Computational Mechanism Design and Auctions

TARK X
Institute for Mathematical Sciences
National University of Singapore

David C. Parkes
parkes@eecs.harvard.edu
Harvard University
June 13, 2005

Mechanism Design (MD)

- Mechanisms: Protocols to impement desired system-wide outcomes in multi-agent systems despite the self-interest and private information of agents.

- Computational MD: the design of such mechanisms.
  - should be "truthful"
  - should be "efficiently computable"
  - should be "computationally feasible" for agents

- Auctions: mechanisms for resource allocation
  - typically "detail free," don't depend on distributional knowledge on types of agents.

- Start with a normative model of agent behavior.
- Design "rules of the game", e.g. to allocate resources or tasks efficiently in equilibrium.

- May also try to design for:
  - robust equilibrium
  - minimal information revelation
  - distributed computation
  - bounded-rational agents
  - adaptive agents

Example: Internet Auctions

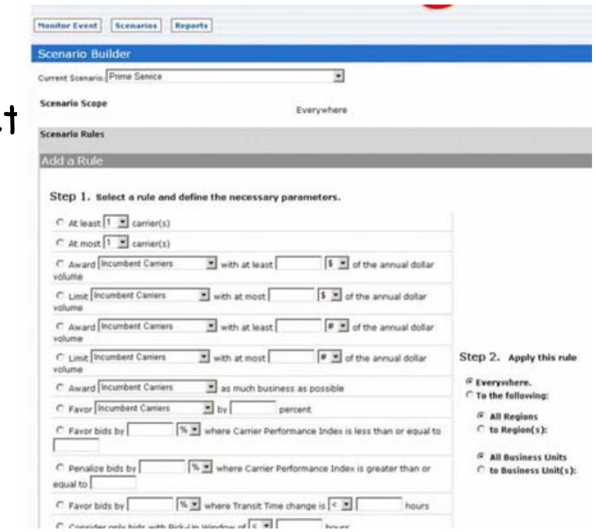
- eBay

Example: Ad Auctions

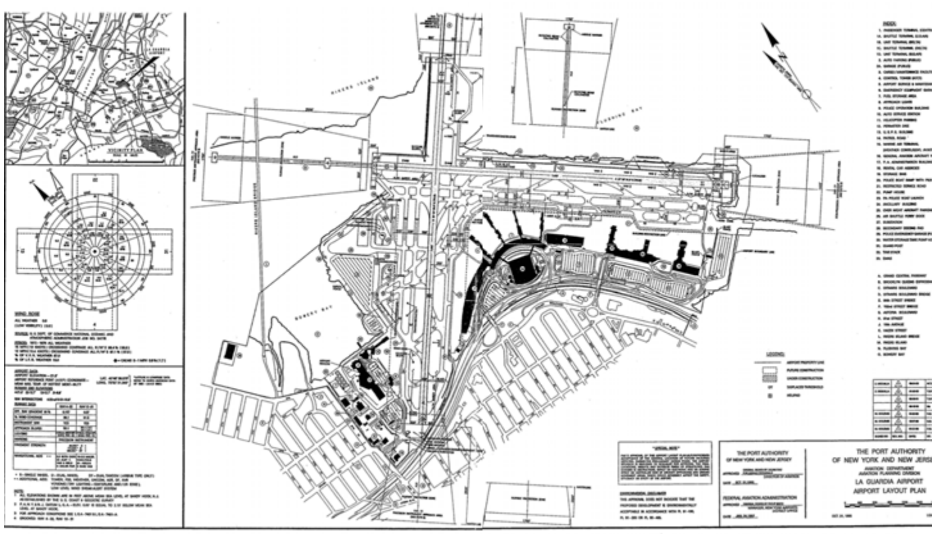
- Google

Example: Procurement Auctions

- CombineNet



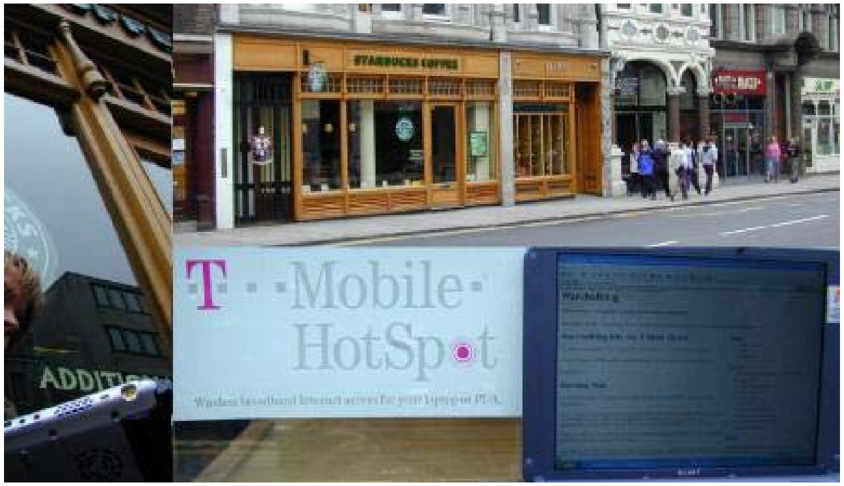
Example: LGA Take-off & Landing



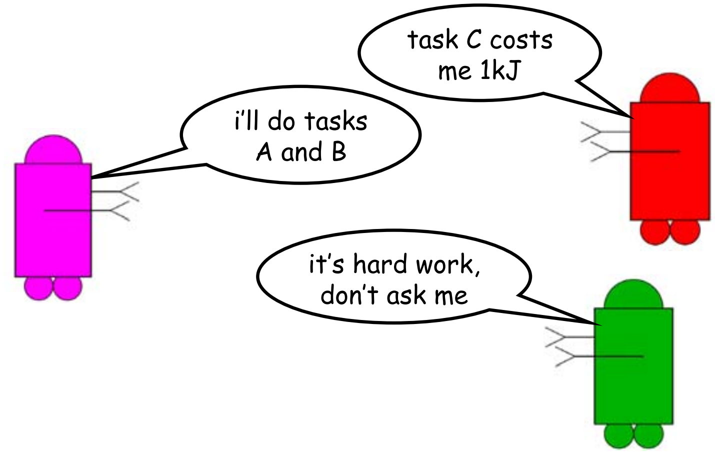
Example: Sensor Networks

- Intel Research Berkeley's 150-mote sensor network

Example: WiFi @ Starbucks



Example: MultiAgent Planning

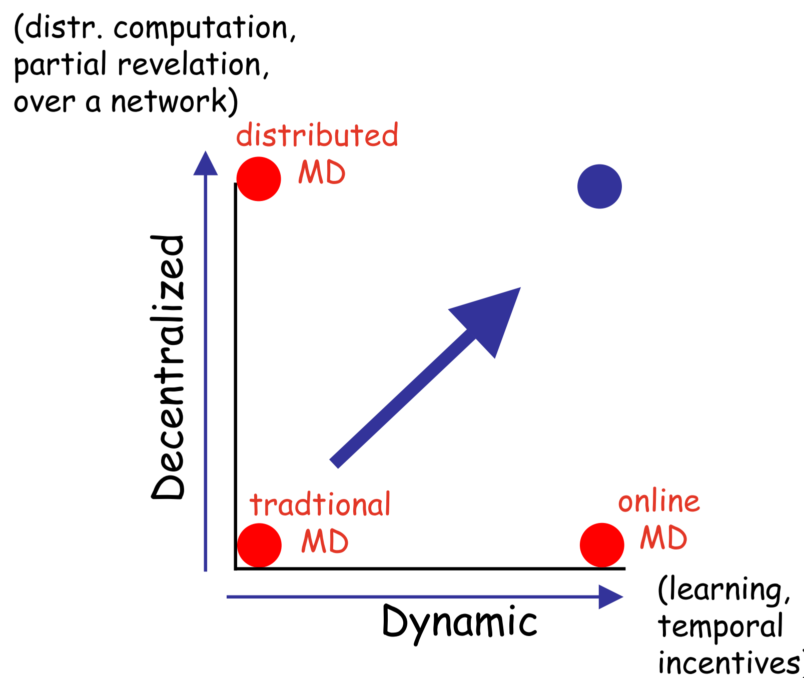


CS/Econ Analogy

(based on Feigenbaum)

• Agents are cooperative	• Agents are self-interested
• Main concern is computational and communication	• Main concern is incentives

Computational Mechanism Design:
- brings both together…



Outline: Tutorial

- Static & Centralized MD
  - algorithmic mechanism design
  - truthful characterizations
- Static & Decentralized MD
  - indirect mechanisms
  - ascending-price auctions
  - distributed implementations
- Dynamic & Centralized MD
  - online auctions, online MD
  - truthful characterizations
- Adaptive & Decentralized MD
  - uncertain rewards, learning

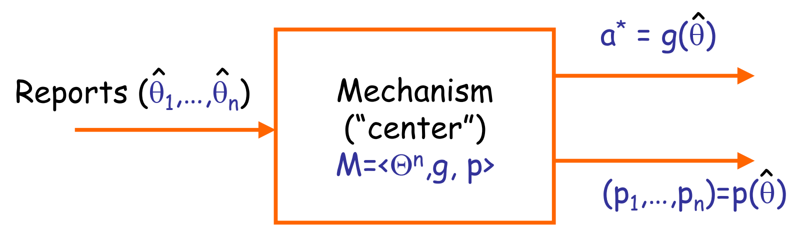
Part I:

Preliminaries
VCG
Truthfulness
AMD

Multi-agent System: Preliminaries

- Set of alternatives $A = \{a,b,\ldots\}$
- Agents $N = \{1,2,\ldots\}$, $|N|=n$
- Agent i has private information (type) $\theta_i \in \Theta_i$
  - e.g., value $v_i(a;\theta_i)$ for alternative $a \in A$
  - often times we'll just write $v_i(a)$
- Quasi-linear utility: $u_i(a,p)=v_i(a;\theta_i)-p$
  for alternative a at price p
  - no budget constraints
- **Goal**: implement a *social choice function* (scf),
  $scf(\theta) \in A$; for instance choose a^* to max $\sum_i v_i(a;\theta_i)$

Truthful Mechanisms



$g: \Theta^n \to A$ outcome rule
$p: \Theta^n \to R^n$ payment rule
Θ^n type space

Truthful reports, $\hat{\theta}_i = \theta_i$ in a **dominant-strategy** equilibrium.
Also called strategyproof.

Example: Second price auction

(Vickrey'61)

Value v_i. Agent i submits bid b_i, and receives utility:

$$u_i(b_1,\ldots,b_n) = \begin{cases} v_i - \max_{j\neq i} b_j, & \text{if } b_i > \max_{j\neq i} b_j \\ 0, & \text{otherwise} \end{cases}$$

Truthful: dominant strategy is to bid, $b^*_i(v_i)=v_i$
Auction is **efficient**.

Proof:
$p_i = \max_{j\neq i} b_j$. agent-independent.
will buy if and only if $b_i > p_i$
should report $b_i=v_i$

The Combinatorial Auction

- Goods G, $|G|=m$
- Alternatives:
  - allocations $S=(S_1,\ldots,S_n)$, with bundle $S_i \subseteq G$
  - feasible: $S_i \cap S_j = \emptyset$ for all agents i, j
- Values $v_i(S_i;\theta_i) \geq 0$ for bundles $S_i \subseteq G$
- Typical goal: $\max_S \sum_i v_i(S_i,\theta_i)$
- Applications: logistics, MBA course scheduling, wireless spectrum, school lunches in Chile, …

Computational Results

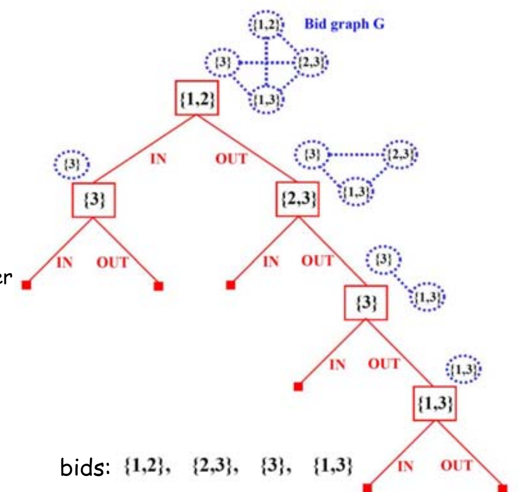
$\mathbf{WD_{XOR}}$: $\max_{x_i(S)} \sum_i v_i(S)\, x_i(S)$
s.t. $\sum_S x_i(S) \leq 1, \forall i$
$\sum_i \sum_{S: j\in S} x_i(S) \leq 1, \forall j$
$x_i(S) \in \{0,1\}$

- XOR bidding language: want at most one bundle
  - {(AB,\$10) xor (CD,\$5) xor (ABC,\$15)}
- NP-hard (MaxWeightSetPacking = WD for single-minded)
- Inapproximable, no better than $\min(l^{1-\varepsilon}, m^{1/2-\varepsilon})$ polytime-approx unless NP=ZPP (Hastad'99,Sandholm'02,Lehmann et al'02)
  - $m^{1/2}$ approx; greedy sort by $v_i(S)/(|S|^{1/2})$ (Lehmann et al.'02)
- No polynomial time approximation scheme (PTAS) unless P=NP (A_ε achieving $1+\varepsilon$ approx, poly-time for fixed ε) (Berman & Fujito'99, Lehmann et al.'05)
- Polynomial special cases exist for WD_{OR} (e.g. Rothkopf et al.'98)
  - {(AB,\$5) or (CD,\$10) or (CE,\$7)}
- restricted valuations: $OXS \subset GS \subset SM \subset XOS \subset CF$ (Lehmann et al.'03)
  - $\log(m)$-approx for CF (Dobzinski et al.05); $2-\varepsilon$ LB
  - $(e/e-1)$-approx for XOS (Dobzinski & Schapira'05); $1+1/2m$ LB (Nisan&Segal'03)

Practical WD Algorithms

(Fujishima et al'99, Sandholm'99, Sandholm et al.'01, Sandholm et al.02, Andersson et al.00, de Vries and Vohra'03)

- Systematic search
  - anytime algorithm
  - provable error bound
- Branch on bids
- LP-based admissible heuristics
- Branch & cut: (Nemhauser & Wolsey'99, Nemhauser'98)
  - cutting planes to strengthen formulations
- Branching heuristics



(Sandholm'05)

Truthfulness: The VCG Mechanism

(Vickrey 61, Clarke 71, Groves 73)

VCG mechanism:

- Collect $\theta=(\theta_1,\ldots,\theta_n)$ from agents.
- $g(\theta)$: Select $a^*\in A$ to maximize $\sum_i v_i(a;\theta_i)$
- $p_i(\theta)=p_{VCG,i}=\sum_{j\neq i} v_j(a^{-i};\theta_j)-\sum_{j\neq i} v_j(a^*;\theta_j)$, where a^{-i} solves $\max_{a\in A}\sum_{j\neq i} v_j(a;\theta_j)$

Theorem. The VCG mechanism is truthful and allocatively-efficient.

Example: Combinatorial Auction

- Buyer 3 wins, and pays 10-0=10.

bundles

agents	A	B	AB
1	5	0	5
2	0	5	5
3	0	0	12

- Buyers 1 and 2 win, and pay 7-5=2 each.

bundles

agents	A	B	AB
1	5	0	5
2	0	5	5
3	0	0	7

(writing $v_i(S,\theta)$ as $v_i(S)$)

- Consider **agent-independent** prices:

  $p_i(S) = V_{-i}(G) - V_{-i}(G\setminus S)$, for all i, all S

  where $V_K(G')=\max_{S\in Feas(G')}\sum_{i\in K} v_i(S_i)$

Proof:

- *First*, show that the efficient allocation S^* solves $\max_S v_i(S) - p_i(S)$, for all i
- $S^*_i \in \arg\max_S v_i(S) + v_{-i}(G\setminus S) - V_{-i}(G)$

- *Second*, show that $p_{VCG,i}=p_i(S^*_i)$
- $p_i(S^*_i) = V_{-i}(G)-V_{-i}(G\setminus S^*_i)$

  $= \sum_{j\neq i} v_j(a^{-i};\theta_j)-\sum_{j\neq i} v_j(a^*;\theta_j) = p_{vcg,i}$

VCG Mechanism

- Generalizes to implement affine-maximizers:

  $g(\theta) = \arg\max_a \sum_i c_i v_i(a,\theta_i) + c(a)$

  $p_{vcg,i}(\theta) = 1/c_i \{ \sum_{j\neq i} c_j v_j(a^{-i},\theta_j) + c(a^{-i}) - \sum_{j\neq i} c_j v_j(a^*,\theta_j) - c(a^*) \}$
- Universal, applies for all domains.
- Unique, only truthful mechanism for unrestricted preferences (K.Roberts'79)
- Unique, only truthful affine-maximizing mechanism for arbtitrarily-restricted preferences (Green&Laffont'77)
- Maximizes expected revenue across all ex post IR and efficient mechanisms (Krishna&Perry'98)

VCG may run at a deficit

- Trade of an item from agent 1 to agent 2
- Agent 1: $v_1 \in [0,1]$
- Agent 2: $v_2 \in [0,1]$
- Alternatives: {no-trade, trade}
- VCG mechanism:
  - receive bids b_1, b_2
  - if $b_2 > b_1$, then trade; and $p_{vcg,1}=0-b_2$, $p_{vcg,2}=b_1-0$
  - otherwise, no trade.

- **Example**: $v_1=0.3$, $v_2=0.6$
- Outcome: trade, $p_{vcg,1}=-0.6$ and $p_{vcg,2}=0.3$
- Budget deficit of $-0.6+0.3=-0.3$

- No-deficit + IR + efficient two-sided trading mechanism is impossible (Myerson & Satterthwaite'83)

Computational Issues

- For center: If used to solve NP-hard problems (e.g. CAs), easily loses truthfulness if substitute an approximation. (Nisan & Ronen'00)
- For agents: required to report complete valuation function (Parkes'01)
  - hard valuation problem
  - privacy
  - communication complexity
- Completely centralized

Example: Approximate VCG

(still NP hard, weighted set-packing problem…)

- Single-minded: type $\theta_i=\langle w_i,S_i\rangle$ s.t.

  $v_i(S;\theta_i)= w_i$, for all $S \supseteq S_i$

  $= 0$, otherwise
- Greedy approximation:
  - sort bids in order of decreasing $w_i / |S_i|$
  - allocate with greedy algorithm

E.g., Agent 1. $(A,10)$, Agent 2. $(AB,19)$, Agent 3. $(B,8)$

Implement $(A, \emptyset, B)$.

Payment by 1: 19 - 8 = 11 ← fails participation!

Payment by 2: 0 ← should overstate value!

Payment by 3: 10 - 10 = 0

Algorithmic Mechanism Design

(Lehmann et al.'99, Nisan & Ronen'00)

- Find truthful and tractable mechanisms $M=\langle\Theta^n,g,p\rangle$
- Still direct-revelation:
  - does not address agent complexity

Idea: Price-Based Mechanisms

(e.g. Segal 02, Bartal et al. 03, Lavi et al. 03, Yokoo 03, goes back earlier…)

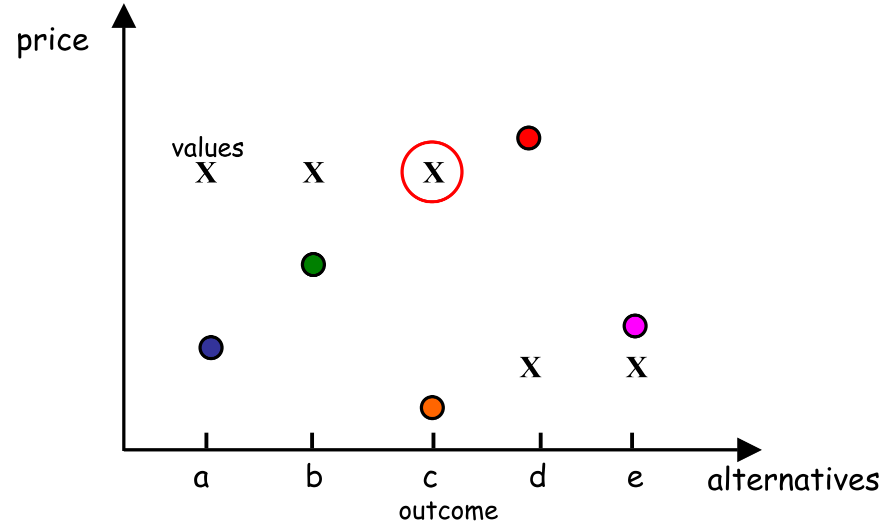
Theorem. Mechanism $M=\langle\Theta^n,g,p\rangle$ is truthful if and only if exists an agent-independent price function
$\pi_i: A \times \Theta_{-i} \rightarrow R$ s.t.

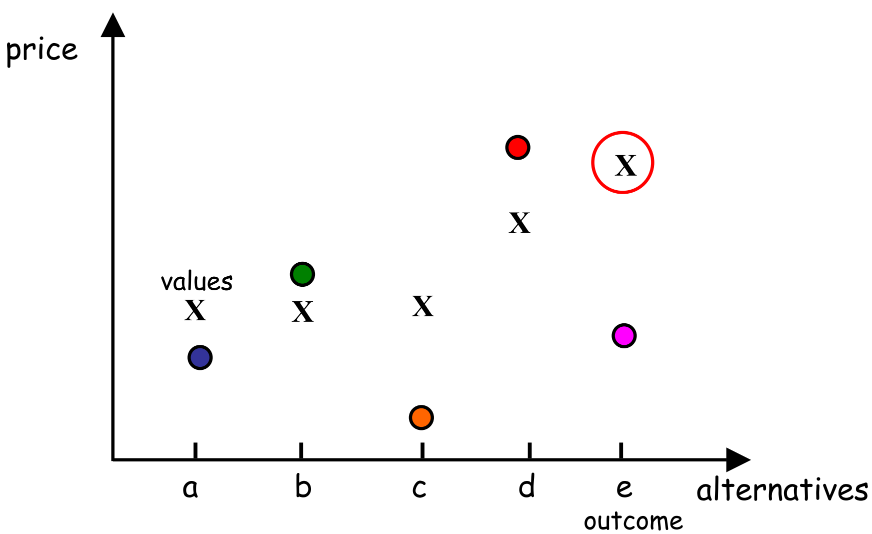
1) the payment $p_i(\theta)=\pi_i(a,\theta_{-i})$, when $a=g(\theta)\in A$ is selected.

2) "admissible" $a=g(\theta)\in \arg\max_{a\in A}\{v_i(a;\theta_i)-\pi_i(a,\theta_{-i})\}$, for all i, all θ.

sufficient: Agent i cannot change prices π_i, and maximizes utility $u_i(a,\pi_i(a,\theta_{-i}))$ by reporting true θ_i

$\Rightarrow$ try to characterize allocation rules for which there exist admissible agent-independent prices.





Every truthful mechanism must be price-based

Proof. Construct $\pi_i(a,\theta_{-i})=p_i(\theta'_i,\theta_{-i})$ when $g(\theta'_i,\theta_{-i})=a$ for some θ'_i, and $\pi_i(a,\theta_{-i})=\infty$ otherwise.

- *Agent-independent*: suppose some θ, some $\theta'_i\neq\theta_i$, with $g(\theta)=g(\theta'_i,\theta_{-i})=a$, but $p_i(\theta)\neq p_i(\theta'_i,\theta_{-i})$. w.l.o.g., $p_i(\theta)>p_i(\theta'_i,\theta_{-i})$, and should declare θ'_i. Contradiction w/ truthfulness.
- *Admissible*: suppose some θ, with $g(\theta)=a$, and $v_i(a,\theta_i)-\pi_i(a,\theta_{-i}) < v_i(b,\theta_i)-\pi_i(b,\theta_{-i})$ for $b\neq a$. Agent should declare θ'_i, contradiction w/ truthfulness.

Example: Single-Minded CAs

(Lehmann, O'Callaghan & Shoham 2003)

- Allocate with greedy scheme, in order $w_i/|S_i|$
- Winner pays $|S_i| \cdot \{w_j/|S_j|\}$, where bid j is the first bid that would win without the bid $\langle w_i, S_i \rangle$

E.g., Agent 1. (A,10), Agent 2. (AB,19), Agent 3. (B,8)

Implement (A, $\emptyset$, B).

Payment by 1: $1 \times (19/2) = 9.5$

Payment by 2: 0

Payment by 3: 0

Proof:

- Prices $\pi_i(S_i,\theta_{-i}) = \min\{ w'_i \in R : \theta'_i = \langle w'_i, S_i \rangle, g_i(\theta'_i,\theta_{-i}) = S_i\}$
- *Winner*: $\pi_i(S_i,\theta_{-i}) = |S_i| \cdot (w_j/|S_j|) \leq w_i$, where j is displaced bid, since $w_i/|S_i| \geq w_j/|S_j|$
- *Loser*: $\pi_i(S_i,\theta_{-i}) > w_i$, since greedy algorithm is **monotonic** and would allocate if $w_i \geq \pi_i(S_i,\theta_{-i})$.

Key Property: Monotonicity

- Bid-monotonic: If bid $\langle w_i, S_i \rangle$ wins, then bid $\langle v_i, T_i \rangle$ for $v_i \geq w_i$ and $T_i \subseteq S_i$ will also win.
- All single-minded greedy allocation rules $g(\cdot)$ that sort by $w_i/|S_i|^k$ for $k \geq 0$ are monotonic.
- Monotonicity of allocation rule is necessary & sufficient for existence of admissible prices for single-minded allocation problems.

- "Critical value" **payment rule**:

$$p_i(\theta) = \pi_i(S_i,\theta_{-i}) = \min\{ w'_i : \theta'_i = \langle w'_i, S_i \rangle, g_i(\theta'_i,\theta_{-i}) = S_i\},$$

Additional Results in AMD

- Multi-item CAs:
  - WDP_{XOR}
  - each bid for a small number of items (determines k)
  - $2(1+r^{k-1}/k)$-approx, for constant $r>1$ and $k<1$
  - (Bartal,Gonen & Nisan.'03)
- Digital goods:
  - Consensus revenue estimate (CORE)
  - random sampling threshold auctions (RSOT)
  - revenue-competitive results
  - (Goldberg, Hartline et al.'01,'03; also Segal'02)
- Building on VCG-based Maximal-in-range (Nisan & Ronen'00):
  - Anytime SP (Schoenebeck & Parkes'04)
  - $m^{1/2}$-approx for CF special case of CAs (Dobzinski & Schapria'05)
- Handling Budget Constraints
  - agent type: value + budget
  - Using sampling approach (Borgs, Immorlica et al.'05)

Part II:

More general characterizations

Indirect mechanisms

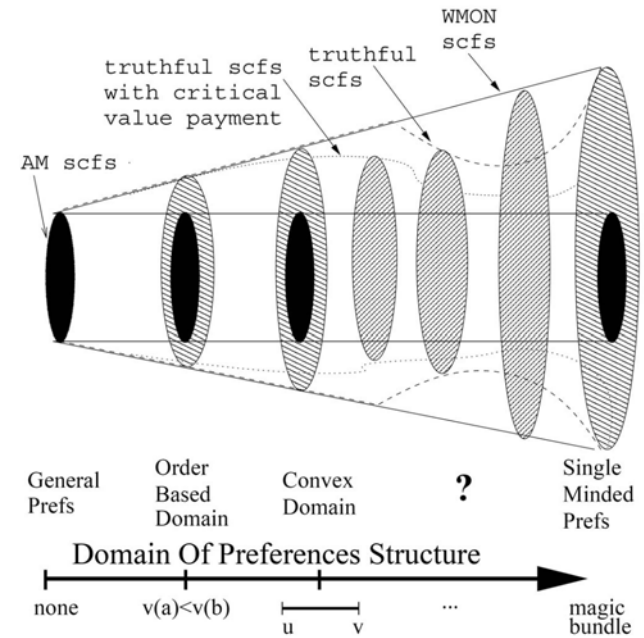
Seeking more general characterizations

- W-MON: $g(v_i,v_{-i})=a$, $g(w_i,v_{-i})=b \Rightarrow w_i(b)-w_i(a) \geq v_i(b)-v_i(a)$
  - "cannot change from a to b unless value on b increases."

- **Necessary** (truthful $\Rightarrow$ WMON) (Rochet'87)
- Suppose $g(v_i,v_{-i})=a$ and $g(w_i,v_{-i})=b$.
- By truthful, $v_i(a)-\pi_i(a,v_{-i}) \geq v_i(b)-\pi_i(b,v_{-i})$ and
  $w_i(b)-\pi_i(b,v_{-i}) \geq w_i(a)-\pi_i(a,v_{-i})$
- Combining, $w_i(b)-w_i(a) \geq v_i(b)-v_i(a)$.

- **Sufficient** for single-parameter domains (e.g. single-minded CAs). Where else?

(Rochet'77,Lavi et al.'03, Roberts'79, Green&Laffont'77,Lehmann et al.02,Saks&Yu'05)



(Constantin & Parkes'05)

Order-based Domains

(Lavi, Mu'alem and Nisan'03)

Domain of types Θ defined in terms of:
- constraints: $R_i(a,b) \in \{=, <, \leq, >, \geq\}$
- null outcomes: $Null \subset A$

Then: $\theta_i \in \Theta_i$ if and only if:

$v_i(a;\theta_i)=v_i(b;\theta_i)$, $\forall a, b$ s.t. $R_i(a,b)$ = "="

$v_i(a;\theta_i)<v_i(b;\theta_i)$, $\forall a, b$ s.t. $R_i(a,b)$ = "<"

...

$v_i(a;\theta_i)=0$, $\forall\, a \in Null$

Includes: CAs, multi-unit auctions, contiguous preferences, unrestricted preferences.

Example: CAs

Alternatives $a \in A$ define allocations

(no externalities)

$R_i(a,b)$ = "=" for all a, b with $S_i^a = S_i^b$

(normalizaton)

$a \in Null$ for all a with $S_i^a = \emptyset$

(free-disposal)

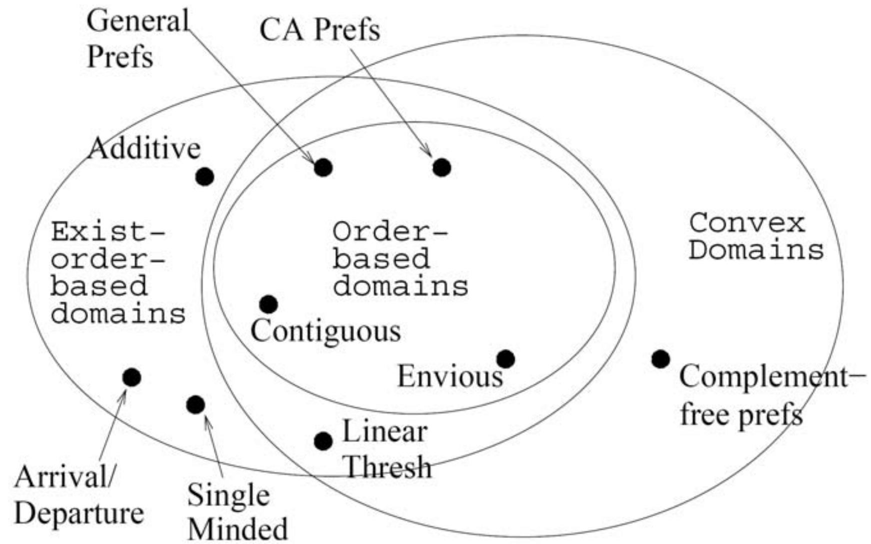
$R_i(a,b)$ = "$\leq$" for all a, b with $S_i^a \subseteq S_i^b$

some results

- Lavi et al.'03: order-based + WMON ⇒ truthful
- Saks & Yu'05: convex + WMON ⇒ truthful
- Gui et al.'04: graph-theoretic characterizations for sufficiency
- Lavi et al.'04: IIA + order-based + truthful ⇒ affine-maximizer

Gaps in characterization

(Constantin & Parkes'05)



Also: bounded-XOR, CF, attribute-based,...

Directions for Characterizations

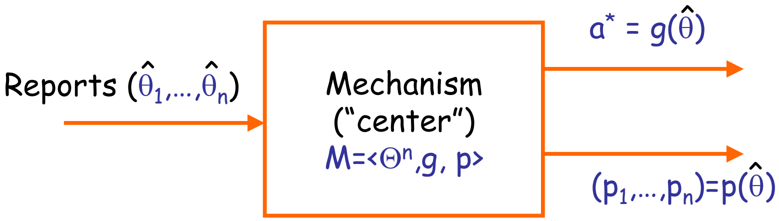
(Constantin & Parkes'05)

- +universal
- +natural ("critical value") price functions
- +additional structure
  - exist-order-based
  - attribute-based
  - multi-order based
- +algorithmically meaningful
  - i.e. would like sufficient conditions that map to algorithmic properties

Outline

- Static & Centralized MD
- Static & Decentralized MD
  - indirect mechanisms
  - ascending-price mechanisms
  - distributed implementations
- Dynamic & Centralized MD
- Adaptive & Decentralized MD

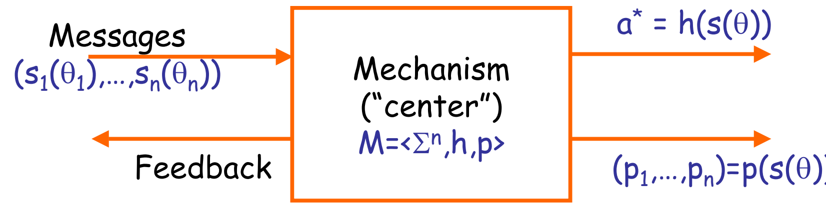
Direct Mechanisms



$g: \Theta^n \rightarrow A$ outcome rule
$p: \Theta^n \rightarrow R^n$ payment rule
Θ^n type space

Seek M for which truth-revelation is a DSE.

Indirect Mechanisms



$h: \Sigma^n \rightarrow A$ outcome rule
$p: \Sigma^n \rightarrow R^n$ payment rule
Σ^n strategy space
strategy $s : \Theta_i \Rightarrow \Sigma_i$

Seek M for which exists some $s^*=(s^*_1,\dots,s^*_n)$ that is an **ex post Nash equilibrium**.

ex post Nash

- ex post Nash: s^*_i is best-response whatever the type of other agents:
  $u_i(s^*_i(\theta_i),s^*_{-i}(\theta_{-i});\theta_i)$
  $\geq u_i(s'_i(\theta_i),s^*_{-i}(\theta_{-i});\theta_i)$, $\forall\theta_{-i}, \forall\theta_i, \forall i, \forall s'_i$

$\boxed{\text{DSE} \subseteq \text{ex post}}$

ex post Nash requires that other agents ($\neq i$) play the equilibrium strategy

still allows an agent to have no information about private types of other agents.

Example: open out-cry, ascending-price single-item auction

Revelation Principle

- **Theorem**: Any scf that can be implemented in an ex post Nash equilibrium in an *indirect* mechanism can be implemented in a DSE in a *direct* mechanism.

- **Proof** (sketch). Via a reduction. If there is some complex mechanism M with equilibrium s^*, then construct a new *direct* mechanism M' in which the center commits to simulate strategy s^* and rules $\langle h,p\rangle$ of M. Truthful reporting is an equilibrium in M' because s^* is an equilibrium in M.

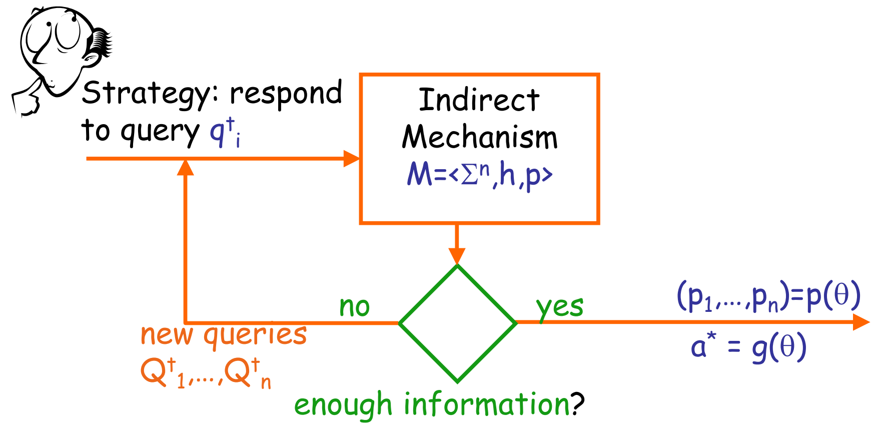
- Why worry about indirect mechanisms?

Computational Advantages of Indirect Mechanisms

(Parkes'99,Parkes'01,Contizer&Sandholm'02,Feigenbaum & Shenker'02)

- Less information revelation (privacy)
  - e.g., the winner does not reveal v_i, and other agents that bid in period t reveal $v_i \geq p^t$
- Avoids unecessary valuation effort
  - e.g., the winner does not need to know exact value, only that $v_i \geq p^T$ in final round T
  - e.g., the losers do not need to know exact value, only that $v_i < p^t$ in drop-out round
- Can distribute computation:
  - e.g., ask agents to submit best-responses in each round; can perform useful computation.

Incremental-Revelation Mechanisms



- Example queries: value(a)?, demand(p)?, is $v_i(a)>v_i(b)$?, …
- **Consistency + VCG outcome $\Rightarrow$ ex post Nash**

Truthfulness via VCG

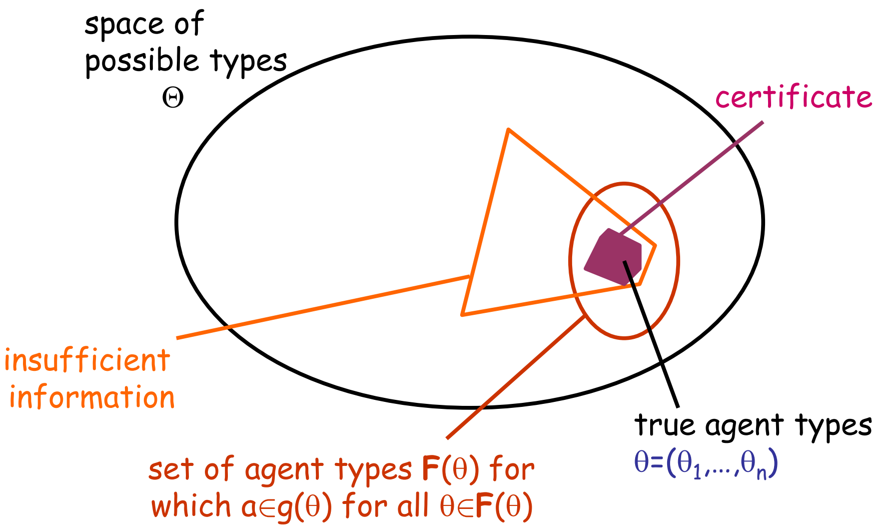
- Let s^* denote the truthful strategy.
- Say M is **consistent** if $s'_i \in \Sigma$, then for all θ_i then $\exists\, \theta'_i$ s.t. $s^*_i(\theta'_i)$ is identical to $s'_i(\theta_i)$.
  - use "activity rules", e.g. no jump bids, no re-entry once dropped out,…
- **Theorem**: Any consistent mechanism that implements the VCG outcome with s^* is truthful in ex post Nash equilibrium. (Gul & Stacchetti'03)
- **Proof** (sketch): Fix s^*_{-i}, fix v_{-i}, consider some v_i. show that any $s'_i \neq s^*_i$ is equivalent to s^*_i for some $v'_i \neq v_i$. Get ex post Nash by appeal to VCG.

Static & Decentralized MD

- Center + Incremental-revelation
  - Characterization of minimal information requirements to implement scfs
  - Design of incremental-revelation mechanisms
  - Price-based, computational-learning theory based
- Distributed computation
  - Good "network complexity"
  - Bring computation and information revelation into an equilibrium

Information Certificates

(Parkes 02)



Characterizations of Minimal information to determine efficient allocation in CAs

(Parkes 02; Segal & Nisan 03)

Price $p_i(S) \geq 0$ for bundles $S \subseteq G$.

Prices $(p_1,\ldots,p_n)$ are CE prices if and only if the efficient allocation S^* satisfies:

(1) $S^*_i \in \arg\max_{S_i} \{v_i(S_i;\theta_i) - p_i(S_i)\}, \ \forall i$

(2) $S^* \in \arg\max_{S_1,\ldots,S_n} \sum_i p_i(S_i)$

Theorem. Any mechanism that implements the efficient allocation also elicits enough information to determine CE prices.

(Also *sufficient*: an allocation S satisfying (1) and (2) for some prices p is efficient.)

Ascending-Price CAs

- Large literature on ascending-price CAs
- Maintain prices p^t, allocation x^t
- Seek CE prices ⇒ efficient allocation

- Collect best-response sets $BR_i^t \subseteq 2^G$
- Solve WD to maximize revenue given bids BR_i^t
  - chose an allocation from bids that maximizes total revenue to auctioneer at current prices
- Increment prices
- Terminate when all agents still bidding receive a bundle in allocation. Typically, adopt final prices as payments.

Minimal VCG Certificates

(Lahaie, Constantin & Parkes'05)

Prices $(p_1,\ldots,p_n)$ are Universal CE prices if and only if:

(1) prices are CE for main economy $E(N)$

(2) prices are CE for marginal economies $E(N\backslash 1),\ldots, E(N\backslash n)$

Example: $v_1 = 10$. $v_2 = 6$. $v_3 = 4$. Price $6 \leq p \leq 10$ is a CE price. But only $4 \leq p \leq 6$ is a CE price in economy $\{2,3\}$. UCE price, $p_{uce}=6$.

Theorem. Any mechanism that implements the outcome of the VCG mechanism must elicit enough information to determine UCE prices.

(Also *sufficient*: an allocation S satisfying (1) prices satisfying (1) and (2), then $p_{vcg,i} = p_i(S_i) - \{\Pi^s(N) - \Pi^s(N\backslash i)\}$. (Parkes&Mishra'04))

Linear-Programming Based Design
(de Vries et al.'04, Parkes&Ungar'00)

- Formulate an LP for the allocation problem.
- Auctions provide Primal-dual/subgradient algorithms.
- Maintain feasible primal and dual solutions: allocation & prices
- Increase prices based on losing bids.
- Terminate when allocation maximizes payoff for all bidders.
- Primal & Dual are optimal:
  (P) efficient allocation
  (D) CE prices
- Also get UCE, then myopic best-response is ex post Nash…

uQCE-invariant Auctions
(Mishra & Parkes'05)

- In round t:
  - collect demand sets at prices p^t
  - if p^t are UCE, then stop
  - else, select adjusted buyers $U^t \subseteq B^+(p^t)$
  - $p^{t+1}_i(S) := p^t_i(S)+1$ for $i \in U^t$, $S \in D_i(p^t)$
- On termination,
  - implement final allocation
  - payments $p^T_i(X_i) - \{\Pi^s(N) - \Pi^s(N \setminus i)\}$

- **Claim**: maintain universal-Quasi-CE prices in each round
  - prices s.t. the seller can maximize revenue at prices in the set of allocations consistent with demand sets
  - for every economy, main & marginal

⇒ terminate with UCE prices, … VCG outcome.

Example: iBundle Extend & Adjust
(Parkes & Ungar'03,Mishra & Parkes'05)

iBEA:
- maintain non-linear and non-anonymous prices $p^t_i(S)$
- choose "pivot" economy that is not yet in CE
- solve WD, increase prices on bundles from losing bidders

- Example:

1: A,3* B,0 AB,3
2: A,0 B,6* AB,6
3: A,0 B,2 AB,4

$p_{vcg,1}=6-6=0$
$p_{vcg,3}=5-3=2$

	Buyer 1			Buyer 2			Buyer 3			Seller revenue in main and marginal economies
	$\{1\}$	$\{2\}$	$\{1,2\}$	$\{1\}$	$\{2\}$	$\{1,2\}$	$\{1\}$	$\{2\}$	$\{1,2\}$	
Values	3	0	3	0	6	6	0	2	4	
1	(0)	0	(0)	0	(0)	(0)	0	0	(0)	{0,0,0,0}
	Pivot: $E(B)$. WD selects $\{\{1\},\{2\},\emptyset\}$. Buyer $\{3\}$ is unsatisfied.									
2	(0)	0	(0)	0	(0)	(0)	0	0	(1)	{1,1,1,0}
	Pivot: $E(B)$. WD selects $\{\emptyset,\emptyset,\{1,2\}\}$. Buyers $\{1,2\}$ are unsatisfied.									
3	(1)	0	(1)	0	(1)	(1)	0	0	(1)	{2,1,1,2}
	Pivot: $E(B)$. WD selects $\{\{1\},\{2\},\emptyset\}$. Buyer $\{3\}$ is unsatisfied.									
4	(1)	0	(1)	0	(1)	(1)	0	(0)	(2)	{2,2,2,2}
	Pivot: $E(B)$. WD selects $\{\{1\},\{2\},\emptyset\}$. Buyer $\{3\}$ is unsatisfied.									
5	(1)	0	(1)	0	(1)	(1)	0	(1)	(3)	{3,3,3,2}
	Pivot: $E(B)$. WD selects $\{\emptyset,\emptyset,\{1,2\}\}$. Buyers $\{1,2\}$ are unsatisfied.									
6	(2)	0	(2)	0	(2)	(2)	(0)	(1)	(3)	{4,3,3,4}
	Pivot: $E(B)$. WD selects $\{\{1\},\{2\},\emptyset\}$. Buyer $\{3\}$ is unsatisfied.									
7	(2)	0	(2)	0	(2)	(2)	(0)	(2)	(4)	{4,4,4,4}
	CEs of economies $E(B), E(B_{-2}), E(B_{-3})$ are reached. $\{\{1\},\{2\},\emptyset\}$ is an efficient allocation of $E(B)$. Pivot: $E(B_{-1})$. WD selects $\{\emptyset,\emptyset,\{1,2\}\}$. Buyer $\{2\}$ is unsatisfied.									
8	(2)	0	(2)	0	(3)	(3)	(0)	(2)	(4)	{5,4,4,5}
	Pivot: $E(B_{-1})$. WD selects $\{\emptyset,\emptyset,\{1,2\}\}$. Buyer $\{2\}$ is unsatisfied.									
9	(2)	0	(2)	0	(4)	(4)	(0)	(2)	(4)	{6,4,4,6}
	An UCE price is reached. Final allocation: $\{\{1\},\{2\},\emptyset\}$. Final payment: $p_1(\{1\}) = 2-[6-4]=0,\ p_2(\{2\}) = 4-[6-4]=2,\ p_3(\emptyset)=0$.									

Auctions -Type of items-	Conditions under which the VCG outcome is achieved	Number of price paths	Search for a CE of economy	Is final price equal to final payment?
Demange et al. '86 -Heterogeneous-	Unit demand	Single	Main	Yes
Ausubel '04 -Homogeneous-	Non-increasing marginal values	Single	Main	Yes
de Vries et al. '04 Ausubel and Milgrom '02 -Heterogeneous-	Buyers are submodular	Single	Main	Yes
de Vries et al. '04 (Modified with multiple price paths) -Heterogeneous-	General valuations	Multiple	Main	Yes (but on different price paths)
Ausubel [2] -Heterogeneous-	Gross substitutes	Multiple	Main and marginal	No
Mishra & Parkes'05 -Heterogeneous-	Buyers are substitutes	Single	Main	No
Mishra & Parkes'05 -Heterogeneous-	General valuations	Single	Main and marginal	No

Communication Complexity of CAs

- Finding an optimal solution requires exponential communication. (Nisan-Segal'04)
- Finding an $O(m^{1/2-\varepsilon})$-approximation requires exponential communication. (Nisan-Segal'04)
- See Blumrosen & Nisan (EC'05), and Segal & Nisan (TARK'X) for worst-case results on communication complexity for demand-query based models.

⇒ what worse-case results can we achieve?

Demand Queries & Learning Theory

[Hudson & Sandholm, Conen & Sandholm, Parkes, Zinkevich et al., Blum et al., Lahaie & Parkes]

- Computational learning theory: Learn exact representation of some target function $f : X \rightarrow Y$ in number of queries that are polynomial in $m=\dim(X)$ and $size(f)$, which is the *minimal size of f in some representation class C*.
- Efficient elicitation: Determine the efficient allocation in number of queries that are polynomial in m (number of goods) and $\max_i\{size(v_i)\}$, where $size(v_i)$ is the *minimal size of valuation v_i in some valuation (bidding) language L.*
- Also, note we wish to stop early (elicit, not learn.)

Part III:

Elicitation via Learning Theory

Distributed Implementations

Bidding languages

(Sandholm'99,Nisan'00)

- XOR: $v_i(S) = \max_{S' \subseteq S} v(S')$
- OR: $v_i(S) = \max_{S'1,\ldots,S'k \in Feas(S)} \sum_k v(S'_k)$
- Generalize to "atomic languages" (Lahaie et al.'05)
- OR*: use dummy goods to construct constraints on feasible combinations of bids (Nisan'00)
- L_{GB} (Boutilier & Hoos'01); Tree-Based BL (Cavallo et al.05) generalize to allow arbitrary logical constraints
- Polynomial: $v_i(S)=a_0 \cdot x_1 + a_1 \cdot (x_1x_3) - a_2 \cdot (x_1x_5) + \ldots$ (Lahaie & Parkes'04)
- Read-once formulae, DNF-formulae (Zinkevich et al.'03)

Style of results

- [Zinkevich et al. 2003; Santi et al. 2004] Learning algorithms for read-once formulae and Toolbox DNF, others…
  - Only use value queries.
- [Blum et al. 2004] Elicitation in poly-queries when learning needs exponential queries
  - Exponential number of linear-price demand queries to learn a sparse XOR representation
- Interesting to explore the role of non-linear price demand queries (Lahaie & Parkes'04)
  - Present prices $p(S)$, candidate bundle S.
  - Yes: $S \in \arg\max_{S'} v_i(S')-p(S')$
  - No, provide some S'' s.t. $v_i(S'')-p(S'') > v_i(S)-p(S)$

Frameworks

Learning

- Function Class C
  - Monotone Boolean functions
- Representation Class C
  - Monotone DNF formulae
- Target function $f: X \rightarrow Y$
  - Boolean domain X
  - m-dimensional
  - Boolean or real-valued range Y

Elicitation

- Valuation Classes $V_1,\ldots,V_n$
  - Free-disposal
- Bidding Languages $V_1,\ldots,V_n$
  - XOR bids
- True valuations $v_i: X \rightarrow Y$
  - Domain X of bundles
  - m goods
  - Range Y of non-negative real values

Queries (1)

Learning

- Membership query
- Present an input x.
- Oracle returns the truth-value $f(x)$.

Elicitation

- Value query
- Present a bundle x.
- Agent returns the exact value $v_i(x)$.

Queries (2)

Learning

- Equivalence query
- Maintain manifest $\tilde{f}$ hypothesis
- Present manifest hypothesis to the oracle
- Oracle replies 'Yes' if $\tilde{f}(x) = f(x), \ \forall x \in X$
- Else presents some input x' such that: $\tilde{f}(x') \neq f(x')$

Elicitation

- Demand query $\tilde{v}_1, ..., \tilde{v}_n$
- Maintain manifest valuations
- Present allocation $(x_1, ..., x_n)$ and candidate CE prices $p_i(x)$
- Agent i replies 'Yes' if $x_i \in \arg\max_{x \in X} v_i(x_i) - p_i(x_i)$
- Else presents a bundle x'_i such that: $v_i(x'_i) - p_i(x'_i) > v_i(x_i) - p_i(x_i)$

Objectives

Learning

- Determine target function exactly.
- Use only membership and equivalence queries.
- Run-time is polynomial in m and size(f)

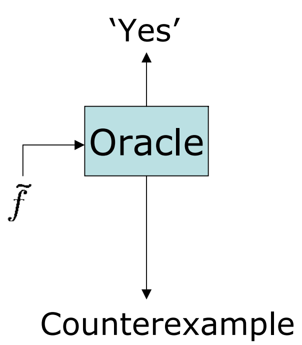
Elicitation

- Determine efficient allocation to the agents.
- Use only value and demand queries.
- Communication is polynomial in n, m and size($v_1,\ldots,v_n$).

Simulation of Equivalence with Demand

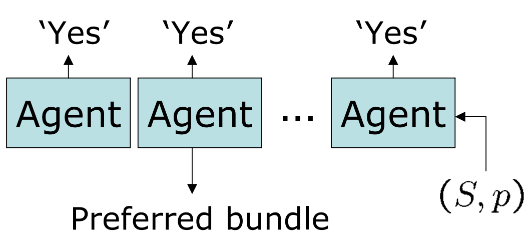
Equivalence

(Learning is solved)



Demand

(Elicitation is solved)

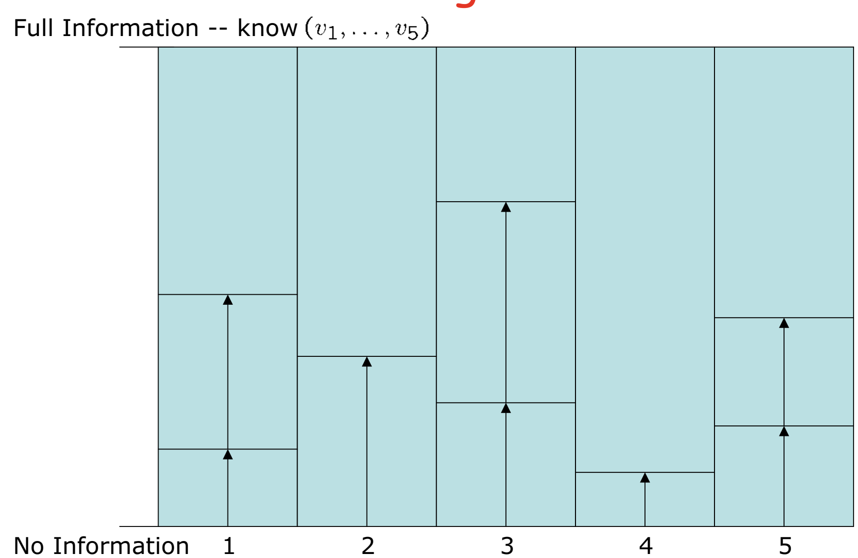


Then

- Preferred bundle, or
- Proposed bundle

is a counterexample.

The Algorithm



Polynomial Elicitation for CAs

(Lahaie & Parkes'04)

Theorem. The efficient allocation can be determined in poly(n,m,size($v_1,\ldots,v_n$)) queries with value and non-linear demand queries for class $V_1 \times \ldots \times V_n$ if they can each be polynomial-query learned.

Polynomials: t terms, m goods, n agents (Schapire & Sellie'93)

$v_i(S)=a_0 \cdot x_1 + a_1 \cdot (x_1x_3) - a_2 \cdot (x_1x_5) + \ldots$

Concise for valuations "almost substitutes"

$O(nmt)$ demand queries, $O(nmt^3)$ value queries

XOR bids: t terms, m goods, n agents

XOR bids can be efficiently learned, generalizing a learning algorithm for monotone DNF (Angluin 87).

compact for valuations "almost complements"

worst-case t+1 demand queries, mt value queries

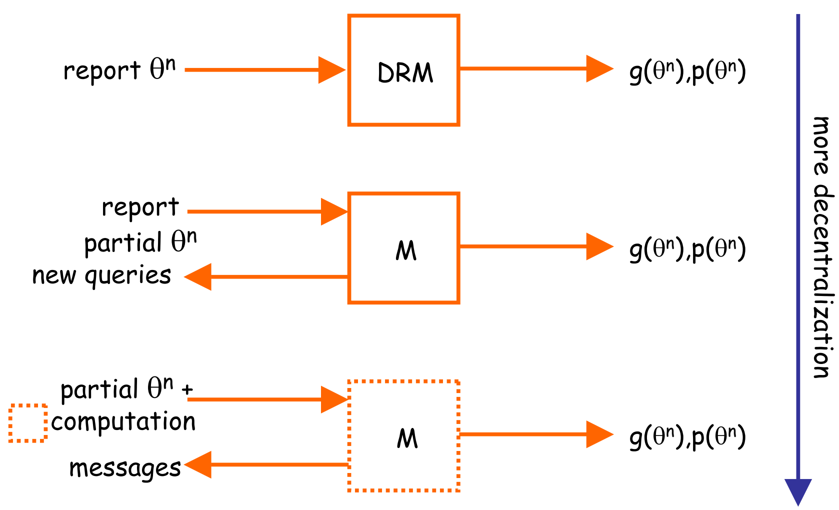
Modification: Universal Queries

(Lahaie, Constantin & Parkes'05)

- Universal Demand Queries $\langle p, \{S_1, S^{-2}_1, \ldots, S^{-n}_1\} \rangle$
  - Compute provisional allocations in main and marginal economies based on manifest valuations
  - Compute candidate UCE prices
  - Report agent i's bundle in each economy, as well as price
  - Agent replies "Yes" if every bundles in demand-set, otherwise provides a counterexample

$\Rightarrow$ terminate with UCE prices, and implement VCG outcome

Where are we?



Static & Decentralized MD

- Center + Incremental-revelation
  - Characterization of minimal information requirements to implement scfs
  - Design of incremental-revelation mechanisms
  - Price-based, computational-learning theory based
- Distributed computation
  - Good "network complexity"
  - Bring computation and information revelation into an equilibrium

Distributed Implementation

(Monderer & Tennenholtz 99; Feigenbaum et al.02; Feigenbaum & Shenker 02; Parkes & Shneidman 04; Shneidman & Parkes 04)

- Distributed Algorithmic Mechanism Design (Feigenbaum et al.'02)
  - distributed algorithm (agents perform computation)
  - achieve good "network complexity"
  - implement outcomes without a center
- Distributed implementation (Parkes & Shneidman'04)
  - distributed algorithm (agents perform computation)
  - perhaps still a center
  - bring computation + message-passing + information-revelation into an equilibrium

Example: Distributed VCGs

- Take $M=\langle\Theta,g,p\rangle$ and distribute computation of $g(\theta)$ and $p(\theta)$ to agents.

- **Example**: distributed *combinatorial auction*:
  - Step 1: agents report θ to center
  - Step 2: dispatches computation of $V(N)$, $V(N\backslash 1),\ldots V(N\backslash n)$ to subsets of agents.
  - Step 3: center receives results, and uses them to implement the outcome of VCG.

New manipulations

- Agent 1 can now deviate from the "intended protocol" and effect a change in:
  - the reported types of other agents
  - the mechanism's rules $\langle g, p\rangle$
  - use observations to implement an adaptive bidding strategy
- For instance, the payment to agent i is
  $p_{vcg,i}=\sum_{j\neq i}v_j(a^{-i};\theta_j)-\sum_{j\neq i}v_j(a^*;\theta_j)$
- Agent i would prefer to:
  - minimize $\sum_{j\neq i}v_j(a^{-i};\theta_j)$, e.g. by obstructing computation of a^{-i}
  - maximize $\sum_{j\neq i}v_j(a^*;\theta_j)$, e.g. by artificially inflating the reported values of other agents for a^*

Idea One: A Partition Principle

(Parkes & Shneidman'04)

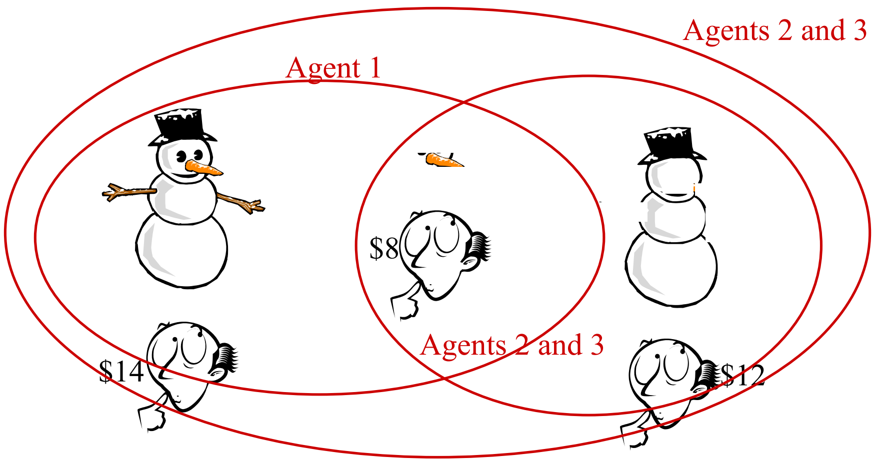
Consider the distributed CA. Assume agents cannot tamper with the reported values of each other.

Theorem. d_M is a "faithful" distributed VCG implementation when the correct solution to $V(N\backslash i)$ is computed whatever the actions of agent i.

OK to ask agents to compute $V(N)$

OK to ask agents $\neq i$ to compute $V(N\backslash i)$

General idea: ask agents to do computation that is in their self-interest to complete, or for which they are indifferent.



Idea Two: Quorums

(Parkes & Shneidman'04)

- Sequence computation into steps: $step^1, step^2, \ldots step^T$.
- Give each step to 3 or more agents:
  - Agents report solution to the center, which selects quorum
  - center can also do random "checking," punish agents to provide focal point.

Assume agents cannot tamper w/ reports of other agents.

Theorem. d_M is a "faithful" distributed implementation when the corresponding centralized mechanism is truthful and when a quorum approach is used for all computation.

Formal definition: Distributed Implementation

(Shneidman & Parkes'04)

$d_M = (f, \Sigma, s^m)$

outcome rule (choice & prices) $f(s(\theta)) \in A \times R^n$

strategy $s_i \in \Sigma_i$

suggested strategy $s^m_i(\theta_i) \in \Sigma_i$ "intended implementation"

Goal: bring $(s^m_1, \ldots, s^m_n)$ into an ex post Nash eq.
strategy: computation, communication, info-revelation.

Decomposition: (R,C,P)

Suggested strategy s^m decomposes:

$s^m_i = (r^m_i, c^m_i, p^m_i)$

info-rev action "only effect is to provide info about type θ_i"

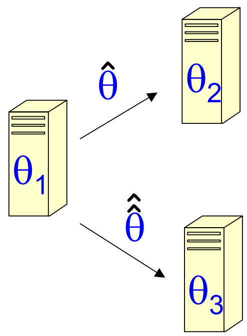
$f(s', s^m_{-i}(\theta_{-i})) = f(s^m_i(\theta'_i), s^m_{-i}(\theta_{-i}))$, for all s' that differ only in r^m_i

comput. action "action can affect outcome rule" (not just info-rev) **(new)**

message-passing action, "send a message, unchanged" **(new)**

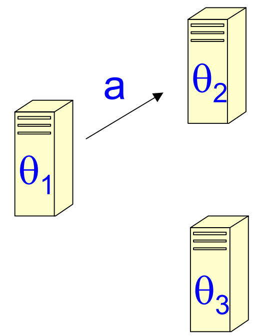
Adopt a message-passing architecture.

Information Revelation Action, r_i



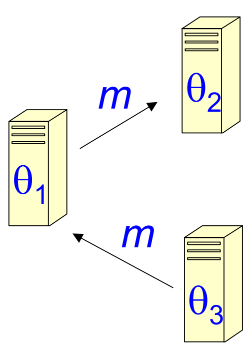
- r_i: reveal private type information to neighbors

Computational Action, c_i



- r_i reveal private type information to neighbors
- c_i: perform some local computation, and report result "a" to a neighbor.

Message Passing Action, p_i



- r_i reveal consistent (perhaps partial or untruthful) type info.
- c_i: perform some local computation, and report result "a" to a neighbor.
- p_i: relay a message from another agent.

Faithful Implementation

Definition. $d_M=(f,\Sigma,s^m)$ is a faithful implementation of outcome $g(\theta)=f(s^m(\theta))$ if strategy s^m is an ex post Nash eq.

Incentive compatibility (IC): will perform all information-revelation actions truthfully in equilibrium.

Algorithm compatibility (AC): will follow the specified computational actions in equilibrium.

Communication compatibility (CC): will follow the specified communication actions in equilibrium.

Theorem. A d_M is faithful when s^m is IC, CC, and AC in the same ex-post Nash equilibrium.

- Only revelation actions (IC):
  - ascending-price auctions
  - standard methods from OR, e.g. Dantzig-Wolfe decomposition
- Computational (AC) and revelation actions (IC):
  - partition principle for VCG
  - quorum approach
- *AC + IC + CC?*
  - e.g. distributed auction on P2P network
  - e.g. shortest-cost path routing on Internet

General Proofs of Faithful Impl.

- Need to be able to argue that there is no useful "joint deviation" amongst:
  - computational actions
  - communication actions
  - information-revelation actions
- Large strategy space:
  - helps to decouple by establishing stronger claims

A General Proof Technique

(Shneidman & Parkes'04)

- Algorithm compatible (AC)
  - an agent implements suggested computation c^m in equilibrium.
- **Strong** *AC*
  - an agent chooses to implement c^m, *whatever* r^m and p^m actions

A General Proof Technique

(Shneidman & Parkes'04)

- Algorithm compatible (AC)
  - an agent implements suggested computation c^m in equilibrium.
- **Strong** *AC*
  - an agent chooses to implement c^m, *whatever* r^m and p^m actions
- Comm. compatible (CC)
  - an agent follows suggested message-passing p^m in equilibrium.
- **Strong** *CC*
  - an agent chooses to implement p^m, *whatever* r^m and c^m actions

A General Proof Technique

(Shneidman & Parkes'04)

- Algorithm compatible (AC)
  - an agent implements suggested computation c^m in equilibrium.
- **Strong** *AC*
  - an agent chooses to implement c^m, *whatever* r^m and p^m actions
- Comm. compatible (CC)
  - an agent follows suggested message-passing p^m in equilibrium.
- **Strong** *CC*
  - an agent chooses to implement p^m, *whatever* r^m and c^m actions

Theorem. If the corresponding centralized mechanism $f(s^m(\theta))$ is **truthful**, and d_M is *strong AC* and *strong CC*, then we have a faithful implementation.

A General Proof Technique contd..

(Shneidman & Parkes'04)

1. Take a truthful mechanism and distributed algorithm.
2. Decompose d_M into disjoint phases.
3. Prove strong-CC and strong-AC for each phase regardless of actions in other phases.
4. Ensure that a "checkpoint" exists in the specification that separates phases.
   -- so that each phase can be proved independently

Application to Lowest-Cost Routing on Internet

(Shneidman & Parkes'04)

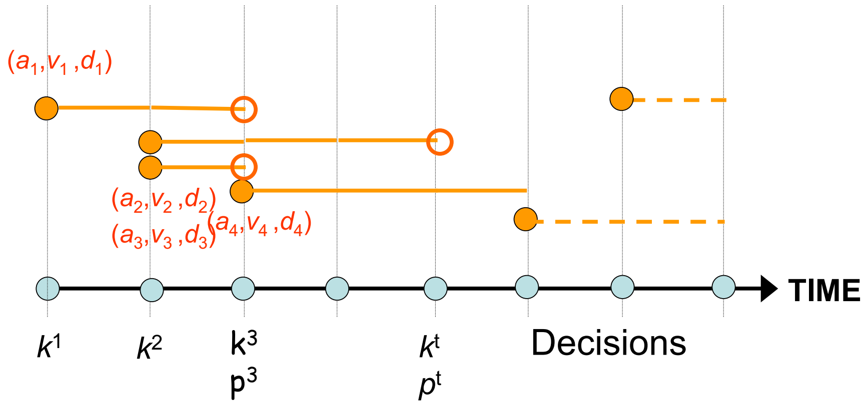
- Feigenbaum et al.'02 (FPPS) studied a distributed algorithm for computing VCG on lowest-cost interdomain routing problem.
- Work in abstract BGP model, achieve with minimal additional space & computational requirements.
- FPSS is not AC or CC: drop, change or spoof routing & pricing messages; deviate from LCP and pricing computation.
- Fix: propose minimal extensions to make this a faithful implementation. Neighbors of nodes on graph perform checking & "catch and punish."

Outline

- Static & Centralized MD
- Static & Decentralized MD
- Dynamic & Centralized MD
  - online auctions, online MD
  - truthful characterizations
- Adaptive & Decentralized MD

Dynamic & Centralized MD

- Agents can arrive and depart dynamically
- Mechanism makes a sequence of decisions, maintains a state of the world.



Dynamic & Centralized MD

- T discrete time points. Decisions $k_1, \ldots, k_T$
- Agent i, type $\theta_i = \langle a_i, v_i, d_i \rangle$ where $v_i(k, \theta_i)$ is its value for a sequence of decisions k
- Dominant-strategy truthful:
  - unit-demand auctions (Lavi & Nisan'00;Hajiaghayi et al'04)
  - reusable items (Hajiaghayi et al.'05, Porter'04)
  - single-minded agents (Awerbuch et al.'03)
  - bounded-demand (Bartal et al.'03)
  - double-auctions (Bredin & Parkes'05)
- Bayesian-Nash truthful:
  - more general sequential decision problem (Parkes & Singh'03, Parkes et al.'04)
  - take an *Markov Decision Process* approach

Example: Last-Minute Tickets

Value	\$100	\$80	\$60
Arrival:	11am	11am	12pm
Patience:	2hrs	2hrs	1hr

"Please bid your value and your patience. A decision will be made by the end of your stated patience."

How should you bid?

Value	\$100	\$80	\$60
Arrival:	11am	11am	12pm
Patience:	2hrs	2hrs	1hr

Auction: sell one ticket in each hour (given demand), to the highest bidder at second-highest bid price.

If truthful, then:
{ <1, \$80>, <2, \$60>}
However, bidder 1 could
a) reduce bid price to \$65
{<2, \$65>, <1, \$60>}
b) delay bid until 12pm
{<2, \$0>, <1, \$60>}

Part IV:

Online Auctions & MD

Adaptive Mechanisms

Basic Model for Online Auctions

- Valuation $v_i = \langle a_i, d_i, w_i \rangle$
- Arrival time: a_i. Departure time: d_i .Value, w_i
- Allocation schedule x.
- $v_i(x) = w_i$, if $x_i(t)=1$ for some $t \in [a_i, d_i]$
  $= 0$, otherwise
- Quasi-linear utility: $u_i(x,p) = v_i(x) - p$

- Auction: $A = \langle f, \tilde{p} \rangle$,
  - allocation rule, $f : V^n \to X$
  - payment rule, $\tilde{p} : V^n \to R^n$

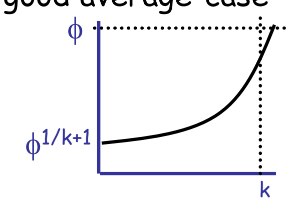
- Truthful auction: reporting value $\langle a_i, d_i, w_i \rangle$ immediately upon arrival is a dominant strategy equilibrium

vs. Powerful Adversarial Model

(Lavi & Nisan'00)

- Assume values in $[L,U]$. Multi-unit. Let $\phi = (U/L)$.
- Adversarial model: choose values and timing.
- Define a "price schedule": $p(j) = L \cdot \phi^{j/k+1}$, for $j=1,\ldots,k$
- Sell units while marginal value $\geq$ price.

Truthful.
$\ln(\phi)$-competitive w.r.t. efficiency and Vickrey revenue,
Matching lower-bound, and good average-case performance in simulation.



vs. Fixed, Unknown Distribution

(Hajiaghayi, Kleinberg, Parkes'04)

- More realistic adversarial model.
  - Lavi & Nisan allowed arbitrary sequencing of arbitrary values
- Instead, we model values as i.i.d. from some unknown distribution.
- Want good performance whatever the distribution.
- Should lead to an auction with better performance in practice.

The Online Selection Problem
"secretary problem"

- Remove incentives, and specialize to the case of disjoint arrival-departure intervals.

5	2	7	1,000	3	. . .

The Online Selection Problem
"secretary problem"

- Remove incentives, and specialize to the case of disjoint arrival-departure intervals.
- Reduces to the secretary problem:
  - interview n job applicants in random order, want to max prob of selecting best applicant (told n)
  - told *relative ordering* w.r.t. applicants already interviewed, must hire or pass

5	2	7	1,000	3	. . .

The Online Selection Problem
"secretary problem"

- Remove incentives, and specialize to the case of disjoint arrival-departure intervals.
- Reduces to the secretary problem:
  - interview n job applicants in random order, want to max prob of selecting best applicant (told n)
  - told *relative ordering* w.r.t. applicants already interviewed, must hire or pass

useful info	1	2	1	1	4	
	5	2	7	1,000	3	. . .

The Secretary Algorithm

- **Theorem** (Dynkin, 1962): The following stopping rule picks the maximum element with probability approaching $1/e$ as $n \to \infty$.
  - Observe the first $\lfloor n/e \rfloor$ elements. Set a threshold equal to the maximum quality seen so far.
  - Stop the next time this threshold is exceeded.
- Asymptotic success probability of $1/e$ is best possible, *even if the numerical values of elements are revealed.*
  - i.e. optimal competitive ratio in the large n limit

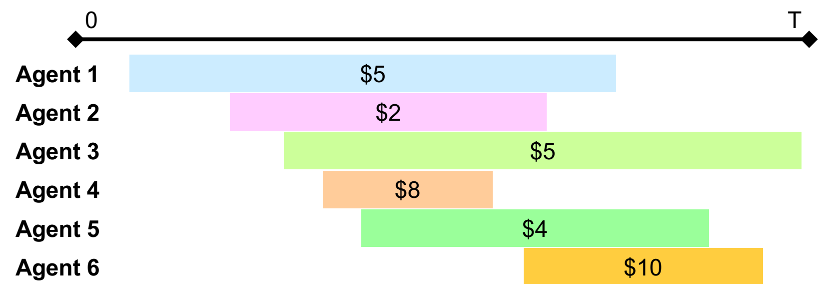
Straw model for an Auction

- **Auction**: $p(t)=\infty$, then set $p(t\geq\tau)=\max_{i\leq j} w_i$ after $j=\lfloor n/e \rfloor$ bids received. Sell to first subsequent bid with $w_i \geq p(t)$, then set $p(t)=\infty$.
- **Not truthful**: Bidders that span transition, and with high enough values, should delay arrival.

> Truthful Auction:
> - At time τ (for n/e arrival) let $p\geq q$ be the top two bids yet received.
> - If any agent bidding p has not yet departed, sell to that agent (breaking ties randomly) at price q.
> - Else, sell to the next agent whose bid is at least p (breaking ties randomly)

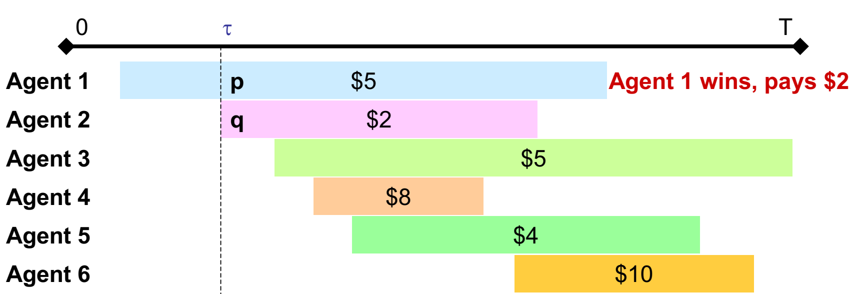
Adaptive Limited-Supply Auction

- At time τ, denoting arrival $j=\lfloor n/e \rfloor$, let $p\geq q$ be the top two bids yet received.
- If any agent bidding p has not yet departed, sell to that agent (breaking ties randomly) at price q.
- Else, sell to the next agent whose bid is at least p.



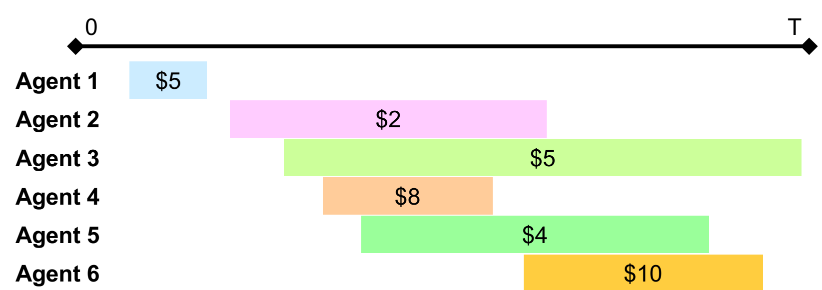
Adaptive Limited-Supply Auction

- At time τ, denoting arrival $j=\lfloor n/e \rfloor$, let $p\geq q$ be the top two bids yet received.
- If any agent bidding p has not yet departed, sell to that agent (breaking ties randomly) at price q.
- Else, sell to the next agent whose bid is at least p.



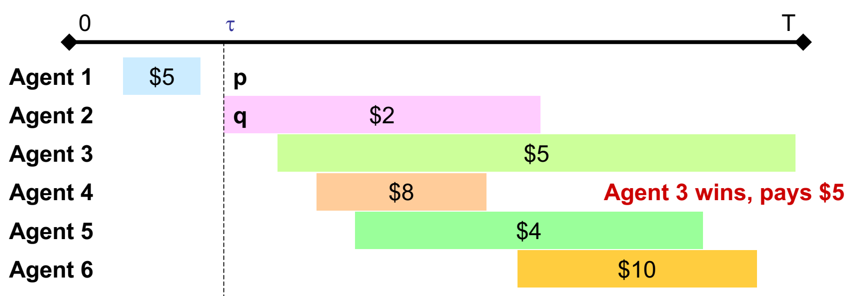
Adaptive Limited-Supply Auction

- At time τ, denoting arrival $j=\lfloor n/e \rfloor$, let $p\geq q$ be the top two bids yet received.
- If any agent bidding p has not yet departed, sell to that agent (breaking ties randomly) at price q.
- Else, sell to the next agent whose bid is at least p.



Adaptive Limited-Supply Auction

- At time τ, denoting arrival $j=\lfloor n/e \rfloor$, let $p \geq q$ be the top two bids yet received.
- If any agent bidding p has not yet departed, sell to that agent (breaking ties randomly) at price q.
- Else, sell to the next agent whose bid is at least p.



Analysis: Competitive Ratio

- Competitive ratio for efficiency is $e+o(1)$, assuming all valuations are distinct.

Proof.

- Case 1: Item sells at time t. Winner is highest bidder among first $\lfloor n/e \rfloor$. With probability $\sim 1/e$, this is also the highest bidder among all n agents.
- Case 2: Otherwise, the auction picks the same outcome as the secretary algorithm, whose success probability is $\sim 1/e$.

General approach: Two phase

- "Learning phase"
  - use a sequence of bids to set price for rest of auction

Transition:
  - be sure that remains truthful for agents on transition
- "Accepting phase"
  - exploit information, retain truthfulness

Necessary and Sufficient Characterization

(Hajiaghayi, Kleinberg, Mahdian, and Parkes'05)

- Price schedule $ps_i(a_i,d_i,v_{-i})$ is **monotonic** if $ps_i(a_i,d_i,v_{-i}) \leq ps_i(a'_i,d'_i,v_{-i})$, for all $a'_i \geq a_i$ and $d'_i \leq d_i$.
- Auction is "price-based" if exists ps_i s.t. $f_i(v)=1$ iff $ps_i(a_i,d_i,v_{-i}) \leq v_i$, and payment $\tilde{p}_i(v)=ps_i(a_i,d_i,v_{-i})$.
- Critical period: first $t \in [a_i,d_i]$ with minimal $ps_i(a_i,t,v_{-i})$

Theorem. An online auction is truthful if and only if the auction is price-based for some monotonic price schedule $ps_i(a_i,d_i,v_{-i})$, and assigns the item after the critical period.

Special case: define $ps_i(a_i,d_i,v_{-i})=\min_{\tau\in[a,d]} ps(\tau,v_{-i})$, for some $ps(t,v_{-i})$

Monotonic Allocation Rules

Another way to get this:

- Allocation rule $f: V^n \to \{0,1\}^n$ is **monotone** if for every agent i and every $v, v' \in V^n$ with $[a'_i,d'_i] \subseteq [a_i,d_i]$, and $w_i \geq w'_i$, we have $f_i(v) \geq f_i(v')$.
- Define Critical Value,
  $v^c(a_i,d_i,v_{-i}) = \min w'_i$ s.t. $f_i(\langle a_i,d_i,w'_i\rangle, v_{-i}) = 1$
  ∞ , if no such w'_i exists),
  - monotonicity implies that $f_i(v)=1$ iff $w_i \geq v^c(a,d,v_{-i})$.

Theorem. Online auction is truthful if and only if the allocation rule, f, is monotonic, sets payment equal to critical value, and assigns item after the critical period.

Application: Reusable Goods

(Hajiaghayi, Kleinberg, Mahdian, and Parkes'05; also Porter'04, Lavi&Nisan'05)

- One good in each time slot (can extend to $k>1$).
- Agent value $\langle a_i,d_i,w_i\rangle$. Value for one time slot in $[a_i,d_i]$.
  (can extend to L_i contiguous slots.)
- No-late departures (i.e. $[a'_i,d'_i] \subseteq [a_i,d_i]$)
  - (WiFi) suppose can verify presence, and fine an agent that reports $d'_i > d_i$ but leaves at d_i
  - (Grid) reasonable to hold result until time d' with some small probability
  - *necessary* to achieve a bounded competitive ratio on efficiency (Lavi & Nisan'05)

Given this, monotone allocation rule $\Rightarrow$ truthful

Online Auction for Reusable Goods

Greedy Allocation rule: In each period, t, allocate the good to the highest unassigned bid.

Payment rule: Pay smallest amount could have bid and still received good.

Note: for impatient bidders this is precisely a sequence of Vickrey auctions.

Montone $\Rightarrow$ truthful

2-competitive (matching LB, c.f. 1.618-competitive result w/out incentives)

Back to our example

Value	\$100	\$80	\$60
Arrival:	11am	11am	12pm
Patience:	2hrs	2hrs	1hr
Duration:	1hr	1hr	1hr

Recall: Sequence of Vickrey auctions, bidder 1 had wanted to delay until 12pm or report \$60+ε.

Truthful auction:
Bidder 1 gets slot 1. Pays \$60
Bidder 2 gets slot 2. Pays \$60

Relaxing: Bayes-Nash equilibrium

- State: $h_t=(\theta_1,\ldots,\theta_t;k_1,\ldots,k_{t-1})$
- Model: $Prob(h_{t+1}|h_t,k_t)$
- Reward: $R(h_t,k_t) = \sum_i R^i(h_t,k_t)$
- Optimal policy: $\pi^*_t : H_t \to K_t$ maximizes value $V^\pi(h_t)=E_\pi\{R(h_t,\pi(h_t))+R(h_{t+1},\pi(h_{t+1}))+\ldots+R(h_T,\pi(h_T))\}$ in all states.

- Bayes-Nash equilibrium: truthful bidding maximizes *expected utility*, in equilibrium and given common knowledge of a model of the problem.

An Online VCG Mechanism

(Parkes & Singh'03)

- Agents report type
- State: reported type + history of decsions
- Reward: depends on reported type of agents present

Online VCG Mechanism:

- Implement optimal policy π^*
- On departure, collect payment

h_{ai}^{-i} is the state in period a_i with agent i removed…

$$R^i_{\le T}(\theta_i;\pi^*) - [V^*(h_{ai})-V^*(h_{ai}^{-i})]$$

total reported value   expected positive effect on value system

Theorem. Online VCG mechanism with an optimal policy π^* for a correct MDP model that satisfies "stalling" is BNIC and implements expected-value maximizing policy

Approximate Online MD

(Parkes et al.,'04)

- Sparse-sampling (Kearns et al. 1999)
- Compute an ε-approximation to the optimal value and action in a state in time independent of the size of state space.
- MDP model M_f used as a generative model.

Approximate Online Mechanism:

Implement policy π' computed by sparse-sampling(ε)

Payments: $R^i_{\le T}(\theta_i;\pi')-[\hat{V}_{ss}(h_{ai})-\hat{V}_{ss}(h_{ai}^{-i})]$

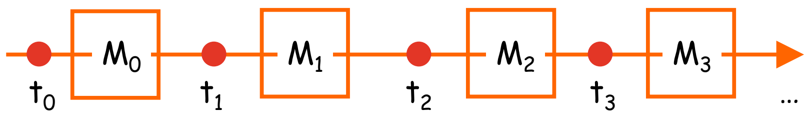
Theorem. Truthful-bidding is an 4ε-BNE of sparse-sampling(ε)-based approximate VCG mechanism.

Outline

- Static & Centralized MD
- Static & Decentralized MD
- Dynamic & Centralized MD
- Adaptive & Decentralized MD
  - uncertain rewards, learning

Learning in Online MD

Like to deploy "black box" mechanism, have it learn and improve over time.
Challenge: maintaining truthfulness while learning

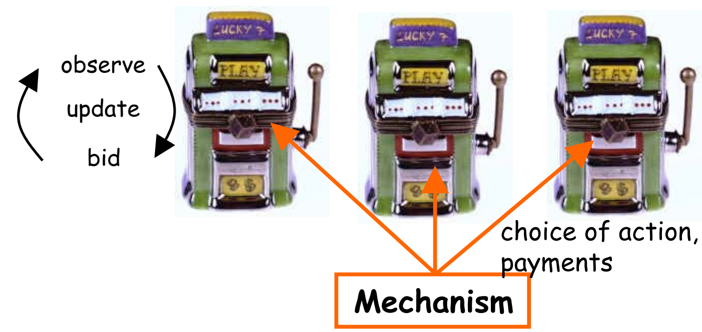


Staged approach to OMD. Not truthful because model inaccurate in early stages.

A Simple Bandits Model

(with Cavallo and Singh)

- Multi-armed bandit (MAB) problem
- N arms (arm == agent)
- Each arm has stationary uncertain reward process, privately observed.
- **Goal**: implement an optimal learning policy



Bayesian-optimal Learning

- No self-interest. Infinite time horizon, discount factor $0 < \gamma < 1$
- n stochastic processes. Information state $s_k(t)$.
- Expected reward $r(s_k(t), k)$ for action $k \in \{1,\ldots,n\}$ in period t.
- Let $f(.,.)$ denote Bayesian updates.
- Update: $s_k(t+1) = f(s_k(t), r_k(t))$, if arm k pulled
  $= s_k(t)$, otherwise.

- **Goal**: $\arg\max_\pi \mathbf{E}\,[\sum_{t=0}^{\infty} \gamma^t\, r(s(t),\pi(s(t)) \mid s(0)]$

Gittins Index

(Gittins & Jones'74)

- Factored algorithm to compute the "Gittins index" for each arm in any state.
- Optimal policy is to pull the arm with the maximal index.
- For finite-state approximations, can compute as optimal MDP value to "restart-in-i" MDP, solve using LP (Katehakis & Veinott'87)
- Analytic results for special-cases (Berry & Fristedt'85)

Straw Auction Model

- Sequence of Vickrey auctions
- Bid Gittins index for each arm
- Pull arm with highest bid, make that arm pay second-highest bid

- Not truthful. Why?
- Agent 1 may have knowledge that the mean reward for arm 2 is smaller than agent 2's current gittins index.
- Learning by 2 would decrease the price paid by 1
- In arm 1's interest to under-bid and allow arm 2 to learn, reduce price in future.

Solution: Long-term Vickrey w/ ε-sampling

(Cavallo, Parkes & Singh'05)

- Each agent maintains Gittins index for its arm.
- In each period t, report $g_k(t)$ and reward $r_k(t-1)$
- With prob $1-\varepsilon$,
  - pull arm with maximal reported $g_k(t)$
- With prob $\varepsilon > 0$,
  - pull arm uniformly at random
  - use to update "ε-statistics"

agent-independent

aligns incentives

- Payments: $T_k(t) = \sum_{j\neq k} r_j(t^\varepsilon(a'_{-k}(t))) - \sum_{j\neq k} r_j(t)$
  - where, $a'_{-k}(t)$ is the optimal action without arm k based on leave-one-out statistics from ε-interleaving samples
  - and $t^\varepsilon(a)$ is the most recent sample for a particular action

Theorem: truthful reporting of Gittins index is a (Perfect) BNE.

Part V:

Wrap-up

Future Directions

- Macroeconomics + Computational Mechanism Design:
  - in grid computing, sensor nets, etc.
  - need to design "central banks"
  - fiscal policy, think about exchange rates, etc.
- Consumption externalities:
  - in grid computing, P2P networks, etc.
- Second-best MD:
  - making tradeoffs between computational cost, informational cost, privacy cost and qualities of approximation
  - equilibrium models for bounded-rational agents
- Learning + CMD:
  - both for agents (learn values for different choices)
  - and for center (learn model of dynamic world)
  - dynamic mediation between learning agents

Review

MD $\rightarrow$ Dec MD $\rightarrow$ online MD $\rightarrow$ adaptive MD

- Price-based mechanisms, monotonicity
- Approximability results (tractable + truthful)
- Elicitation: ascending-price, CLT-based, role of bidding languages
- Distr. implementation: extended equilibrium concepts; AC, CC and IC.
- Online: temporal IC issues, dominant vs BNE models
- Learning: connections to MAB, bring learning into an equilibrium.

Thank You

R.Cavallo, F.Constantin, J.Corbo, J.Feigenbaum, M.Hajiaghayi, A.Juda, J.Kalagnanam, L.Kang, R.Kleinberg, S.Lahaie, M.Mahdian, L.Michael, D.Mishra, C.Ng, T.Sandholm, G.Schoenebeck, M.Seltzer, J.Shneidman, S.Singh, S.Suri, L.Ungar, J.Woodard, D.Yanovsky, S.Youssef.

More information:

www.eecs.harvard.edu/econcs