max_i \varepsilon_i(\mathbf{x})$, and:*

$$\varepsilon_i(\mathbf{x}) = \max_{t'_{-i} \in \theta_{-i}} p_i^v(\mathbf{x}, t'_{-i}) - \min_{t_{-i}} p_i^v(\mathbf{x}, t_{-i}) \qquad (4)$$

Such a partial revelation mechanism will obviously determine an allocation whole social welfare is within δ of optimal if all agents reveal their partial types truthfully. The partial VCG payment scheme also induces γ-ex-post incentive compatibility (where $\gamma = \delta + \varepsilon(x^*(\theta))$). This means that the gain an agent can attain by revealing its partial type incorrectly is bounded by γ, when all others reveal truthfully, even if the agent knows the types of the others. Finally, it is approximately individually rational, so no agent can gain more than δ (even if it knows the others' types) by not participating in the mechanism.

When dealing with *approximate* incentive properties, one must be aware of the fact that a small deviation from the truth by one agent can cause major changes in the mechanism's allocation (thus leading, say, to large losses in efficiency). But with partial VCG payments, an agent can gain at most γ compared to revealing its partial type truthfully. In most settings, the computational cost of finding a good lie—due to the considerable uncertainty in the value of a lie due to uncertainty about the types of others)—will be substantial. Thus, if γ is small enough, it will not be worth the cost: our *formal, approximate* incentive compatibility is sufficient to ensure *practical, exact* incentive compatibility. Of course, if the query strategy used by $\mathcal{M}$ only tackles *SW-regret* (i.e., regret w.r.t. efficiency) δ, we may not be satisfied with the bound γ—we address this in the next section.

To develop a sense of the difficulty associated with manipulating such a mechanism, consider that an agent must be able to compute an untruthful strategy (or lie) with greater utility than truth-telling to exploit our approximate incentive guarantee. An optimal lie reduces an agent's payment to his true VCG payment without changing the choice of the efficient allocation.[4] To compute such a lie, the agent must have considerable (and accurate) information about the types of the others. For example, in a one-item auction with two agents a and b with valuations 7 and 5, respectively, a's optimal lie is one that leads to the efficient allocation—he wins the item—while reducing his payment to the true VCG payment (5). If a knows that b has type 5, the optimal lie is clear: pretend that his type is $5 + \epsilon$, leading to the same (efficient) allocation but with payment arbitrarily close to 5 (depending on the elicitation strategy used).

Of course, things are more difficult than this. Types of other are only known probabilistically—in our example, a will only have a distribution over b's type, and by underbidding he runs the risk of losing the item. If the elicitation process is nondeterministic, agents may also have beliefs about its execution given the types of others: $Pr(\theta(h(m, \pi)))|t)$. Thus manipulation requires a large amount of computation, requiring simulation of the elicitation process (e.g., the regret computations above) for all type vectors, to determine $Pr(\theta|t_i)$ for any "lie" t_i it might report. The costliness of such computations (e.g., in time, cognitive, or computational resources) implies that manipulation is not worthwhile unless the bound in Thm. 1 is quite loose.

[4] If the allocation is δ-efficient, the optimal lie would also attempt to manipulate the approximation of the allocation function. This makes it even harder to compute.

A similar argument can be made regarding approximate individual rationality: determining the gain from not participating will be very difficult. A *potential* small loss will be worthwhile for an agent given the savings our mechanism provides in revelation and computational costs (relative to the full revelation alternative).

Payment Elicitation

One might not be satisfied with the guarantees provided by the expected value of ε defined in Thm. 1; if it is too large, our bound $\gamma = \delta + \varepsilon(x^*(\theta))$ on manipulability may not induce truthful partial type revelation. In this case, we would like to continue eliciting in a second phase, after reaching allocation certainty, until we can guarantee that manipulation is bounded by a pre-specified, *type-independent* ε. As discussed above, for a suitably small ε, we can expect to induce truthful revelation for purely practical reasons.

The elicitation strategies above will not provide useful queries for payment elicitation since allocation certainty has been achieved. So we directly elicit information to determine payments that reduce the worst-case bounds on manipulability. This two-phase approach is similar in spirit to other elicitation schemes that first determine an efficient allocation and then elicit further information to determine VCG payments [9; 12]. Our model differs slightly in that we do not *require* allocation or payment certainty.

Once allocation certainty has been reached, i's payment uncertainty depends only on other agent valuations for the chosen allocation, as well as the optimal allocation in the sub-economy with agent i removed. In practice, one can compute the types $t^{\top}_{-i}$ and $t^{\bot}_{-i}$ that define i's max and min payments in $\mathbf{x}^*$ in Eq. 4, respectively, as well as the allocations $\mathbf{x}^{\top}_{-i}$ and $\mathbf{x}^{\bot}_{-i}$ that are optimal, under those types, in the sub-economy. Given these, we have:

$$\begin{aligned}\varepsilon_i(\mathbf{x}^*) &= SW_{-i}(\mathbf{x}^{\top}_{-i}; t^{\top}_{-i}) - SW_{-i}(\mathbf{x}^*; t^{\top}_{-i}) \\ &\quad -SW_{-i}(\mathbf{x}^{\bot}_{-i}; t^{\bot}_{-i}) + SW_{-i}(\mathbf{x}^*; t^{\bot}_{-i})\end{aligned}$$

In the spirit of the *current solution* strategy, we query the GAI parameter, among those involving these three allocations, with the most uncertainty. Note that if ε is required to be very close to zero, it is possible that we will have to elicit enough information to determine the efficient allocation of the sub-economy x^*_{-i} (so that $\mathbf{x}^{\top}_{-i} = \mathbf{x}^{\bot}_{-i}$) with certainty. This is not necessarily the case however, and, unlike other two-phase approaches, we might terminate without knowing either the x^*_{-i}'s, their social welfare, or the social welfare of $\mathbf{x}^*$ in the sub-economies.

Direct Optimization

The design approach presented so far follows that of most other work in the literature by decomposing the mechanism into two phases: one to reduce allocation uncertainty to a satisfactory level and choose an allocation $\mathbf{x}^*$; and another that independently tackles payment uncertainty in $\mathbf{x}^*$. However, in a partial revelation setting, the allocation defines both the incurred loss in efficiency as well as a large part of the payment uncertainty. The choice of $\mathbf{x}^*$ should therefore account for both criteria. Moreover, the true type of the agents is unique and, along with $\mathbf{x}^*$, defines both efficiency and payment uncertainty. The optimization of both criteria should therefore not be independent. Finally, when designing approximately incentive compatible mechanisms, the main objective is to reduce manipulability below a given threshold. The sum of the efficiency and payment uncertainty bounds is only an upper bound: actual manipulability may be significantly lower, which can allow us to terminate with fewer queries.

If the true type profile is t, the manipulability of our mechanism, when choosing $\mathbf{x}^*$ and applying our payments $p^{\top}_i$, is the maximum over all agents of the difference between the agent's best-case utility and its actual utility. So the manipulability of agent i is expressed as:

$$\alpha_i(\mathbf{x}^*, t) = \max_{\hat{\mathbf{x}}}[v_i(\hat{\mathbf{x}}; t_i) - p^v_i(\hat{\mathbf{x}}; t_{-i})] - v_i(\mathbf{x}^*; t_i) + p^{\top}_i(\mathbf{x}^*; \theta_{-i})$$

The manipulability of the mechanism is $\max_i\{\alpha_i(\mathbf{x}^*, t)\}$, and the *worst-case* manipulability is $\alpha = \max_t \max_i\{\alpha_i(\mathbf{x}^*, t)\}$. We say $\mathcal{M}$ is α-*manipulable* if this expression holds.

Thm 2. *Let $\mathcal{M}$ be an α-manipulable mechanism using partial VCG payments. Then $\mathcal{M}$ is an α-efficient,* α-ex post *individually rational, and* α-ex post *incentive compatible.*

Finding the allocation that minimizes worst-case manipulability is equivalent to solving the following optimization:

$$\begin{aligned}\mathbf{x}^* = \quad & arg\min_{\mathbf{x}} \max_{i,\hat{\mathbf{x}}} R_i(\mathbf{x}, \hat{\mathbf{x}}) \\ \text{where:} \quad & R_i(\mathbf{x}, \hat{\mathbf{x}}) = \\ \max_t \quad & [\, v_i(\hat{\mathbf{x}}; t_i) - v_i(\mathbf{x}; t_i) + p^{\top}_i(\mathbf{x}; \theta_{-i}) - p^v_i(\hat{\mathbf{x}}; t_{-i})] \\ = \quad & \max_{t_i} (\, v_i(\hat{\mathbf{x}}; t_i) - v_i(\mathbf{x}; t_i)\,) \\ & + \max_{t'_{-i}} p^v_i(\mathbf{x}; t'_{-i}) - \min_{t_{-i}} p^v_i(\hat{\mathbf{x}}; t_{-i})\end{aligned}$$

This is also a regret minimization problem, where regret is with respect to the *global* utility of an agent.

The high-level idea of regret-based elicitation naturally applies: given an *a priori* partial type of the agents, we compute the minimax-optimal allocation $\mathbf{x}^*$ and, in the process, the witness corresponding to the adversary's choice $\hat{\mathbf{x}}_i$ for each i. If the regret of $\mathbf{x}^*$ (i.e., manipulability) is not small enough, we choose a query that attempts to reduce the regret of our current solution and iterate until we reach the given threshold.

The regret minimization problem can be reformulated as:

$$\begin{aligned}\min_{\mathbf{x},\delta} \quad & \delta \quad \text{such that } \forall\hat{\mathbf{x}},\ \forall i, \\ \delta \quad & \geq \quad \max_{t_i} (\, v_i(\hat{\mathbf{x}}; t_i) - v_i(\mathbf{x}; t_i)\,) \\ & \quad + \max_{t'_{-i}} p^v_i(\mathbf{x}; t'_{-i}) - \min_{t_{-i}} p^v_i(\hat{\mathbf{x}}; t_{-i})\end{aligned}$$

The maximum payment of i in $\mathbf{x}$ is not linear in $\mathbf{x}$, but can be linearized by generating (or enumerating) allocations that are potentially optimal when agent i is removed (the allocations used to define VCG payments).

For each i, generating the witness $\hat{\mathbf{x}}_i$ that most violates the constraints given a current solution $\mathbf{x}^*$ involves solving:

$$\hat{\mathbf{x}}_i = arg\max_{\hat{\mathbf{x}}} \left[\max_{t_i} [\, v_i(\hat{\mathbf{x}}; t_i) - v_i(\mathbf{x}^*; t_i)\,] - \min_{t_{-i}} p^v_i(\hat{\mathbf{x}}; t_{-i})\right]$$

Since the minimum payment is not linear either, this optimization requires its own round of constraint generation. Witness generation is equivalent to a minimax regret problem. We leave details for a longer version of the paper.

Elicitation Strategies and Empirical Results

We consider three elicitation strategies with the aim of reducing manipulability (i.e., α) and *SW-regret* (i.e., maximum loss in efficiency δ) of the chosen outcome with as few queries as possible.

To set the stage, recall that we have defined CSSs (current solution elicition strategies) w.r.t. both social welfare (call this SW-CSS) and payment uncertainty (P-CSS) in previous sections. Though the details have been omitted, computing the α-minimizing $\mathbf{x}^*$ provides us with three witnesses: $\hat{\mathbf{x}}$, corresponding to the adversary's choice, and $\mathbf{x}^{\top}_{-i}$ and $\mathbf{x}^{\perp}_{-i}$, corresponding to the optimal allocations in the subeconomies that define the payments of agent i in $\mathbf{x}^*$ and $\hat{\mathbf{x}}$, respectively. This leads to a third CSS (M-CSS, for manipulability) that queries the parameter among these four allocations that has the largest gap. While M-CSS seems appealing on the surface, it in fact performs quite poorly since it tends to ask queries that reduce payment uncertainty early on for allocations that won't in fact be realized. So instead, we will use these as sub-strategies in the three methods we explore.

The first strategy we test is *two-phase (2P)*, in which we first run SW-CSS until SW-regret δ reaches zero (or some small threshold), finding an efficient allocation, then run P-CSS to determine appropriate payments until $\delta + \varepsilon$ is less than some threshold. This is much like standard two-phase approaches. The *α-two phase (α2P)* strategy works exactly like 2P, but terminates when the manipulability bound α is below some threshold. Intuitively, this more accurately reflects the quality of the current decision. The third strategy is called *common-hybrid (CH)* and proceeds as follows. (a) Let A be the set of GAI-parameters instantiated in the two allocations (SW-regret minimizing and its witness) that determine SW-regret; let B be the analogous set of parameters among the four allocations that determine worst-case manipulability. If these sets have any parameter in common, we query that common parameter with the largest gap. (b) If no parameters are in common, then we use a hybrid method that chooses between SW-CSS and M-CSS, with a bias towards SW-CSS early on (for reasons explained above).[5]

We compared 2P, α2P, and CH on a car rental problem of moderate size (based on [3]), where a buyer wants to rent a car from one of two dealers, and the buyer's valuation and dealer costs exhibit GAI structure. A car is defined by eight attributes (e.g., engine size, seating, etc.) with domain size ranging from two to nine. Each of the three agents' utilities has 13 factors with factor sizes ranging from one to four variables (giving a total of 825 utility parameters). We also compared these strategies with a myopically optimal strategy (MY) on small, randomly generated, supplier-selection problems, where a buyer negotiates with several sellers over a multi-attribute item to trade. These problems have 81 utility parameters. MY considers querying the midpoint of each parameter, computes the two new global regret levels that could result from each each response, and asks the query with the best average regret reduction. Clearly this strategy is only computationally feasible on small problems, but it provides an interesting comparison.

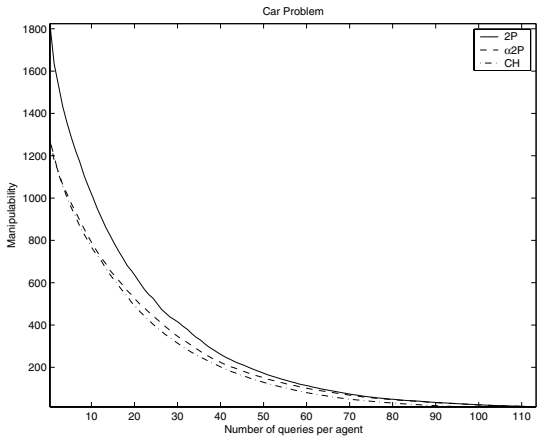

Figure 1: Car Rental Problems. Average of 40 runs. 2 sellers, 1 buyer; 13 factors/agent; 1-4 variables/factor; 2-9 values/variable. 825 parameters total.

Figure 1 shows how manipulability α is reduced as a function of the number of queries (per agent) for 2P, α2P and CH on the car rental problem. α2P and CH, the two strategies that exploit manipulability, exhibit better anytime behavior than 2P, with a slight advantage for CH. 2P and α2P reach near-zero manipulability in roughly the same number of queries (around 110 per agent), while CH reaches the same level in about 95 queries. Independent of the specific strategy, our results make a strong case for regret-based elicitation in mechanism design, as it effectively minimizes elicitation effort. On average only 8% of the utility parameters are ever queried by CH. Furthermore, these are not completely determined, since we only tighten the initial bounds. Regret-based elicitation terminates with 92% of the initial utility uncertainty remaining on average (as measured by the "perimeter" of the partial type space) whereas halving the gap of the most uncertain parameter (a theoretically motivated method uninformed by regret [3]) leaves only 64% of the uncertainty remaining after the same number of queries, and is still very far from reaching zero-manipulability. This indicates that regret-based strategies focus on *relevant* information, rather than reducing uncertainty for its own sake, thus reducing revelation and improving decision quality. Note that initial regret prior to elicitation is on average 99% of optimal social welfare. Despite this, zero-regret (true efficiency) is attained in only 71 and 77 queries, respectively, for α2P and CH, despite the complexity of the problem (involving 825 parameters).

Figure 2 compares CH and MY on small random prob-

[5]The best M-CSS query is selected only if its utility gap is at least b times that of the best SW-CSS query. We use $b = \max(0, 10 - 0.005 \sum_{i=1}^{n} i)$ at query n in our experiments. With this setting, after 60 queries (total over all agents) M-CSS will always be selected.

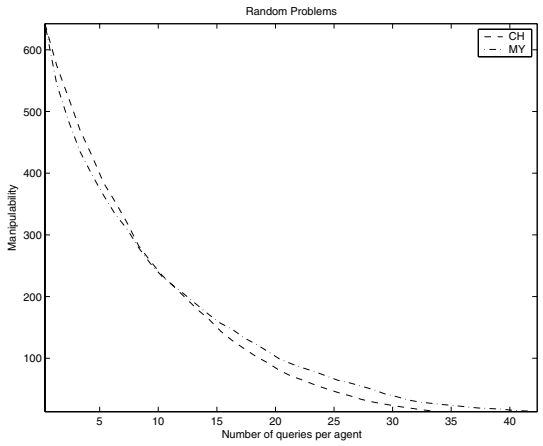

Figure 2: Small Problems. Average of 40 runs. 2 sellers, 1 buyer; 3 factors/agent; 2 variables/factor; 3 values/variable. 81 parameters total.

lems. The large amount of additional computation required by MY allows for reasonable anytime behavior, but it is still outperformed by CH except in the earliest stages of elicitation. CH reaches near-zero manipulability in about 15 fewer queries (45 vs. 60). While counter-intuitive, this behavior is plausibly explained by the fact that since CH focuses on "relevant" parameters, it implicitly provides some sequential guidance: the parameters instantiated in the various regret allocations are likely to remain relevant throughout a large part of the elicitation process. However, further investigation of this phenomenon is needed.

Concluding Remarks

We have described a regret-based approach to the design of incremental, partial revelation mechanisms, using minimax regret to make allocation decisions in the presence of type uncertainty. We examined the incentive properties of several regret-based schemes. With only approximate allocation certainty, we showed that the partial VCG payment scheme allows one to bound *ex post* manipulability. We also described how to elicit additional payment information to provide *a priori* bounds when our payment scheme is used. Finally, we argued for a unified approach in which elicitation is directed specifically at reducing global manipulability (which automatically bounds the loss in efficiency). With sufficiently small bounds, the cost of manipulation will generally outweigh its potential gain, so that *formal, approximate* incentive compatibility will be sufficient to ensure *practical, exact* incentive compatibility. Our payment scheme also has the positive side-effect of increasing revenue relative to VCG. Our approach can be applied in any mechanism design context, unlike most previous incremental elicitation schemes, and yet still exploits the inherent structure of the specific setting. Empirically, our schemes seem to be very effective.

We have recently begun to explore one-shot partial revelation mechanisms [10], and several other interesting directions remain to be explored. Precisely determining the complexity of manipulation is an important task in further justifying our emphasis on approximate incentive compatibility and individual rationality. Further study into *sequentially* optimal elicitation and a theoretic analysis of the communication complexity of regret-based mechanisms are of great interest as well.

Acknowledgments Thanks to Vincent Conitzer, Kevin-Leyton Brown and David Parkes for helpful discussions. This work was funded by NSERC.

References

[1] L. Blumrosen and N. Nisan. Auctions with severely bounded communication. *FOCS-02*, pp.406–416, Vancouver, 2002.

[2] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans. Constraint-based optimization with the minimax decision criterion. *CP-03*, pp.168–182, Kinsale, 2003.

[3] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans. Regret-based utility elicitation in constraint-based decision problems. *IJCAI-05*, pp.929–934, Edinburgh, 2005.

[4] C. Boutilier, T. Sandholm, and R. Shields. Eliciting bid taker non-price preferences in (combinatorial) auctions. *AAAI-04*, pp.204–211, San Jose, 2004.

[5] W. Conen and T. Sandholm. Partial-revelation VCG mechanisms for combinatorial auctions. *AAAI-02*, pp.367–372, Edmonton, 2002.

[6] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, Cambridge, 2005.

[7] R. Fadel and I. Segal. The communication cost of selfishness: ex post implementation. *TARK-05*, Singapore, pp.165–176, 2005.

[8] P. C. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *Intl. Econ. Rev.*, 8:335–342, 1967.

[9] B. Hudson and T. Sandholm. Generalizing preference elicitation in Combinatorial Auctions. *AAMAS-03*, pp.1014–1015, Melbourne, 2003.

[10] N. Hyafil and C. Boutilier. Mechanism design with partial revelation. *Working paper*, 2006.

[11] S. Lahaie, F. Constantin, and D. Parkes. More on the Power of Demand Queries in CAs: Learning Atomic Languages and Handling Incentives. *IJCAI-05*, 2005.

[12] S. Lahaie and D. C. Parkes. Applying learning algorithms to preference elicitation. *ACM EC-04*, pp.180–188, 2004.

[13] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, New York, 1995.

[14] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. *ACM EC-00*, 242-252, Minneapolis, 2000.

[15] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *J. Econ. Th.*, 2005.

[16] D. Parkes. Auction design with costly preference elicitation. *Annals of Math. and AI* 44:269–302, 2005.

[17] T. Sandholm, V. Conitzer, and C. Boutilier. Automated design of multistage mechanisms. *Workshop on Incentive Based Computing*, pp.2–12, Compiegne, 2005.

[18] M. Zinkevich, A. Blum, and T. Sandholm. On polynomial-time preference elicitation with value queries. *ACM EC-03*, pp.176–185, San Diego, 2003.