On Decentralized Incentive Compatible Mechanisms for Partially Informed Environments

by Ahuva Mu’alem
June 2005

presented by Ariel Kleiner and Neil Mehta
Contributions

• Brings the concept of *Nash Implementation* (NI) to the CS literature.
  – Not about learning
• Overcomes a number of limitations of VCG and other commonly-used mechanisms.
• Introduces concepts of *partial information* and *maliciousness* in NI.
• Provides instantiations of results from NI that are relevant to CS.
Overview

• Extension of Nash Implementation to decentralized and partial information settings
• Instantiations of elicitation and trade with partial information and malicious agents
• Applications to peer-to-peer (P2P) networking and shared web cache
Motivation

• Standard models of Algorithmic Mechanism Design (AMD) and Distributed AMD (DAMD) assume
  – rational agents
  – quasi-linear utility
  – private information
  – dominant strategy play

• This paper seeks to relax these last two assumptions in particular.
Motivation: Dominant Strategies

• Dominant Strategy Play: Each player has a best response strategy regardless of the strategy played by any other player
  – Corresponds to Private Information / Weak Information Assumption
  – Vickrey-Clarke-Groves (VCG) mechanisms are the only known general method for designing dominant-strategy mechanisms for general domains of preferences with at least 3 different outcomes. (Roberts’ classical impossibility result)
Motivation: Review of VCG

Def. [VCG mechanism] Implement efficient outcome, 
\[ k^+ = \max_k \sum_j v_j(k, \hat{\theta}_j), \] 
and compute transfers 
\[ t_i(\hat{\theta}) = \sum_{j \neq i} v_j(k^{-i}, \hat{\theta}_j) - \sum_{j \neq i} v_j(k^+, \hat{\theta}_j) \]

where \( k^{-i} = \max_k \sum_{j \neq i} v_j(k, \hat{\theta}_j) \).

Thm. The VCG mechanism is strategyproof, efficient, and:
(1) individual-rational (IR), i.e. utility from participation \( \geq 0 \), for all reports, when \( V(N) \geq V(N \setminus i) \)

(2) no-deficit (ND), i.e. total payments \( > 0 \), for all instances, when no positive externalities and
\[ \sum_{j \neq i} v_j(k^{-i}) \geq \sum_{j \neq i} v_j(k^+). \]
Motivation: Restrictions of VCG

- In distributed settings, without available subsidies from outside sources, VCG mechanisms are not budget-balanced.
- Computational hardness
Motivation: Additional Restrictions

• Social goal functions implemented in dominant strategies must be monotone.
  – Very restrictive - (e.g. Rawls’s Rule)
• Recent attempts at relaxing this assumption result in other VCG or “almost” VCG mechanisms.
Background:
Complete Information Setting

- set of agents $N = \{1, \ldots, n\}$ each of which has a set $S_i$ of available strategies as well as a type $\theta_i$

- set of outcomes $A = \{a, b, c, d, \ldots\}$

- social choice rule $f$ maps a vector of agent types to a set of outcomes

- All agents know the types of all other agents, but this information is not available to the mechanism or its designer.
Background: Complete Information Setting

• A mechanism defines an outcome rule \( g \) which maps joint actions to outcomes.

• The mechanism implements the social choice rule \( f \) if, for any set of agent types, an equilibrium exists if and only if the resulting outcome is prescribed by the social choice rule.

• We will primarily consider subgame-perfect equilibrium (SPE) implementation with extensive-form games.
Background: SPE-implementation

• Advantages of SPE-implementation:
  – relevant in settings such as the Internet, for which there are standards-setting bodies
  – generally results in “non-artificial constructs” and “small” strategy spaces; this reduces agent computation
  – sequential play is advantageous in distributed settings
  – resulting mechanisms are frequently decentralized and budget-balanced
Theorem (Moore and Repullo): For the complete information setting with two agents in an economic environment, any social choice function can be implemented in the subgame-perfect Nash equilibria of a finite extensive-form game. [This result can be extended to settings with more than two agents.]
Background: SPE-implementation

Stage 1: elicitation of Bob’s type, $\theta_B^T$

Stage 2: elicitation of Alice’s type, $\theta_A^T$

Stage 3: Implement the outcome defined by the social choice function: $f(\theta_A^T, \theta_B^T)$. 
Background: SPE-implementation

We require that $p, q, F > 0$ and choose $(a, p)$ and $(b, q)$ here such that

$$v_A(a, \theta_A') - v_A(b, \theta_A') > p - q > v_A(a, \theta_A) - v_A(b, \theta_A)$$

$\iff v_A(a, \theta_A') - p > v_A(b, \theta_A') - q$

$v_A(b, \theta_A) - q > v_A(a, \theta_A) - q$

Outcome

- $(a, p+F, -F)$: fine paid by Alice
- $(b, q+F, F)$: fine paid by Bob

- challenge valid
- challenge invalid
Example: Fair Assignment Problem

- Consider two agents, Alice and Bob, with existing computational loads $L_A^T$ and $L_B^T$.
- A new task of load $t>0$ is to be assigned to one agent.
- We would like to design a mechanism to assign the new task to the least loaded agent without any monetary transfers.
- We assume that both Alice and Bob know both of their true loads as well as the load of the new task.
Example: Fair Assignment Problem

- By the Revelation Principle, the fair assignment social choice function cannot be implemented in dominant strategy equilibrium.

- However, assuming that load exchanges require zero time and cost, the desired outcome can be easily implemented in SPE.
Example: Fair Assignment Problem

Alice

Agree

DONE

Refuse

Bob

Perform

DONE

Exchange then Perfrom

DONE
Example: Fair Assignment Problem

• However, the assumption of no cost for load exchanges is unrealistic.

• We now replace this assumption with the following assumptions:
  – The cost of assuming a given load is equal to its duration.
  – The duration of the new task is bounded: t<T.
  – The agents have quasilinear utilities.

• Thus, we can now adapt the general mechanism of Moore and Repullo.
Example:
Fair Assignment Problem

Stage 1: elicitation of Bob’s load
Stage 2: elicitation of Alice’s load
Stage 3: Assign the task to the agent with the lower elicited load.
Example: Fair Assignment Problem

from stage 1

Alice

Bob

LA’ = LA

LA’ ≤ LA

LA’ ≠ LA

ASSIGN TASK (STAGE 3)

Alice

Bob

LA’ = LA

LA’ ≤ LA

LA’ ≠ LA

challenge valid

challenge invalid

• Alice is assigned new task.
• No load transfer occurs.
• Alice pays $\epsilon$ to Bob.
• DONE

• Alice is assigned new task.
• Alice transfers original load to Bob.
• Alice pays Bob $L_A - 0.5 \cdot \min\{\epsilon, L_A - L_A'\}$
• Alice pays $\epsilon$ to mechanism.
• Bob pays fine of $T + \epsilon$ to mechanism.
• DONE
**Definition:** An agent B is p-informed about agent A if B knows the type of A with probability p.

- This relaxation of the complete information requirement renders the concept of SPE-implementation more amenable to application in distributed network settings.
- The value of p indicates the amount of agent type information that is stored in the system.
Elicitation: Partial Information Setting

• Modifications to complete-information elicitation mechanism:
  – use iterative elimination of weakly dominated strategies as solution concept
  – assume \( L_A^T, L_B^T \leq L \)
  – replace the fixed fine of \( \varepsilon \) with the fine
    \[
    \beta_p = \max\{L, T \cdot (1-p)/(2p-1)\} + \varepsilon
    \]
Example:
Fair Assignment Problem

from stage 1

Alic

\( e_{LA} \)

Bob

\( L_A' \leq L_A \)

\( L_A' = L_A \)

\( L_A' \neq L_A \)

ASSIGN TASK (STAGE 3)

\( \text{challenge valid} \)

\( \text{challenge invalid} \)

- Alice is assigned new task.
- No load transfer occurs.
- Alice pays \( \beta_p \) to Bob.
- DONE

- Alice is assigned new task.
- Alice transfers original load to Bob.
- Alice pays Bob \( L_A - 0.5 \cdot \min\{\beta_p, L_A - L_A'\} \)
- Alice pays \( \beta_p \) to mechanism.
- Bob pays fine of \( T + \beta_p \) to mechanism.
Elicitation: Partial Information Setting

**Claim:** If all agents are p-informed, with $p > 0.5$, then this elicitation mechanism implements the fair assignment goal with the concept of iterative elimination of weakly dominated strategies.
Elicitation: Extensions

• This elicitation mechanism can be used in settings with more than 2 agents by allowing the first player to “point” to the least loaded agent. Other agents can then challenge this assertion in the second stage.

• Note that the mechanism is almost budget-balanced: no transfers occur on the equilibrium path.
Application: Web Cache

• Single cache shared by several agents.
• The cost of loading a given item when it is not in the cache is C.
• Agent i receives value $v_i^T$ if the item is present in the shared cache.
• The efficient goal requires that we load the item iff $\sum v_i^T \geq C$. 
Application: Web Cache

• Assumptions:
  – agents’ future demand depends on their past demand
  – messages are private and unforgeable
  – an acknowledgement protocol is available
  – negligible costs
  – Let $q_i(t)$ be the number of loading requests initiated for the item by agent $i$ at time $t$. We assume that $v_i^T(t) = \max\{V_i(q_i(t-1)), C\}$. $V_i(\cdot)$ is assumed to be common knowledge.
  – Network is homogeneous in that if agent $j$ handles $k$ requests initiated by agent $i$ at time $t$, then $q_i(t) = k\alpha$. 
Application: Web Cache

• For simplicity, we will also assume
  – two players
  – $v_i^T(t) =$ number of requests initiated by $i$ and observed by any informed $j$ (i.e., $\alpha = 1$ and $V_i(q_i(t-1)) = q_i(t-1)$).
Application: Web Cache

**Stage 1:** elicitation of Bob’s value, \( v_B^T(t) \)

**Stage 2:** elicitation of Alice’s value, \( v_A^T(t) \)

**Stage 3:** If \( v_A + v_B < C \), then do nothing.

Otherwise, load the item into the cache, with Alice paying
\[
p_A = C \cdot \frac{v_A}{v_A + v_B}
\]
and Bob paying
\[
p_B = C \cdot \frac{v_B}{v_A + v_B}.
\]
Application: Web Cache

from stage 1

Alice

Bob

$v_A$ = $v_A$

$v_A' \geq v_A$

$v_A' \neq v_A$

COMPLETE STAGE 3

Bob

• Alice pays $C$ to finance loading of item into cache.
• Alice pays $\beta_p = \max\{0, C \cdot (1-2p)/p\} + \epsilon$ to Bob.
• DONE

provides $v_A'$ valid records (i.e., validates challenge)
otherwise

• Bob pays $C$ to finance loading of item into cache.
• DONE
Application: Web Cache

**Claim:** It is a dominated strategy to overreport one’s true value.

**Theorem:** A strategy that survives iterative elimination of weakly dominated strategies is to report the truth and challenge only when one is informed. The mechanism is efficient and budget-balanced and exhibits consumer sovereignty, positive transfer, and individual rationality.
Seller and Buyer: Overview

• One good
• Two states: High and Low
• Buyers and sellers have value s.t. \( l_S < h_S < l_b < h_b \)
  – Values are observable to agents, but not to mechanism
• Price equals the average of the buyer’s and seller’s value in each state
  – State H: \( b_v = \frac{h_b + l_b}{2} \)
  – State L: \( b_l = \frac{h_s + l_s}{2} \)
• Prices are set s.t. trade is feasible regardless of state
  – i.e., \( p_l, p_h \in (h_S, l_b) \)
• Payoffs are \( u_b = x v_b - t, u_s = t - x v_s \)
Seller and Buyer: Payoffs

Payoffs are written as: 
\((U_{Buyer}, U_{Seller})\)

\[\text{Buyer} \quad \text{Offer } p_I \quad (l_b-p_I, p_I-l_s) \quad \text{Trade} \quad (l_b-p_h, p_h-l_s) \quad \Delta, -\Delta \]

\[\text{Buyer} \quad \text{Offer } p_H \quad (h_b-p_I, p_I-h_s) \quad \text{No Trade} \quad (h_b-p_h, p_h-h_s) \quad \Delta, -\Delta \]

\[\text{Seller} \quad \text{Offer } p_I \quad (l_b-p_I, p_I-l_s) \quad \text{Trade} \quad (l_b-p_h, p_h-l_s) \quad \Delta, -\Delta \]

\[\text{Seller} \quad \text{Offer } p_H \quad (h_b-p_I, p_I-h_s) \quad \text{No Trade} \quad (h_b-p_h, p_h-h_s) \quad \Delta, -\Delta \]

\[\text{Nature} \quad L \quad \text{Offer } p_I \quad (l_b-p_I, p_I-l_s) \quad \text{Trade} \quad (l_b-p_h, p_h-l_s) \quad \Delta, -\Delta \]

\[\text{Nature} \quad H \quad \text{Offer } p_H \quad (h_b-p_I, p_I-h_s) \quad \text{No Trade} \quad (h_b-p_h, p_h-h_s) \quad \Delta, -\Delta \]
Seller and Buyer: Mechanism

- The mechanism defines a transfer, $\Delta$, from the seller to the buyer, that occurs when no trade occurs
- $\Delta = l_b - p_h + \varepsilon$
- Without this $\Delta$, (i.e., with only $p_l$ and $p_h$), no mechanism exists that Nash-implements the market
**Claim 4:** Given the state, there exists a unique subgame perfect equilibrium.
Seller and Buyer: Maliciousness

• What would happen if the buyer chose to not trade, even if the true state were H?
  – This is a form of punishment, as the buyer forgoes utility of $h_b - l_b - \varepsilon$
  – Why might the buyer do this?

• Definition: A player is q-malicious if his payoff equals:
  
  $$(1-q) \text{(his private surplus)} - q \text{(the sum of other players’ surpluses)}, \quad \forall \ q \text{ in } [0,1].$$

• (That is, higher q’s are associated with more malicious players)
Seller and Buyer: Maliciousness

• **Claim:** For \( q < 0.5 \), the unique subgame perfect equilibrium for \( q \)-malicious players is unchanged.

• Do we like this definition?
• When do we observe \( q \)-maliciousness?
• Could we have arrived at a more principled definition by considering maliciousness as a rational strategy in repeated games?
Application: P2P Networking

- Suppose there are three agents: Bob, Alice and Ann
- Bob wants file f but doesn’t know if Alice has the file, or if Ann has the file (or if both do).
- A Problem of imperfect information
Application: P2P Networking

• If Bob copies a file \( f \) from Alice, Alice then knows that Bob holds a copy of the file, and stores this information as a certificate \((Bob, f)\)
  – Certificates are distributable
  – An agent holding the certificate is “informed”

• Assume:
  – System, file size homogeneous
  – Agent gains \( V \) for downloading a file
  – Only cost is \( C \) for uploading a file
  – \( up_i \) and \( down_i \) are the number of up- and down-loads by agent \( i \)
  – Agent \( i \) enters the system only if \( up_i \cdot C < down_i \cdot V \)
Application: P2P Networking Mechanism

- 3 p-informed agents: B, A₁, A₂
- B is directly connected to A₁ and A₂

Case 1: B knows that an agent A₁ has the file
  - i.e., B has the certificate (A₁,f)
    B can apply directly to agent A₁ and request the file.
    If A₁ refuses, then B can seek court enforcement of his request
Application: P2P Networking Mechanism

• Case 2: B doesn’t know which agent has the file

**Stage 1:** Agent B requests the file f from A₁
- If A₁ reports “yes,” B downloads f from A₁
- Otherwise
  - If A₂ agrees, goto next stage
  - Else (challenge) A₂ sends a certificate (A₁, f) to B
    - If the certificate is correct, then \( t(A₁, A₂) = \beta p \)
      » \( t(A₁, A₂) \) is the transfer from A₁ to A₂
    - If the certificate is incorrect, \( t(A₂, A₁) = |C| + \varepsilon \)

**Stage 2:** Agent B requests the file f from A₂. Switch the roles of A₂ and A₁.
Seller and Buyer: Payoffs

Payoffs are written as: $(U_{Buyer}, U_{Seller})$

A1
- "Yes" $(V, C, 0)$
- "No" $A2$
  - Agree $(V, 0, C)$
    - "Yes" $(V, -\beta_p + C, \beta_p)$
    - "No" $A2$
      - Challenge $(V, -\beta_p + C, \beta_p)$
        - True $(0, |C| + \varepsilon, -|C| - \varepsilon)$
        - False $(0, -|C| - \varepsilon, |C| + \varepsilon)$
  - Challenge $(V, -\beta_p + C, \beta_p)$
    - True $(0, |C| + \varepsilon, -|C| - \varepsilon)$
    - False $(0, -|C| - \varepsilon, |C| + \varepsilon)$
Application: P2P Networking Mechanism

- **Claim:** *The basic mechanism is budget-balanced (transfers always sum to 0) and decentralized*

- **Theorem:** For $\beta_p = |C|/p + \varepsilon$, $p \in (0,1]$, one strategy that survives weak domination is to say “yes” if $A_i$ holds the file, and to only challenge with a valid certificate. In equilibrium, $B$ downloads the file if some agent holds it, and there are no transfers.
Application: P2P Networking Chain Networks

• $i+1$ $p$-informed agents: $B, A_i$
• $B$ is directly connected to $A_1$, and each $A_i$ to $A_{i+1}$
• Assume an acknowledgment procedure to confirm receipt of a message
• Fine $\beta p + 2\varepsilon$ is paid by an agent for not properly forwarding a message
• **Stage i**
  – Agent $B$ forwards a request for file $f$ to $A_i$ (through $\{A_k\}_{k \leq i}$)
  – If $A_i$ reports “yes,” $B$ downloads $f$ from $A_i$
  – If $A_i$ reports “no”
    • If $A_j$ sends a correct certificate $(A_k, f)$ to $B$, then $t(A_k, A_j) = \beta p$
    • Otherwise, $t(A_k, A_j) = C + \varepsilon$
      If $A_j$ reports he has no copy of the file, then any agent in between can challenge
Discussion

• What is the enforcement story in a decentralized setting? Who implements the mechanism and outcome?
• Motivation was in part budget-balancing. We still rely on transfers, but off the equilibrium path. How are transfers implemented?
• Subgame perfection assumes agent rationality.
• We presently have mechanisms only for \( p > 0.5 \) and \( q < 0.5 \), and we do not consider information maintenance costs or incentives for information propagation (e.g., in the P2P setting).
• Settings with more than 2 agents: what if multiple malicious agents collude?