Mechanism Design Tutorial

David C. Parkes, Harvard University

Indo-US Lectures Week in Machine Learning, Game Theory and Optimization

Outline

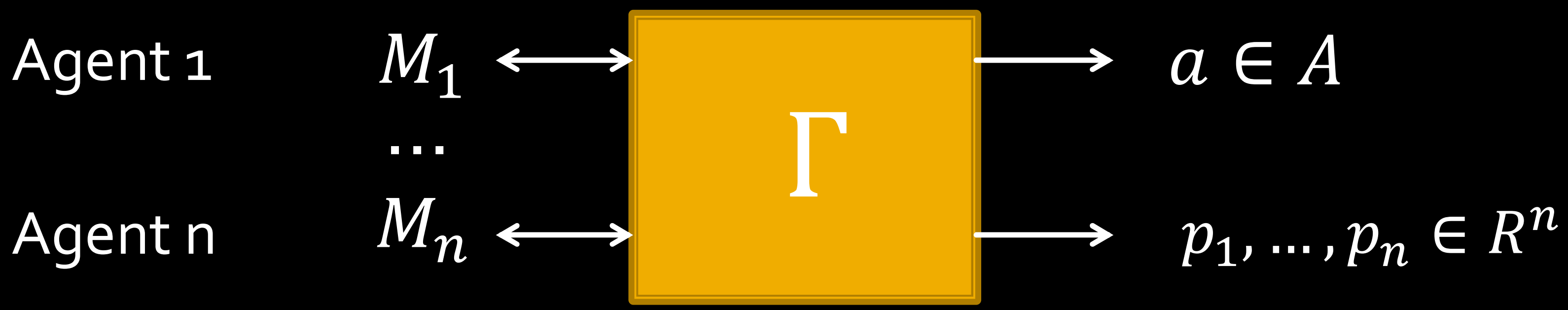
- Classical mechanism design
  - Preliminaries (DRMs, revelation principle)
  - Positive results – Groves, Single-parameter (Myerson)
    - min makespan task assignment
  - Negative results – Gibbard-Satterthwaite
- Algorithmic mechanism design
  - Knapsack auction
  - Price-of-anarchy analysis

Mechanism design

- A alternatives; $i \in N$ agents, value $v_i: A \mapsto R$, $v_i \in V$
- Utility $u_i(a, p) = v_i(a) - p$

Mechanism design

- A alternatives; $i \in N$ agents, value $v_i : A \mapsto R$, $v_i \in V$
- Utility $u_i(a,p) = v_i(a) - p$
- Design a game $\Gamma(M_1, \dots, M_n) \in A \times R^n$, attain desiderata <u>in equilibrium</u>



Examples

- Auction; e.g., servers, bandwidth, ad space
- Coordination; e.g., meetings, tasks
- Public choice; e.g., build a new school
- Matching; e.g., residents to hospitals

- Desiderata: efficiency, maxmin fairness, envy-free, participation, revenue, budget-balance …

Game theory for MD

- Incomplete information game; Valuation $v_i \sim F$
- Behavior b_i ; Strategy $s_i(v_i) \in B$

Game theory for MD

- Incomplete information game; Valuation $v_i \sim F$
- Behavior b_i ; Strategy $s_i(v_i) \in B$

- Dominant strategy equilibrium
- $u_i^{\Gamma}(s_i(v_i), b_{-i}) \geq u_i^{\Gamma}(b_i, b_{-i})$, all i, all b_{-i}, all b_i

Game theory for MD

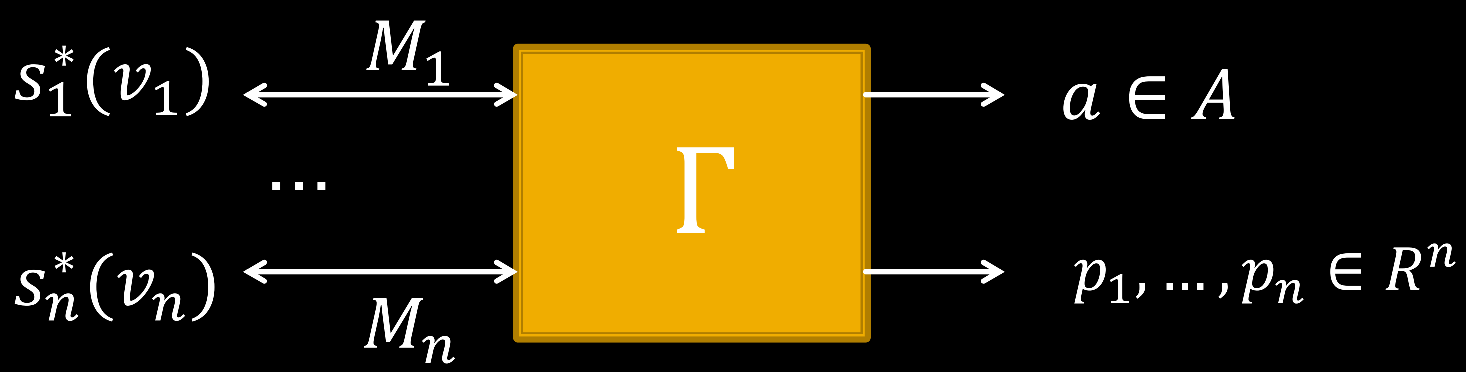
- Incomplete information game; Valuation $v_i \sim F$
- Behavior b_i ; Strategy $s_i(v_i) \in B$

- Dominant strategy equilibrium
- $u_i^\Gamma(s_i(v_i), b_{-i}) \geq u_i^\Gamma(b_i, b_{-i})$, all i, all b_{-i}, all b_i

- Bayes-Nash equilibrium
- $E_{\{v_{-i}\}}[u_i^\Gamma\left(s_i^*(v_i), s_{-i}^*(v_{-i})\right)] \geq E_{\{v_{-i}\}}[u_i^\Gamma\left(b_i, s_{-i}^*(v_{-i})\right)]$, all i, all b_i

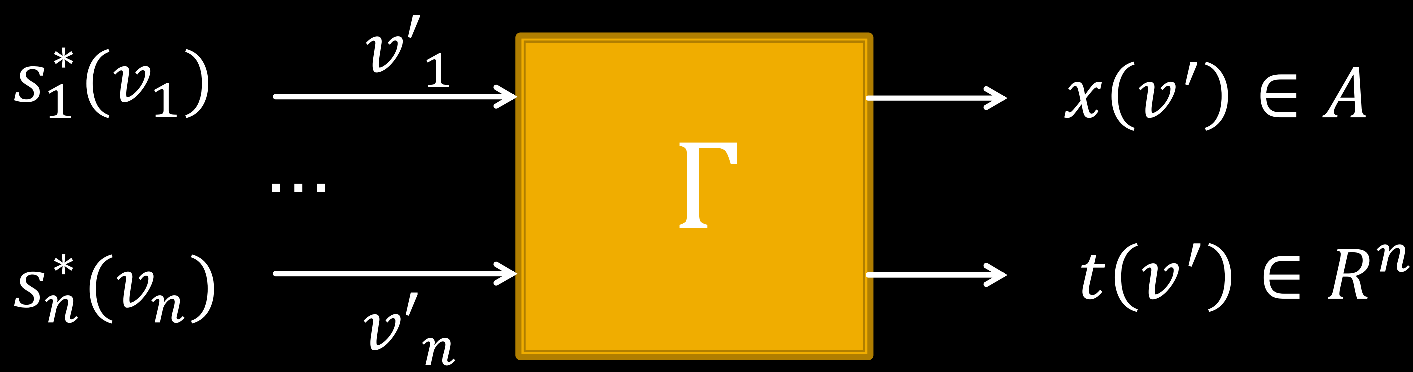
Implementation

- Mechanism Γ *implements* a social choice function $f: V^n \mapsto A$ if $\Gamma_1(s^*(v)) = f(v)$ for all $v = (v_1, \ldots, v_n)$, in equilibrium s^*.



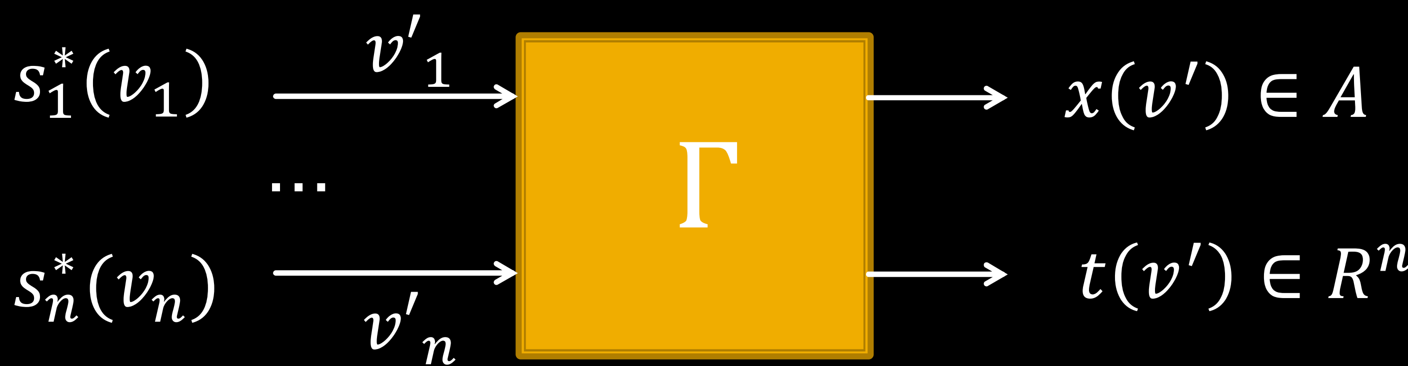
Direct Revelation Mechanisms

- Choice rule x; Payment rule t



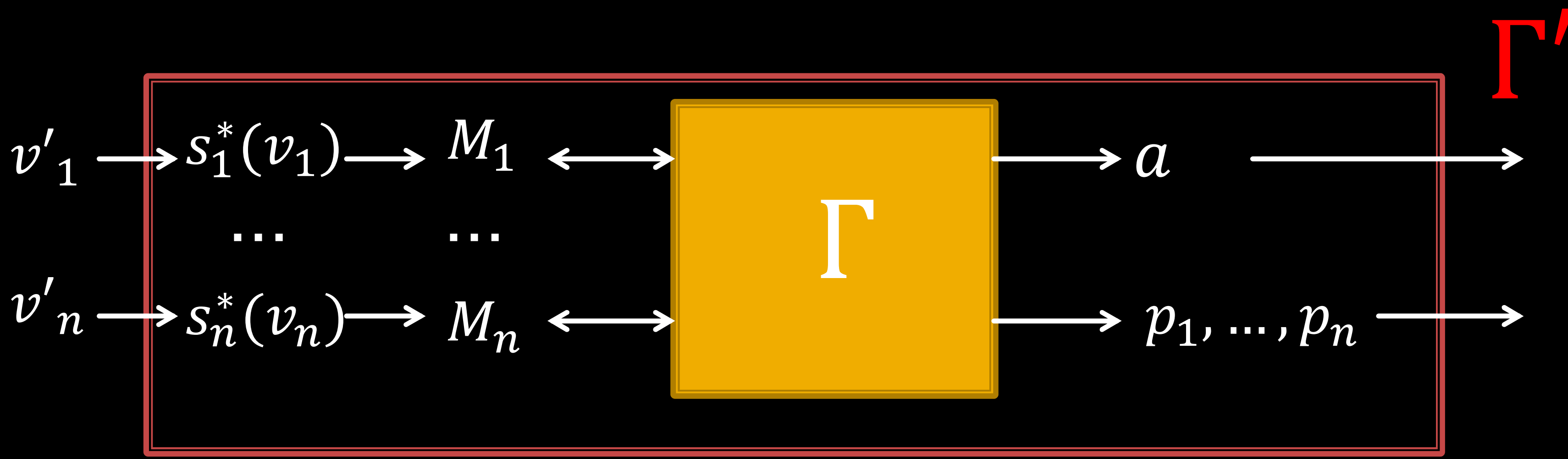
Direct Revelation Mechanisms

- Choice rule x; Payment rule t



- DRM Γ is (Dom/Bayes) incentive compatible if truthful reporting is a (DSE/BNE). ("Strategyproof, "Truthful.")

Revelation Principle



- Theorem: Any scf f implemented by Γ can be implemented by an incentive compatible DRM.

*Positive results **Negative results

Outline

- Classical mechanism design
  - Preliminaries (DRMs, revelation principle)
  - Positive results – Groves, Single-parameter (Myerson)
    - Min makespan
  - Negative results – Gibbard-Satterthwaite
- Algorithmic mechanism design
  - Knapsack auction
  - Price-of-anarchy analysis

Groves mechanism

- $f(v) \in \arg\max_a \sum_i \alpha_i v_i(a) + \beta(a)\,;\, \alpha_i > 0$

Affine maximization (Simple case, $\alpha_i = 1, \beta(a) = 0$)

Groves mechanism

- $f(v) \in \arg\max_{a} \sum_i \alpha_i v_i(a) + \beta(a)\,;\, \alpha_i > 0$

Affine maximization (Simple case, $\alpha_i = 1, \beta(a) = 0$)

- $x_G(v') \in \arg\max_{a} \sum_i v_i'(a)$
- $t_{G,i}(v') = h_i(v_{-i}') - \sum_{j \neq i} v_j'(a)$, for $a = x(v')$

(arbitrary fcn)

Groves mechanism

- $f(v) \in \arg\max_a \sum_i \alpha_i v_i(a) + \beta(a)\,;\ \alpha_i > 0$

Affine maximization (Simple case, $\alpha_i = 1, \beta(a) = 0$)

- $x_G(v') \in \arg\max_a \sum_i v_i'(a)$
- $t_{G,i}(v') = h_i(v'_{-i}) - \sum_{j \neq i} v_j'(a)$, for $a = x(v')$
  (arbitrary fcn)
  O
- Utility: $v_i\big(x(v')\big) + \sum_{j \neq i} v_j'\big(x(v')\big) - h_i(v'_{-i})$

$\Rightarrow$ truthful! (and efficient!)

VCG mechanism

- Special case of Groves.
- Payment rule: Negative externality
- $t_{vcg,i}(v') = \sum_{j \neq i} v'_j(a_{-i}) - \sum_{j \neq i} v'_j(a)$,
  for $a = x_G(v')$, $a_{-i} = x_G(v'_{-i})$.

VCG mechanism

- Special case of Groves.
- Payment rule: Negative externality
- $t_{vcg,i}(v') = \sum_{j \neq i} v'_j(a_{-i}) - \sum_{j \neq i} v'_j(a)$,

  for $a = x_G(v')$, $a_{-i} = x_G(v'_{-i})$.

- Truthful, efficient, participation, no-deficit*

(Negative result (Roberts): if $|A| \geq 3, V = R_+^{|A|}$, then only truthful mechanisms are Groves mechanisms.)

VCG Example 1

- Single-item Auction
- Values \$10, \$4, \$2

- $x(v)$: assign to A1
- $t_1(v) =$

VCG Example 1

- Single-item Auction
- Values \$10, \$4, \$2

- $x(v)$: assign to A1
- $t_1(v) = 4 - 0 = 4$; zero to others
- … a second-price auction

VCG Example 2

- Combinatorial Auction
- Items {A,B,C}

- $x(v)$: $(\emptyset, A, B)$
- $t_2(v) =$

agent	A	B	AB
1	0	0	10
2	6	0	6
3	0	8	8

VCG Example 2

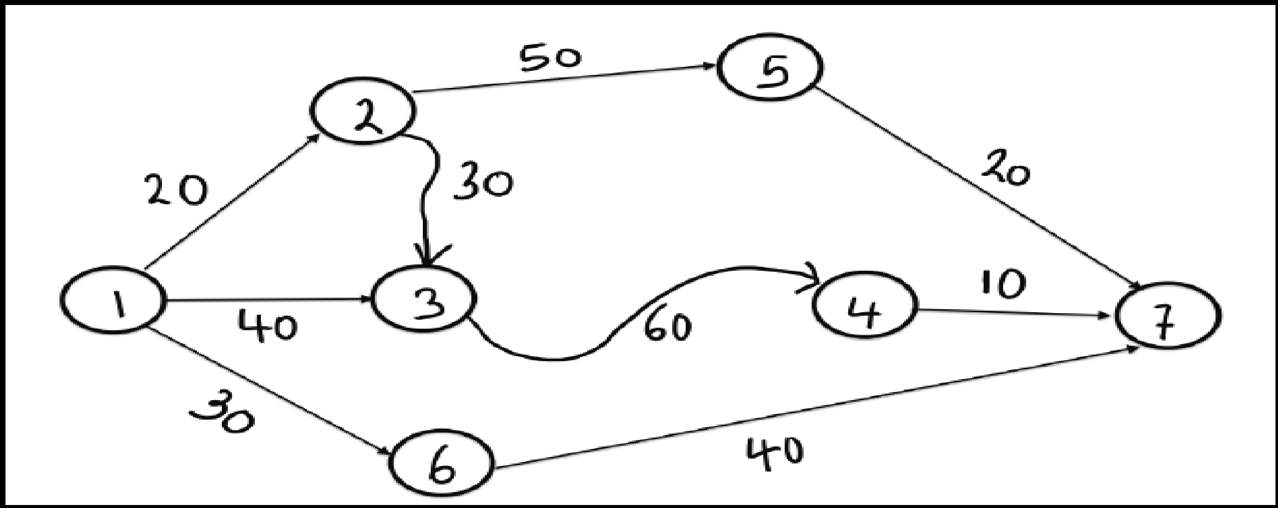
- Combinatorial Auction
- Items {A,B,C}

- $x(v)$: $(\emptyset, A, B)$
- $t_2(v) = 10 - 8 = 2$
- $t_3(v) = 10 - 6 = 4$
- Agent 1 pays zero.

agent	A	B	AB
1	0	0	10
2	6	0	6
3	0	8	8

(revenue low, and NP-hard winner determination.)

VCG Example 3



- Agents= Edges; Value = -Cost
- Externality: (- total cost without) – (- total cost with); e.g., for edge 17 this is -90-(-40)=-50.

VCG Example 4

- Double Auction
- A1: buyer, value \$10
- A2: seller, value \$8

- $x(v)$: trade

VCG Example 4

- Double Auction
- A1: buyer, value $10
- A2: seller, value $8

- $x(v)$: trade
- Payments
  - A1: $8 - 0 = 8$ (pays $8)
  - A2: $0 - 10 = -10$ (paid $10!)

Single-parameter domains

- Private info $w_i \in R$; induces $v_i(w_i, a) \in R$

Single-parameter domains

- Private info $w_i \in R$; induces $v_i(w_i, a) \in R$
- E.g., *Min makespan task assignment*
- Agents A1, A2. Tasks T1,T2,T3 (sizes 1, 2 and 4)
- Private :: Unit processing time ($w_1 = \frac{1}{2}, w_2 = 1$)

	task A	*task B*	*task C*
machine 1	*0.5**	*1*	*2**
machine 2	*1*	*2**	*4*

Min make-span = max(2.5,2)=2.5

What would VCG do?

Single-parameter domains

$$v_i(w_i, a) = w_i \quad \times \quad q_i(a)$$

private $w_i \in [L, \infty)$

known $q_i : A \mapsto R_+$
summarization fcn

(1) Auction: w_i is value, $q_i(a)$ is (prob) agent allocated?
(2) Task assignment: w_i is (−processing time), $q_i(a)$ total load

Single-parameter domains

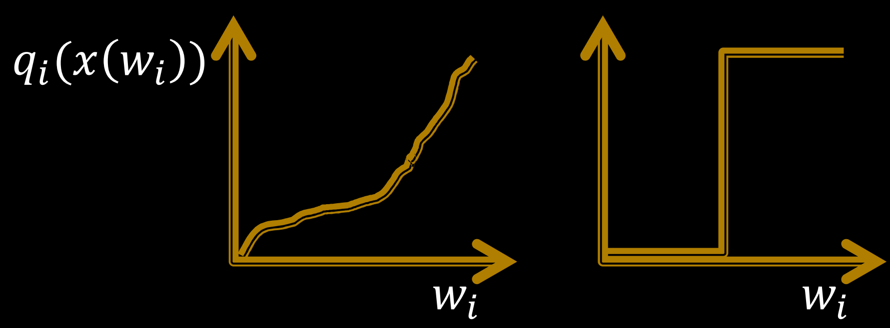
$$v_i(w_i, a) = w_i \quad \times \quad q_i(a)$$

private $w_i \in [L, \infty)$

known $q_i: A \mapsto R_+$
summarization fcn

(1) Auction: w_i is value, $q_i(a)$ is (prob) agent allocated?
(2) Task assignment: w_i is (–processing time), $q_i(a)$ total load

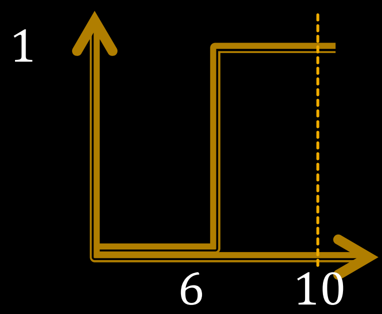
- Allocation rule $x: [L, \infty)^n \mapsto A$
- Fix w'_{-i}, monotonic x



Myerson mechanism (s.p. domain)

- Given <u>monotonic</u> x, then mechanism truthful if:
- $t_i(w') = w_i' q_i\big(x(w')\big) - \int_{z=L}^{w_i'} q_i\big(x(z, w_{-i}')\big) dz - h_{-i}(w_{-i}')$

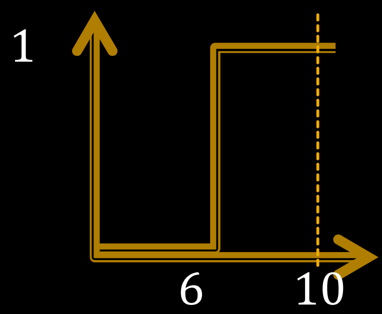
0



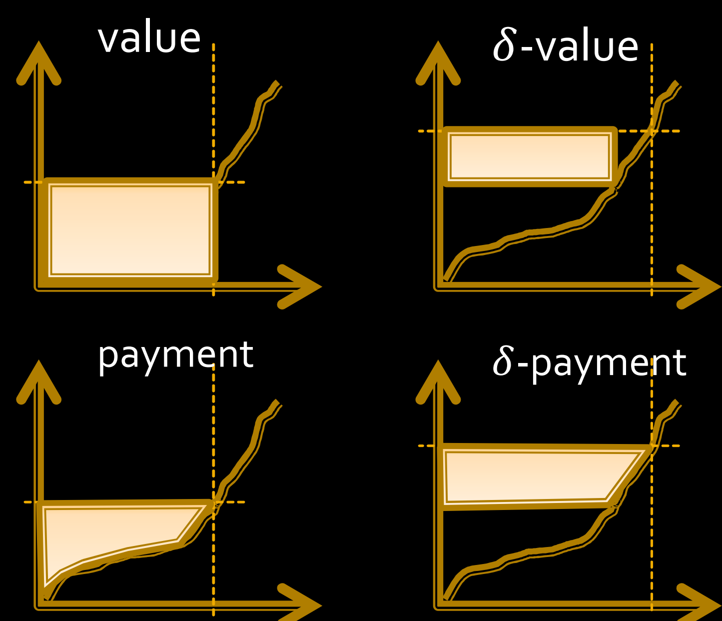
=**critical value** in 0-1 domains

Myerson mechanism (s.p. domain)

- Given <u>monotonic</u> x, then mechanism truthful if:
- $t_i(w') = w_i' q_i\big(x(w')\big) - \int_{z=L}^{w_i'} q_i\big(x(z, w'_{-i})\big)dz - h_{-i}(w'_{-i})$ ↗ 0



=**critical value** in 0-1 domains



(basically necessary)

Min makespan scheduling

(Archer and Tardos'01)

- Unit processing time ($w_1 = \frac{1}{2}, w_2 = 1$)

	task A	task B	task C
machine 1	0.5*	1	2*
machine 2	1	2*	4

c-approx: $\frac{Z_m(w)}{Z_{opt}(w)} <= c$

Min makespan scheduling

(Archer and Tardos'01)

- Unit processing time ($w_1 = \frac{1}{2}, w_2 = 1$)

	task A	task B	task C
machine 1	0.5*	1	2*
machine 2	1	2*	4

c-approx: $\frac{Z_m(w)}{Z_{opt}(w)} <= c$

- Thm. VCG is an n-approx, and truthful.
- Proof. UB: $\frac{Z_m}{Z_{opt}} \leq \sum_j c_{min,j} \; / \; (1/n) \sum_j c_{min,j} = n$

Min makespan scheduling

(Archer and Tardos'01)

- Unit processing time ($w_1 = \frac{1}{2}, w_2 = 1$)

	task A	task B	task C
machine 1	0.5*	1	2*
machine 2	1	2*	4

c-approx: $\frac{Z_m(w)}{Z_{opt}(w)} <= c$

- Thm. VCG is an n-approx, and truthful.
- Proof. UB: $\frac{Z_m}{Z_{opt}} \leq \sum_j c_{min,j} \;/\; (1/n) \sum_j c_{min,j} = n$
- LB: n machines, n tasks (size 1).
- Machine 1 unit cost 1. Machine 2..n unit cost $1 + \epsilon, \epsilon > 0$
- Min makespan $1 + \epsilon$. VCG makespan n.
- $\lim_{\epsilon \to 0} n/(1+\epsilon) = n$

What else can we do?

LexOpt mechanism

(Archer and Tardos'01)

- Adopt $x(v)$ to min makespan, particular tie-breaking rule.
- Thm. LexOpt is monotonic ($w'_i > w_i$, load $a'_i \geq a_i$)

LexOpt mechanism

(Archer and Tardos'01)

- Adopt $x(v)$ to min makespan, particular tie-breaking rule.
- Thm. LexOpt is monotonic ($w'_i > w_i$, load $a'_i \geq a_i$)
- Suppose $a' \neq a$.
- (Case 1) $span_{w_i'}(a') = span_{w_i}(a')$. $span_{w_i'}(a) \leq span_{w_i}(a) \leq span_{w_i}(a') = span_{w_i'}(a')$. Contradiction.

LexOpt mechanism

(Archer and Tardos'01)

- Adopt $x(v)$ to min makespan, particular tie-breaking rule.
- Thm. LexOpt is monotonic ($w'_i > w_i$, load $a'_i \geq a_i$)
- Suppose $a' \neq a$.
- (Case 1) $span_{w_i'}(a') = span_{w_i}(a')$. $span_{w_i'}(a) \leq span_{w_i}(a) \leq span_{w_i}(a') = span_{w_i'}(a')$. Contradiction.
- (Case 2)$span_{w_i'}(a') < span_{w_i}(a')$.

$-w_i q_i(a) \leq span_{w_i}(a) \leq span_{w_i}(a') = -w_i q_i(a')$, since *i* is bottleneck in a' at w_i. Monotone.

Aside: Computation

(Dhangwatnotai et al. 11, Christodoulou and Kovacs 10)

- min- makespan is NP-hard
- Standard PTAS optimizes over a restricted range of candidate assignments, set construction violates monotonicity.
- Exists a monotone PTAS, both randomized and deterministic.

Outline

- Classical mechanism design
  - Preliminaries (DRMs, revelation principle)
  - Positive results – Groves, Single-parameter (Myerson)
    - Min makespan
  - Negative results – Gibbard-Satterthwaite
- Algorithmic mechanism design
  - Knapsack auction
  - Price-of-anarchy analysis

Gibbard-Satterthwaite

- No money. $V \equiv$ all strict preferences (e.g., $a \succ b \succ c$)
- $|A| \geq 3, x(v)$ onto.
- Dictatorial: Same agent always gets top choice

Gibbard-Satterthwaite

- No money. $V \equiv$ all strict preferences (e.g., $a \succ b \succ c$)
- $|A| \geq 3, x(v)$ onto.
- Dictatorial: Same agent always gets top choice

- Theorem. The only truthful mechanisms are dictatorial with all strict prefs, |A|>=3 onto.

Simple proof (T ⇒ D) for 2-agent case

Monotonicity (M)

- abcdef → badcfd

v_i v'_i

If $\boldsymbol{x(v_i, v'_{-i}) = d}$***, then*** $\boldsymbol{x(v'_i, v'_{-i}) = d}$.

Proof. If report ..cd… and get d, then report ..dc… and get g, if g > {c,d} then "cd" type deviates; else, "dc" type deviates.

Simple proof (T ⇒ D) for 2-agent case

Monotonicity (M)

- abcdef → badcfd

v_i v'_i

If $x(v_i, v'_{-i}) = d$, then $x(v'_i, v'_{-i}) = d$.

Proof. If report ..cd… and get d, then report ..dc… and get g, if g > {c,d} then "cd" type deviates; else, "dc" type deviates.

Consistency (C)

- ***If every agent .. $a \succ b$ .. then don't pick b.***

Proof. Suppose pick b. Still pick b if all a>b > … (M)

Onto, so exists v with x(v)=a. Still pick a if all a>b>… (M).

Contradiction.

Simple proof (T ⇒ D) for 2-agent case

Monotonicity (M)

- abcdef → badcfd

v_i v'_i

If $\boldsymbol{x(v_i, v'_{-i}) = d}$***, then*** $\boldsymbol{x(v'_i, v'_{-i}) = d}$.

Proof. If report ..cd... and get d, then report ..dc... and get g, if g > {c,d} then "cd" type deviates; else, "dc" type deviates.

Consistency (C)

- ***If every agent .. a ≻ b .. then don't pick b.***

Proof. Suppose pick b. Still pick b if all a>b > ... (M)

Onto, so exists v with x(v)=a. Still pick a if all a>b>... (M).

Contradiction.

Impossibility

- *"1 is a dictator on a": if 1 reports a top, a picked*

Simple proof (T ⇒ D) for 2-agent case

(Svensson'99)

Monotonicity (M)

- abcdef → badcfd

  v_i v'_i

If $\boldsymbol{x(v_i, v'_{-i}) = d}$***, then*** $\boldsymbol{x(v'_i, v'_{-i}) = d}$.

Proof. If report ..cd... and get d, then report ..dc... and get g, if g > {c,d} then "cd" type deviates; else, "dc" type deviates.

Consistency (C)

- ***If every agent .. a ≻ b .. then don't pick b.***

Proof. Suppose pick b. Still pick b if all a>b > ... (M)

Onto, so exists v with x(v)=a. Still pick a if all a>b>... (M).

Contradiction.

Impossibility

- *"1 is a dictator on a": if 1 reports a top, a picked*

P1: a>b>c; b>a>c.
Can't pick c (C). Consider a.

P2: a > b > c; b > c > a
Can't pick c (C). Can't pick b (T). Select a.

Consider any P3, top(1)=a
Pick a (M, consider P2 -> P3)
Argue 1 also dictator on b, c.

Outline

- Classical mechanism design
  - Preliminaries (DRMs, revelation principle)
  - Positive results – Groves, Single-parameter (Myerson)
    - Min makespan
  - Negative results – Gibbard-Satterthwaite
- Algorithmic mechanism design
  - Knapsack auction
  - Price-of-anarchy analysis

Algorithmic Mechanism Design

- New concern is to obtain computational tractability as well as incentive compatibility
- Emphasis also placed on bidding languages, preference elicitation.

Knapsack auction

(Mu'alem and Nisan'08)

- m items, agent i value w_i for Q_i units (known)
- Goal: maximize total value. 0-1 knaspack problem. NP-hard
- Can't use VCG. 2-approx: $\frac{Z_{opt}(w)}{Z_m(w)} \leq 2$, all w

Knapsack auction

(Mu'alem and Nisan'08)

- m items, agent i value w_i for Q_i units (known)
- Goal: maximize total value. 0-1 knaspack problem. NP-hard
- Can't use VCG. 2-approx: $\frac{Z_{opt}(w)}{Z_m(w)} \leq 2$, all w
- x: order by decreasing w'_i/Q_i.
- If $\sum_k w'_i \geq \max w'_i$ sell $\{1 \ldots k\}$ else sell to h
- Charge critical value (Myerson)

Knapsack auction

(Mu'alem and Nisan'08)

- m items, agent i value w_i for Q_i units (known)
- Goal: maximize total value. 0-1 knaspack problem. NP-hard
- Can't use VCG. 2-approx: $\frac{Z_{opt}(w)}{Z_m(w)} \leq 2$, all w
- x: order by decreasing w'_i / Q_i.
- If $\sum_k w'_i \geq \max w'_i$ sell $\{1 \ldots k\}$ else sell to h
- Charge critical value (Myerson)

- Example: \$5@2, \$6@1, \$6@3, \$12@5; supply 5 units
- Compare (6+5,12) -> allocated to A4. Pay \$11.
- Suppose A2 reports 8? Now {1,2} allocated. A2 pays \$7.

Knapsack auction - Analysis

(Mu'alem and Nisan'08)

- Theorem. Truthful and 2-approx.

Knapsack auction - Analysis

(Mu'alem and Nisan'08)

- Theorem. Truthful and 2-approx.
- Monotone: (Case 1) Allocated and in {1..k}. Still in. (Case 2) Allocated and highest. May cause {1..k} to win but still in.

Knapsack auction - Analysis

(Mu'alem and Nisan'08)

- Theorem. Truthful and 2-approx.

- Monotone: (Case 1) Allocated and in {1..k}. Still in. (Case 2) Allocated and highest. May cause {1..k} to win but still in.

- 2-approx: suppose $k < n$.
- $Z_{opt} \leq Z_{opt}^{R} = \sum_{j=1}^{k} w_j + \gamma\, w_{k+1} \leq \sum_{j=1}^{k+1} w_j \leq \sum_{j=1}^{k} w_j + \max_j w_j = Z_{1..k} + Z_h \leq 2\max(Z_{1..k}, Z_h) = 2Z_m$

Outline

- Classical mechanism design
  - Preliminaries (DRMs, revelation principle)
  - Positive results – Groves, Single-parameter (Myerson)
    - Min makespan
  - Negative results – Gibbard-Satterthwaite
- Algorithmic mechanism design
  - Knapsack auction
  - Price-of-anarchy analysis

Price of anarchy + MD

- PoA: worst-case ratio of optimal obj to obj in equilibrium
- Extension theorems (Roughgarden, 09, 12; Lucier, Paes Leme 11; Syrgkanis 12, Syrgkanis Tardos 13)

Price of anarchy + MD

- PoA: worst-case ratio of optimal obj to obj in equilibrium
- Extension theorems (Roughgarden, 09, 12; Lucier, Paes Leme 11; Syrgkanis 12, Syrgkanis Tardos 13)

- For auctions:
  - PoA for complete-information auction under property P -> PoA in Bayes Nash equilibrium
  - PoA for complete-information auction under property P -> PoA for composition of auctions.
- Comment: now worry about <u>all</u> equilibrium

Example: Extension from NE to BNE

- For any b, exists b_i' s.t. $\sum_i u_i(b'_i, b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$ (smoothness)
  - Do this under P: b'_i only depends on w_i

Example: Extension from NE to BNE

- For any b, exists b_i' s.t. $\sum_i u_i(b'_i, b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$ (smoothness)
  - Do this under P: b'_i only depends on w_i
- If b is a NE then $UTIL(b) = \sum_i u_i(b_i, b) \geq \sum_i u_i(b'_i, b)$

$$\Rightarrow UTIL(b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$$

$$\Rightarrow SW(b) + (\mu - 1)Rev(b) \geq \lambda\, Opt(w)$$

$$\Rightarrow SW(b) + (\mu - 1)SW(b) \geq \lambda\, Opt(w)$$

$\Rightarrow \mu\, SW(b) \geq \lambda\, Opt(w)$, and $POA \leq \mu / \lambda$

Example: Extension from NE to BNE

- For any b, exists b_i' s.t. $\sum_i u_i(b'_i, b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$ (smoothness)
  - Do this under P: b'_i only depends on w_i
- If b is a NE then $UTIL(b) = \sum_i u_i(b_i, b) \geq \sum_i u_i(b'_i, b)$

$$\Rightarrow UTIL(b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$$
$$\Rightarrow SW(b) + (\mu - 1)Rev(b) \geq \lambda\, Opt(w)$$
$$\Rightarrow SW(b) + (\mu - 1)SW(b) \geq \lambda\, Opt(w)$$
$$\Rightarrow \mu\, SW(b) \geq \lambda\, Opt(w)$$, and $POA \leq \mu / \lambda$

- Extends to BNE immediately

Apply to FPSB auction

- PoA for FPSB is 1! But, want to bound under P.

Apply to FPSB auction

- PoA for FPSB is 1! But, want to bound under P.
- Want: $\sum_i u_i(b'_i, b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$

Apply to FPSB auction

- PoA for FPSB is 1! But, want to bound under P.
- Want: $\sum_i u_i(b'_i, b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$
- For any bids, $u_i\left(\frac{w_i}{2}, b_{-i}\right) + p(b) \geq w_i / 2$
  - Either gain $\frac{w_i}{2}$ from deviation, or $p(b) > \frac{w_i}{2}$

Apply to FPSB auction

- PoA for FPSB is 1! But, want to bound under P.
- Want: $\sum_i u_i(b'_i, b) + \mu\, Rev(b) \geq \lambda\, Opt(w)$
- For any bids, $u_i\left(\frac{w_i}{2}, b_{-i}\right) + p(b) \geq w_i / 2$
  - Either gain $\frac{w_i}{2}$ from deviation, or $p(b) > \frac{w_i}{2}$
- $\Rightarrow u_i\left(\frac{w_i}{2}, b_{-i}\right) + p(b) x_i^*(w) \geq \frac{w_i}{2} x_i^*(w)$
- $\Rightarrow \sum_i u_i(b'_i, b) + Rev(b) \geq \frac{1}{2}\, Opt(v)$; thus PoA $\leq$ 2

Direction for AMD?

- Lucier and Borodin (2010)
- First price Single-minded Combinatorial auction
- Optimal allocation rule, the PoA is *m* (*m* items)
- But, if the allocation rule is approximate (sqrt-*m* greedy), then (½,sqrt(m)) – smooth, and O(sqrt(m)) PoA.

- Design mechanisms that are smooth, and provide good worst-case properties.

References

- See "Economics and Computation", Parkes and Seuken CUP (forthcoming, 2014)
- Chapters 8 and 10