Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2009-061 December 4, 2009

Perfect and General Virtual Implementation For Perfectly Informed Players

Jing Chen and Silvio Micali

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Perfect and General Virtual Implementation For Perfectly Informed Players

Jing Chen
CSAIL, MIT
Cambridge, MA 02139, USA
jingchen@csail.mit.edu

Silvio Micali
CSAIL, MIT
Cambridge, MA 02139, USA
silvio@csail.mit.edu

December 4, 2009

Abstract

We show that, when the players are perfectly informed about each other, essentially all social-choice functions can be rationally robustly implemented via an extensive-form public-action mechanism that (1) is perfectly robust against collusion, (2) requires only a linear number of computation steps and communication bits, and (3) preserves the privacy of the players' types to a very high extent.

1 Notations

- The set $N = \{1, \ldots, n\}$ is the set of players;
- For each player i, Γ_i is the space of types of i, and $\Gamma = \prod_i \Gamma_i$ is the space of type profiles;
- The outcome space is Ω;
- A perfect-knowledge context is a 4-tuple $C = (T, \Omega, u, \mathbb{C})$, where
  - $T \in \Gamma$ is the profile of the players true types;
  - each u_i is the utility function for player i, mapping player i's true type T_i and a state ω to a real number;
  - $\mathbb{C}$ is the collusion structure, namely, a partition of N.

  Each subset $S \in \mathbb{C}$ of players is called an agent. S, as well as each player in S, is collusive if $|S| > 1$, or independent otherwise. S is also called a collusive set if $|S| > 1$. C is called an independent context if every player is independent, or a collusive context otherwise. When C is an independent context, $\mathbb{C}$ is typically omitted for simplicity.

  Ω is publicly known to everybody, including the mechanism designer. T, u, and $\mathbb{C}$ are publicly known to the players, and we emphasize that the mechanism designer knows nothing about them.
- A function $f : \Gamma \times \Omega \to \mathbb{R}$ is called a (deterministic) numerical social choice function.

  Note a (deterministic) traditional social-choice function F maps type profiles to outcomes, $F : \Gamma \to \Delta(\Omega)$, and, given a context C with true-type profile T, the goal of a mechanism is to produce the outcome $F(T)$. But it is clear that this goal can be achieved by considering the numerical social-choice function f that, on input (γ, ω) returns 1 if $\omega = F(\gamma)$ and 0 otherwise, and maximizing $f(T, \cdot)$.

Our assumptions about contexts C and social choice functions f are the same as [5], namely:

(1) For each player i, u_i is not a constant function, and for any two different types $\gamma_i, \gamma_i' \in \Gamma_i$, $u_i(\gamma_i, \cdot)$ is not a linear transformation of $u_i(\gamma_i', \cdot)$.

(2) For each player i, there exists a function $g_i : \Gamma_i \to \Omega$ such that for any two different types $\gamma_i, \gamma_i' \in \Gamma_i$, $u_i(\gamma_i, f_i(\gamma_i)) > u_i(\gamma_i, f_i(\gamma_i'))$.

(3) A mechanism M specifies the players' strategies, and the outcome function, denoted also by M. Given a strategy profile σ, $M(\sigma)$ consists of an outcome $\omega \in \Omega$ and a price profile $P \in \mathbb{R}^n$ — that is, M can impose a price to each player. Given C and (ω, P), each player i's utility is $U_i(T_i, \omega, P_i) = u_i(T_i, \omega) - P_i$.

We use the solution concept proposed in [3, 4], namely, rationally robust implementation. Our mechanism is based on that in [2], in particular, it is an extensive-form mechanism with public actions. The setting considered is finitely, in particular: the type space is finite, the outcome space is finite, and the image spaces of the social choice function f and g_i's for each i are all finite.

2 Our Mechanism

Notation In the mechanism below,

- ϵ and ϵ_j^i, for $i \in \{2, \ldots, n\}$ and $j \in \{1, \ldots, n\}$, are constants such that $\frac{1}{5n} > \epsilon > \epsilon_1^2 > \cdots > \epsilon_n^2 > \epsilon_1^3 > \ldots > \epsilon_n^3 > \cdots > \epsilon_1^n > \cdots > \epsilon_n^n > 0$.
- Numbered steps are taken by the players, while steps marked by letters are taken by the mechanism.
- Sentences between quotation marks are comments, and could be excised if no clarification is needed.
- We denote by n_r the number of outcomes (ω, P) with revenue r. For all such outcomes, we denote by $0 \leq f_r(\omega, P) < n_r$ the rank of the outcome (ω, P) in the lexicographic order that first considers the state and then the price profile (where $P_1, \ldots P_n$ precedes $P'_1 \ldots P'_n$ whenever $P_1 > P'_1$, etc.).
- $U = \max_{T \in \Gamma, \omega \in \Omega} f(T, \omega)$, where f is the given numerical social-choice function.
- $L = \min_i \min_{T_i, T'_i \in \Gamma_i} u_i(T_i, g_i(T_i)) - u_i(T_i, g_i(T'_i))$.
- ϵ' is a positive constant such that $\epsilon' < \frac{L}{2nU}$.

Mechanism $\mathcal{M}$

(1) *Player 1 announces a state $\omega^\star$ and a profile K^1 of natural numbers.*

"$(\omega^\star, K^1)$ is player 1's proposed outcome, allegedly an outcome of maximum revenue."

(a) *Set $\omega = \perp$, and $P_i = 0\ \forall i$. If $\sum_i K_i^1 = 0$, the mechanism ends right now. Otherwise, proceed to Step 2.*

"Whenever the mechanism ends, ω and P will be, respectively, the final state and price profile."

(2,…,n) *In Step i, $2 \leq i \leq n$, player i publicly announces a profile Δ^i of natural numbers such that $\Delta_i^i = 0$.*

"By so doing i suggests to raise the current price of j, that is $K_j^1 + \sum_{\ell=2}^{i-1} \Delta_j^\ell$, by the amount Δ_j^i."

(b) *For each player i, publicly select bip_i and $P_i^\star$ as follows. Let $R_i = \{j : \Delta_i^j > 0\}$. If $R_i \neq \emptyset$, then bip_i is highest player in R_i, and $P_i^\star = K_i^1 + \sum_{\ell=2}^{bip_i} \Delta_i^\ell$. Else, $bip_i = 1$ and $P_i^\star = K_i^1$.*

"We refer to bip_i as the best informed player about i, and to $P_i^\star$ as the provisional price of i."

($n+1$) *Each player i such that $P_i^\star > 0$ simultaneously announces YES or NO. By default, each player i such that $P_i^\star = 0$ announces YES, and player 1 announces YES if $bip_1 = 1$.*

"Each player i announces YES or NO to $\omega^\star$ as the final state and to $P_i^\star - \epsilon$ as his own price. (By default player 1 accepts his own price if no one raises it.) At this point the players are done playing, and the mechanism proceeds as follows. If all say YES, the updated proposal $(\omega^\star, P^\star)$ is implemented with probability 1. Else:

- With very high probability the null outcome is chosen, except that the best-informed players of those saying NO are punished.

- With small probability the null outcome is chosen
- With very small probability, proportional to the number of players saying YES, we implement $(\omega^\star, P^\star)$ as if all said YES.

Importantly, as we shall see, all get a small reward at the end for their knowledge."

(b') *Publicly flip a biased coin* c_0 *such that* $\Pr[c_0 = Heads] = 1-\epsilon$. *If* $c_0 = Heads$ *then proceed to Step c. Otherwise do the following:*

(n+2) Each player i simultaneously announces a type $\gamma_i \in \Gamma_i$.

(c') If $\omega^\star = \mathrm{argmax}_{\omega' \in \Omega} f(\gamma, \omega')$, *then let* $v = f(\gamma, \omega)$ *and reset* P_1 *to* $P_1 - \epsilon' v$; *otherwise reset* P_1 *to* $\frac{L}{2n}$.

(d') Choose a random player i uniformly, reset ω *to be* $g_i(\gamma_i)$, *and HALT.*

(c) *Let* Y *be the number of players announcing YES. If* $Y = n$, *then reset* ω *to* $\omega^\star$ *and each* P_i *to* $P_i^\star - \epsilon$, *and go to Step g. If* $Y < n$, *proceed to Step d.*

(d) *Publicly flip a biased coin* c_1 *such that* $\Pr[c_1 = Heads] = 1 - \epsilon$.

(e) *If* $c_1 = Heads$, *reset* P_{bip_i} *to* $P_{bip_i} + 2P_i^\star$ *for each player i announcing NO.*

(f) *If* $c_1 = Tails$, *letting* $B = \sum_{i \text{ announces } NO} P_i^\star$, *flip a biased coin* c_2 *such that* $\Pr[c_2 = Heads] = \frac{Y}{nB}$.

If $c_2 = Heads$, *reset* ω *to* $\omega^\star$ *and each* P_i *to* $P_i^\star - \epsilon$.

If $c_2 = Tails$, ω *and* P *continue to be* $\perp$ *and* 0^n.

(g) *Reset* P_1 *to* $P_1 - \epsilon - 2\epsilon \sum_j K_j^1 + \epsilon \frac{f_r(\omega^\star)}{n_r}$ *and each other* P_i *to* $P_i - \epsilon - \sum_j \epsilon_j^i \Delta_j^i$.

"Although players' prices may be negative, we prove that the mechanism never loses money, and that in the unique rational play the utility of every player is non-negative. For clarity, our rewards are proportional to prices and raises."

3 Statement of Our Theorem

Theorem 1. *For every numerical social choice function* f, *every context* $C = (T, \Omega, u, \mathbb{C})$, *and every rationally robust play* σ *of* $(C, \mathcal{M})$, *letting* $(\omega, P) = \mathcal{M}(\sigma)$, *then, with probability* $\geq 1 - \epsilon$:

$$\omega = \mathrm{argmax}_{\omega' \in \Omega} f(T, \omega').$$

In a forthcoming paper we shall extend the above mechanism in a simple way (e.g., by introducing envelopes to the mechanism) to (virtually) implement also all probabilistic numerical (and thus traditional) social choice functions.

References

[1] D. Abreu and H. Matsushima. Virtual Implementation in Iteratively Undominated Strategies: Complete Information. *Econometrica*, Vol. 60, No. 5, pages 993-1008, Sep., 1992.

[2] J. Chen, A. Hassidim, and S. Micali. Robust Perfect Revenue From Perfectly Informed Players. To appear at ICS'10.

[3] J. Chen and S. Micali. A New Approach to Auctions and Resilient Mechanism Design. *Symposium on Theory of Computing*, pages 503-512, 2009. Full version available at http://people.csail.mit.edu/silvio/Selected Scientific Papers/Mechanism Design/.

[4] J. Chen and S. Micali. Rational Robustness for Mechanism Design. Submitted to STOC'10, full version available at *http://people.csail.mit.edu/silvio/Selected Scientific Papers/Mechanism Design/*.

[5] J. Glazer and M. Perry. Virtual Implementation in Backwards Induction. *Games and Economic Behavior*, Vo.15, pages 27-32, 1996.