Intelligent Agents
Mechanism Design

Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

Mechanism Design

- Game Theory + Social Choice
- Goal of a mechanism
  - Obtain some outcome (function of agents' preferences)
  - But agents are rational
    - They may lie about their preferences
- Goal of mechanism design
  - Define the rules of a game so that in equilibrium the agents do what we want

Fundamentals

- Set of possible outcomes, O
- Agents $i \in I$, $|I|=n$, each agent i has type $\theta_i \in \Theta_i$
  - Type captures all private information that is relevant to agent's decision making
- Utility $u_i(o, \theta_i)$, over outcome $o \in O$
- Recall: goal is to implement some system-wide solution
  - Captured by a social choice function (SCF)

$$f: \Theta_1 \times \ldots \times \Theta_n \rightarrow O$$

$f(\theta_1, \ldots \theta_n) = o$ is a collective choice

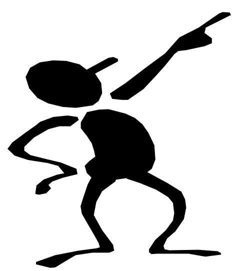
Examples of social choice functions

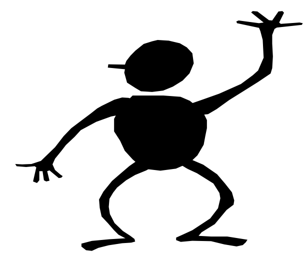
- Voting: choose a candidate among a group
- Public project: decide whether to build a swimming pool whose cost must be funded by the agents themselves
- Allocation: allocate a single, indivisible item to one agent in a group

Mechanisms (From Strategies to Games)

- Recall: We want to implement a social choice function
  - Need to know agents' preferences
  - They may not reveal them to us truthfully
- Example:
  - 1 item to allocate, and want to give it to the agent who values it the most
  - If we just ask agents to tell us their preferences, they may lie

I like the bear the most!





No, I do!

Mechanism Design Problem

- By having agents interact through an institution we might be able to solve the problem
- Mechanism:

$$M=(S_1,\ldots,S_n, g(.))$$

Strategy spaces of agents

Outcome function

$$g:S_1 x \ldots x S_n \rightarrow O$$

Implementation

- A mechanism $M=(S_1,\ldots,S_n,g(.))$ implements social choice function $f(\theta)$ iff

  there is an equilibrium strategy profile

  $$s^*(.)=(s^*_1(.),\ldots,s^*_n(.))$$

  of the game induced by M such that

  $$g(s_1^*(\theta_1),\ldots,s_n^*(\theta_n))=f(\theta_1,\ldots,\theta_n)$$

  for all $(\theta_1,\ldots,\theta_n) \in \Theta_1 x \ldots x\, \Theta_n$

Implementation

- We did not specify the type of equilibrium in the definition
- Dominant

$u_i(s_i^*(\theta_i), s_{-i}(\theta_i), \theta_i) \geqq u_i(s_i'(\theta_i), s_{-i}(\theta_{-i}), \theta_i), \forall\, i, \forall\, \theta, \forall\, s_i' \neq s_i^*, \forall\, s_{-i}$

- Nash

$u_i(s_i^*(\theta_i), s^*_{-i}(\theta_{-i}), \theta_i) \geqq u_i(s_i'(\theta_i), s^*_{-i}(\theta_{-i}), \theta_i), \forall\, i, \forall\, \theta, \forall\, s_i' \neq s_i^*$

- Bayes-Nash

$E[u_i(s_i^*(\theta_i), s^*_{-i}(\theta_{-i}), \theta_i)] \geqq E[u_i(s_i'(\theta_i), s^*_{-i}(\theta_{-i}), \theta_i)], \forall\, i, \forall\, \theta, \forall\, s_i' \neq s_i^*$

Direct Mechanisms

- Recall that a mechanism specifies the strategy sets of the agents
  - These sets can contain complex strategies
- Direct mechanisms:
  - Mechanism in which $S_i = \Theta_i$ for all i, and $g(\theta) = f(\theta)$ for all $\theta \in \Theta_1 \times \ldots \times \Theta_n$
- Incentive-compatible:
  - A direct mechanism is incentive-compatible if it has an equilibrium s^* where $s^*_i(\theta_i) = \theta_i$ for all $\theta_i \in \Theta_i$ and all i
  - (truth telling by all agents is an equilibrium)
  - Called strategy-proof if truth telling by all agents leads to dominant-strategy equilibrium

Dominant Strategy Implementation

- Is a certain social choice function implementable in dominant strategies?
  - In principle we would need to consider all possible mechanisms

- Revelation Principle (for Dom Strategies)
  - Suppose there exists a mechanism $M=(S_1,\ldots,S_n,g(.))$ that implements social choice function f() in dominant strategies. Then there is a direct strategy-proof mechanism, M', which also implements f().

Revelation Principle

- "The computations that go on within the mind of any bidder in the nondirect mechanism are shifted to become part of the mechanism in the direct mechanism" [McAfee&McMillian 87]

- Consider the incentive-compatible direct-revelation implementation of an English auction (open-bid)

McAfee, R., & McMillan, J. Auctions with entry.
Economics Letters, 23, 343-347. **1987**.

Revelation Principle: Proof

- $M=(S_1,\ldots,S_n,g())$ implements SCF $f()$ in dom str.
  - Construct direct mechanism $M'=(\Theta^n,f(\theta))$
  - By contradiction, assume

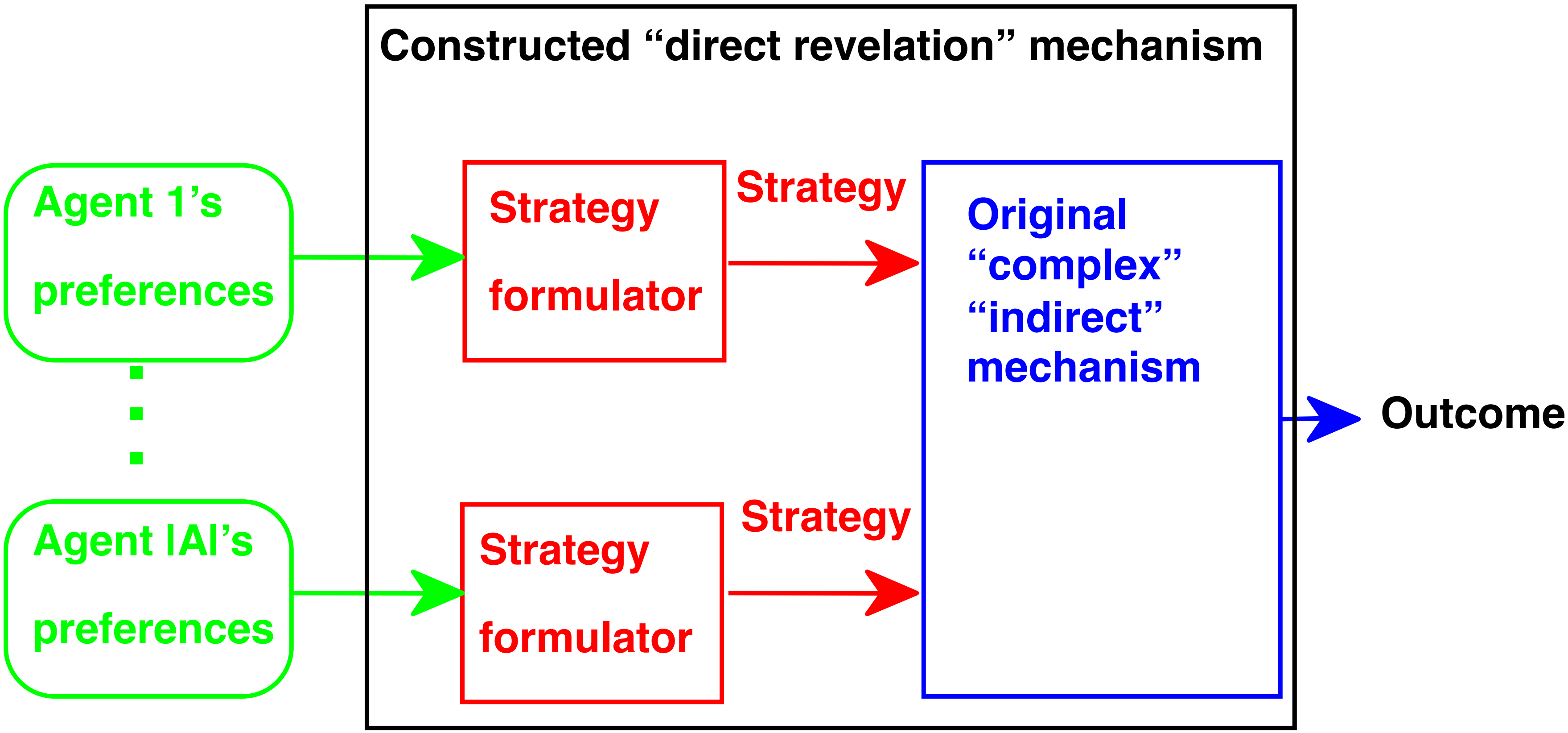
  $\exists\ \theta_i' \neq \theta_i$ s.t. $u_i(f(\theta_i',\theta_{-i}),\theta_i) > u_i(f(\theta_i,\theta_{-i}),\theta_i)$

  for some $\theta_i' \neq \theta_i$, some θ_{-i}.
  - But, because $f(\theta)=g(s^*(\theta))$, this entails

  $u_i(g(s_i^*(\theta_i'),s_{-i}^*(\theta_{-i})),\theta_i) > u_i(g(s^*(\theta_i),s^*(\theta_{-i})),\theta_i)$

  Which contradicts the fact that s^* is a dominant-strategy equilibrium in M

Revelation Principle: Intuition



Theoretical Implications

Literal interpretation: Need only study direct mechanisms

- This is a smaller space of mechanisms
- Negative results: If no direct mechanism can implement SCF f() then no mechanism can do it

- Analysis tool:
  - Best direct mechanism gives us an upper bound on what we can achieve with an indirect mechanism
  - Analyze all direct mechanisms and choose the best one

Practical Implications

- Incentive-compatibility is “free” from an implementation perspective
- BUT!!!
  - A lot of mechanisms used in practice are not direct and incentive-compatible
  - Maybe there are some issues that are being ignored here

Quick review

- We now know
  - What a mechanism is
  - What it means for a SCF to be dominant strategy implementable
  - If a SCF is implementable in dominant strategies then it can be implemented by a direct incentive-compatible mechanism
- We do not know
  - What types of SCF are dominant strategy implementable

Gibbard-Satterthwaite (G-S) Thm

- Assume
  - $\mathbf{O}$ is finite and $|\mathbf{O}| \geq 3$
  - Each $o \in \mathbf{O}$ can be achieved by social choice function $f()$ for some θ

Then:

$f()$ is truthfully implementable in dominant strategies (i.e., strategy-proof) if and only if $f()$ is dictatorial

Circumventing G-S

- Use a weaker equilibrium concept
  - Nash, Bayes-Nash
- Design mechanisms where computing a beneficial manipulation is hard
  - Many voting mechanisms are NP-hard to manipulate (or can be made NP-hard with small "tweaks") [Bartholdi, Tovey, Trick 89] [Conitzer, Sandholm 03]
- Randomization
- Agents' preferences have special structure

Quasi-Linear Preferences

- Outcome $o=(x,t_1,\ldots,t_n)$
  - x is a "project choice" and $t_i \in \mathbf{R}$ are transfers (money)
- Utility function of agent i
  - $u_i(o,\theta_i)=u_i((x,t_1,\ldots,t_n),\theta_i)=v_i(x,\theta_i)-t_i$
- Quasi-linear mechanism: $M=(S_1,\ldots,S_n,g(.))$ where $g(.)=(x(.),t_1(.),\ldots,t_n(.))$

Example:

- x="joint pool built" or "not",
- m_i = \$= mechanism addendum
  - E.g., equal sharing of construction cost: $-c / |A|$,
  - $v_i(x) = w_i(x) - c / |A|$
  - $u_i = v_i(x) + m_i$

Social choice functions and quasi-linear settings

- SCF is efficient if for all types $\theta=(\theta_1,\ldots,\theta_n)$
  - $\sum^n_{i=1} v_i(x(\theta),\theta_i) \geq \sum^n_{i=1} v_i(x'(\theta),\theta_i) \;\; \forall\, x'(\theta)$
  - Aka social welfare maximizing

- SCF is budget-balanced (BB) if
  - $\sum^n_{i=1} t_i(\theta)=0$

  – Weakly budget-balanced if

    $\sum^n_{i=1} t_i(\theta)\geq 0$

Groves Mechanisms [Groves 1973]

- A Groves mechanism, $M=(S_1,\ldots,S_n, (x,t_1,\ldots,t_n))$ is defined by
  - Choice rule $x^*(\theta')=\text{argmax}_x \sum_i v_i(x,\theta_i')$
  - Transfer rules
    - $t_i(\theta')=h_i(\theta_{-i}')-\sum_{j\neq i} v_j(x^*(\theta'),\theta'_j)$

where $h_i(.)$ is an (arbitrary) function that does not depend on the reported type θ_i' of agent i

Groves Mechanisms

- Thm: Groves mechanisms are strategy-proof and efficient (We have gotten around Gibbard-Satterthwaite!)

  Proof:

  Agent i's utility for strategy θ_i', given θ_{-i}' from agents $j \neq i$ is

  $u_i(\theta_i') = v_i(x^*(\theta'), \theta_i) - t_i(\theta')$

  $= v_i(x^*(\theta'), \theta_i) + \sum_{j \neq i} v_j(x^*(\theta'), \theta'_j) - h_i(\theta'_{-i})$

  Ignore $h_i(\theta_{-i})$. Notice that

  $x^*(\theta') = \operatorname{argmax} \sum_i v_i(x, \theta'_i)$

  i.e., it maximizes the sum of reported values.

  Therefore, agent i should announce $\theta_i' = \theta_i$ to maximize its own payoff

- Thm: Groves mechanisms are unique (up to $h_i(\theta_{-i})$)

VCG Mechanism
(aka Clarke tax mechanism, aka Pivotal mechanism)

- Def: Implement efficient outcome,

$$x^*=\text{argmax}_x \sum_i v_i(x,\theta_i')$$

Compute transfers

$$t_i(\theta')=\sum_{j\neq i} v_j(x^{-i},\theta'_j) - \sum_{j\neq i} v_j(x^*, \theta_i')$$

Where $x^{-i}=\text{argmax}_x \sum_{j\neq i} v_j(x,\theta_j')$

$\boxed{\text{VCGs are efficient and strategy-proof}}$

Agent's equilibrium utility is:

$$u_i(x^*,t_i,\theta_i)=v_i(x^*,\theta_i)-[\sum_{j\neq i} v_j(x^{-i},\theta_j) - \sum_{j\neq i} v_j(x^*,\theta_j)]$$

$$= \sum_j v_j(x^*,\theta_j) - \sum_{j\neq i} v_j(x^{-i},\theta_j)$$

= marginal contribution to the welfare of the system

Vickrey Auction

- Highest bidder gets item, and pays second highest amount
- Also a VCG mechanism
  - Allocation rule: Get item if $b_i = \max_i[b_j]$
  - Payment rule: Every agent pays

$$t_i(\theta_i') = \sum_{j \neq i} v_j(x^{-i}, \theta'_j) - \sum_{j \neq i} v_j(x^*, \theta_i')$$

$\max_{j \neq i}[b_j]$

$\max_{j \neq i}[b_j]$ if i is not the highest bidder,

0 if it is

Example: Building a pool

- The cost of building the pool is \$300
- If together all agents think the pool's value is more than \$300, then it will be built
- Clarke Mechanism:
  - Each agent announces their value, v_i
  - If $\sum v_i \geq 300$ then it is built
  - Payments $t_i(\theta_i') = \sum_{j \neq i} v_j(x^{-i}, \theta'_j) - \sum_{j \neq i} v_j(x^*, \theta_i')$ if built, 0 otherwise

v1=50, v2=50, v3=250

Pool should be built

t_1=(250+50)-(250+50)=0

t_2=(250+50)-(250+50)=0

t_3=(0)-(100)=-100

Not budget balanced

Web Mining Agents

- Task: Mine a certain number of books
- Agent pays for opportunity to do that if, for good results, agent gets high reward (maybe from sb else)
- Idea: Run an auction for bundles of books/reports/articles/papers to analyze

Implementation in Bayes-Nash equilibrium

- Goal is to design the rules of the game (aka mechanism) so that in **Bayes-Nash** equilibrium (s_1, …, s_n), the outcome of the game is $f(\theta_1,\ldots,\theta_n)$

- Weaker requirement than dominant strategy implementation
  - An agent's best response strategy may depend on others' strategies
    - Agents may benefit from counterspeculating
  - Can accomplish more than under dominant strategy implementation
    - E.g., budget balance & Pareto efficiency (social welfare maximization) under quasilinear preferences …

- There is also a mechanism for this setting:
  - D'AGVA mechanism [d'Aspremont & Gerard-Varet 79; Arrow 79]

Participation Constraints

- Agents cannot be forced to participate in a mechanism
  - It must be in their own best interest
- A mechanism is **individually rational** (IR) if an agent's (expected) utility from participating is (weakly) better than what it could get by not participating

Participation Constraints

- Let $u_i^*(\theta_i)$ be an agent's utility if it does not participate and has type θ_i
- Ex ante IR: An agent must decide to participate before it knows its own type
    - $E_{\theta \in \Theta}[u_i(f(\theta),\theta_i)] \geq E_{\theta_i \in \Theta_i}[u_i^*(\theta_i)]$
- Interim IR: An agent decides whether to participate once it knows its own type, but no other agent's type
    - $E_{\theta_{-i} \in \Theta_{-i}}[u_i(f(\theta_i,\theta_{-i}),\theta_i)] \geq u_i^*(\theta_i)$
- Ex post IR: An agent decides whether to participate after it knows everyone's types (after the mechanism has completed)
    - $u_i(f(\theta),\theta_i) \geq u_i^*(\theta_i)$

Quick Review

- Gibbard-Satterthwaite
  - Impossible to get non-dictatorial mechanisms if using dominant strategy implementation and general preferences
- Groves
  - Possible to get dominant strategy implementation with quasi-linear utilities
    - Efficient
- Clarke (or VCG)
  - Possible to get dominant strat implementation with quasi-linear utilities
    - Efficient, interim IR
- D'AGVA
  - Possible to get Bayesian-Nash implementation with quasi-linear utilities
    - Efficient, budget balanced, ex ante IR

Other mechanisms

- We know what to do with
  - Voting
  - Auctions
  - Public projects

- Are there any other "markets" that are interesting?

Bilateral Trade (e.g., B2B)

- Heart of any exchange
- 2 agents (one buyer, one seller), quasi-linear utilities
- Each agent knows its own value, but not the other’s
- Probability distributions are common knowledge

- Want a mechanism that is
  - Ex post budget balanced
  - Ex post Pareto efficient: exchange to occur if $v_b > v_s$
  - (Interim) IR: Higher expected utility from participating than by not participating

Myerson–Satterthwaite Thm

- **Thm**: In the bilateral trading problem, no mechanism can implement an ex–post BB, ex post efficient, and interim IR social choice function (even in Bayes–Nash equilibrium).

Does market design matter?

- You often here “The market will take care of “it”, if allowed to.”
- Myerson–Satterthwaite shows that under reasonable assumptions, the market will **NOT** take care of efficient allocation

Acknowledgements

Paper: Automated Mechanism Design

By Tuomas Sandholm

Presented by Dimitri Mostinski
November 17, 2004

Sandholm T. Automated Mechanism Design: A New Application Area for Search Algorithms. In: Rossi F. (eds) Principles and Practice of Constraint Programming – CP 2003. LNCS, vol 2833. **2003**.

Problems with Manual MD

- The most famous and most broadly applicable general mechanisms, VCG and dAGVA, only maximize social welfare

- The most common mechanisms assume that the agents have quasilinear preferences $u_i(o; t_1, .. ,t_N) = v_i(o) - t_i$

Impossibility results:

- "No mechanism *works* across a *class* of settings" for different definitions of "works" and different classes of settings
  - E.g., Gibbard-Satterthwaite theorem

Automatic Mechanism Design (AMD)

- Mechanism is computationally created for the specic problem instance at hand
  - Too costly in most settings w/o automation
- Circumvent impossibility results

AMD formalism

- An automatic mechanism design setting is
  - A finite set of outcomes O
  - A finite set of N agents
  - For each agent I
    - A finite set of types Θ_i
    - A probability distribution γ_i over Θ_i
    - A utility function $u_i : \Theta_i \times O \rightarrow \mathbb{R}$
    - An objective function whose expectation the designer wishes to maximize $g(o; t_1, \ldots, t_N)$

More AMD formalism

- A mechanism consists of
  - An outcome selection function
    $o : \Theta_1 x .. x\ \Theta_N \rightarrow O$ if it is deterministic
  - A distribution selection function
    $p : \Theta_1 x .. x\ \Theta_N \rightarrow P(O)$ if it is randomized
  - For each agent i a payment selection function
    $\pi_i : \Theta_1 x .. x\ \Theta_N \rightarrow R$ if it involves payments

Individual Rationality

- In an AMD setting with an IR constraint there exists a fallback outcome o_0 such that for every agent i $u_i(\theta_i, o_0) = 0$

Incentive Compatibility

- The agents should never have an incentive to misreport their type
- Two most common *solution concepts* are
  - *implementation in dominant strategies*
    - Truth telling is the optimal strategy even if all other agents' types are known
  - *implementation in Bayesian Nash equilibrium*
    - Truth telling is the optimal strategy if other agents' types are not yet known, but they are assumed to be truthful

Formally the AMD problem

- Given
  - Automated mechanism design setting
  - An IR notion (ex interim, ex post, or none)
  - A solution concept (dominant strategies or Bayesian Nash equilibrium)
  - Possibility of payments and randomization
  - A target value G
- Determine
  - If there exists a mechanism of the specified type that satisfies both the IR notion and the solution concept, and gives an expected value of at least G for the objective.

Complexity results

- AMD is NP-hard (by reduction to MINSAT) if
  - Payments are not allowed
  - Payments are allowed but the designer is looking for something other than social welfare maximization
- AMD can be solved in (expected) polynomial time using randomized algorithm for LP problems

Conclusion: Some results of AMD

- It reinvented the Myerson auction which maximizes the seller's expected revenue in a 1-object auction
- It created expected revenue maximizing combinatorial auctions
- It created optimal mechanisms for a public good problem (deciding whether or not to build a bridge)
- … also for multiple goods