Computational Mechanism Design and Auctions

NIPS Tutorial, Vancouver.

David C. Parkes

Harvard University

December 13, 2004

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 simplify problem facing agents

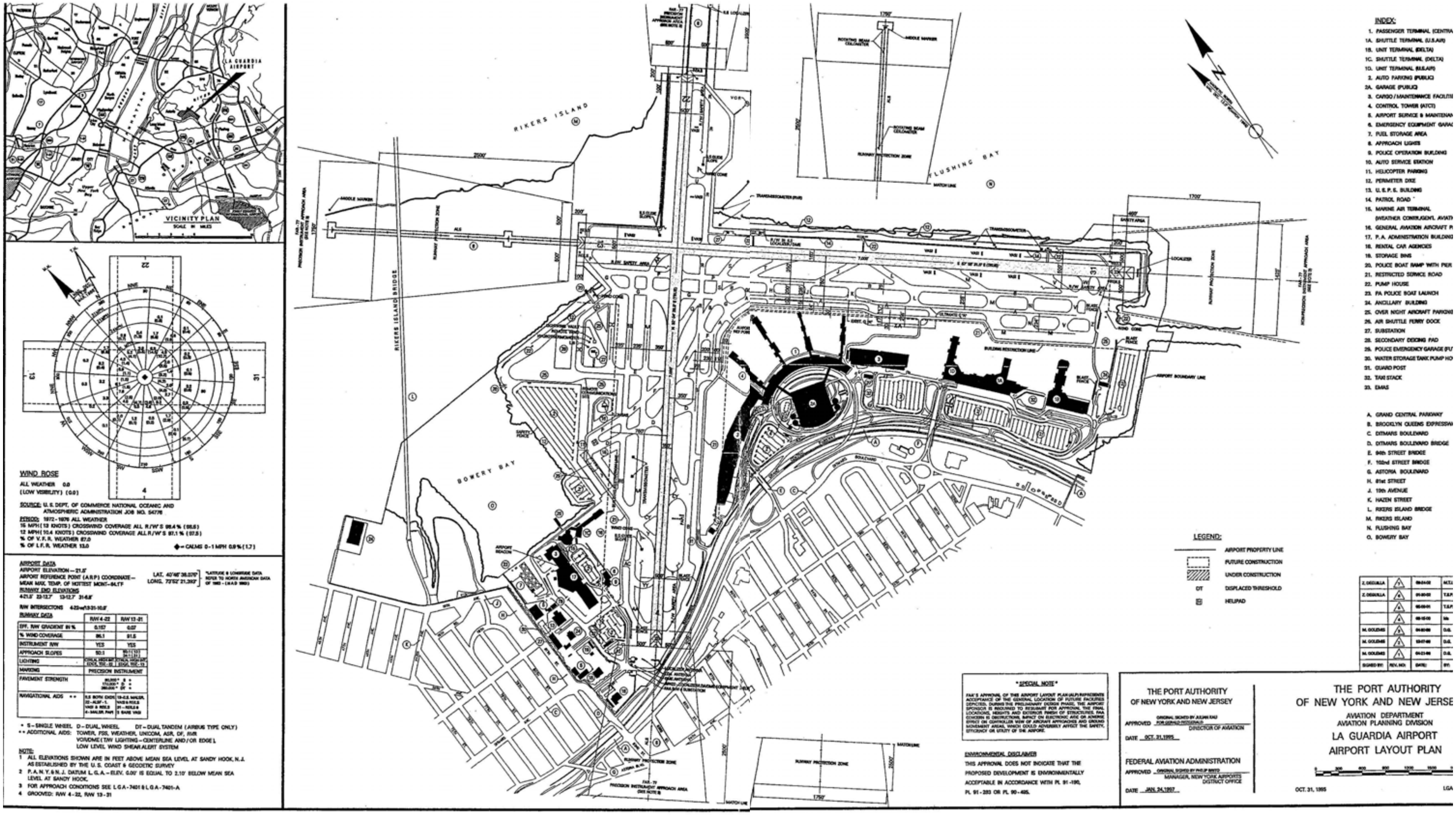
- Auctions: mechanisms for resource allocation
  - typically "detail free"
  - meaning, the rules don't depend on *a priori* knowledge of the auctioneer

- Start with a normative model of agent behavior
- Design rules of a game, so that when agents play as assumed the "right things" happen
- May also try to design for:
  - robust equilibrium
  - minimal information revelation
  - distribtued computation
  - bounded-rational agents
  - adaptive agents...

- Theme: coordinated behavior amongst self-interested agents

Example: Laguardia Airport

(static and centralized MD problem)

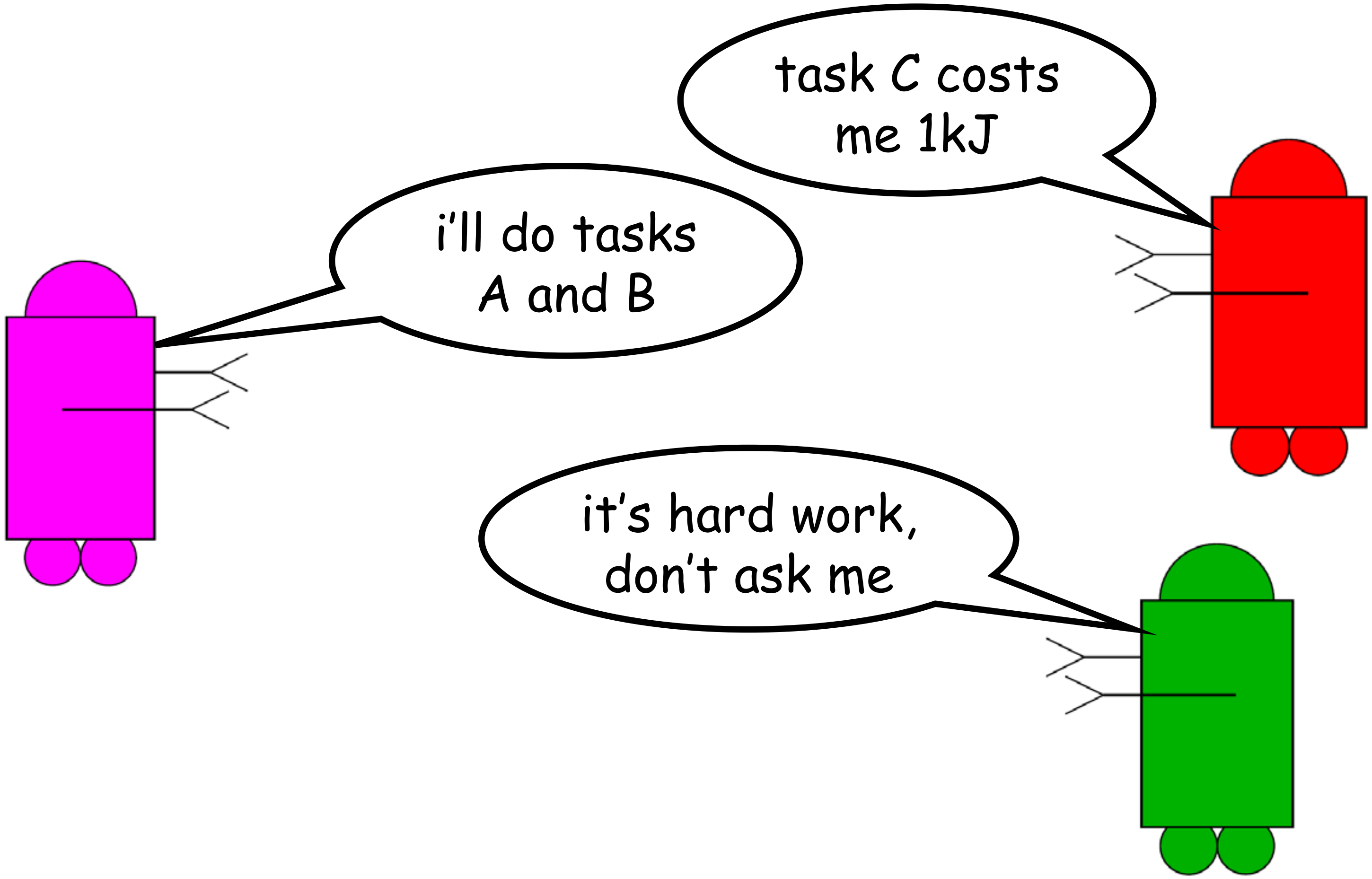


Example: WiFi @ Starbucks

(dynamic and centralized MD problem)

Example: MultiAgent Planning

(static and distributed MD problem)



Example: Hubble Telescope Scheduling
(adaptive and distributed MD problem)

sequential decision problem

CALTECH

RL,
coordination,
self-interest

CS/Econ Analogy

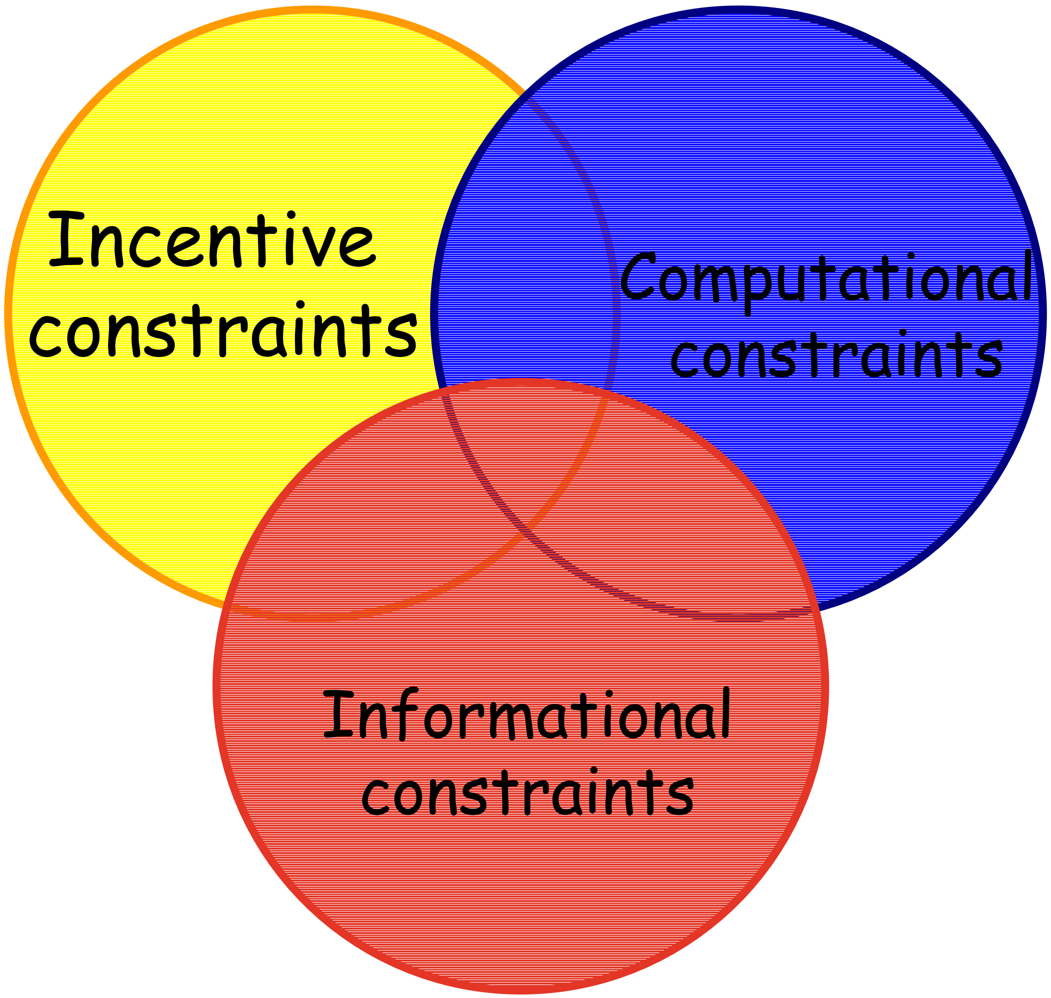
(based on Feigenbaum)

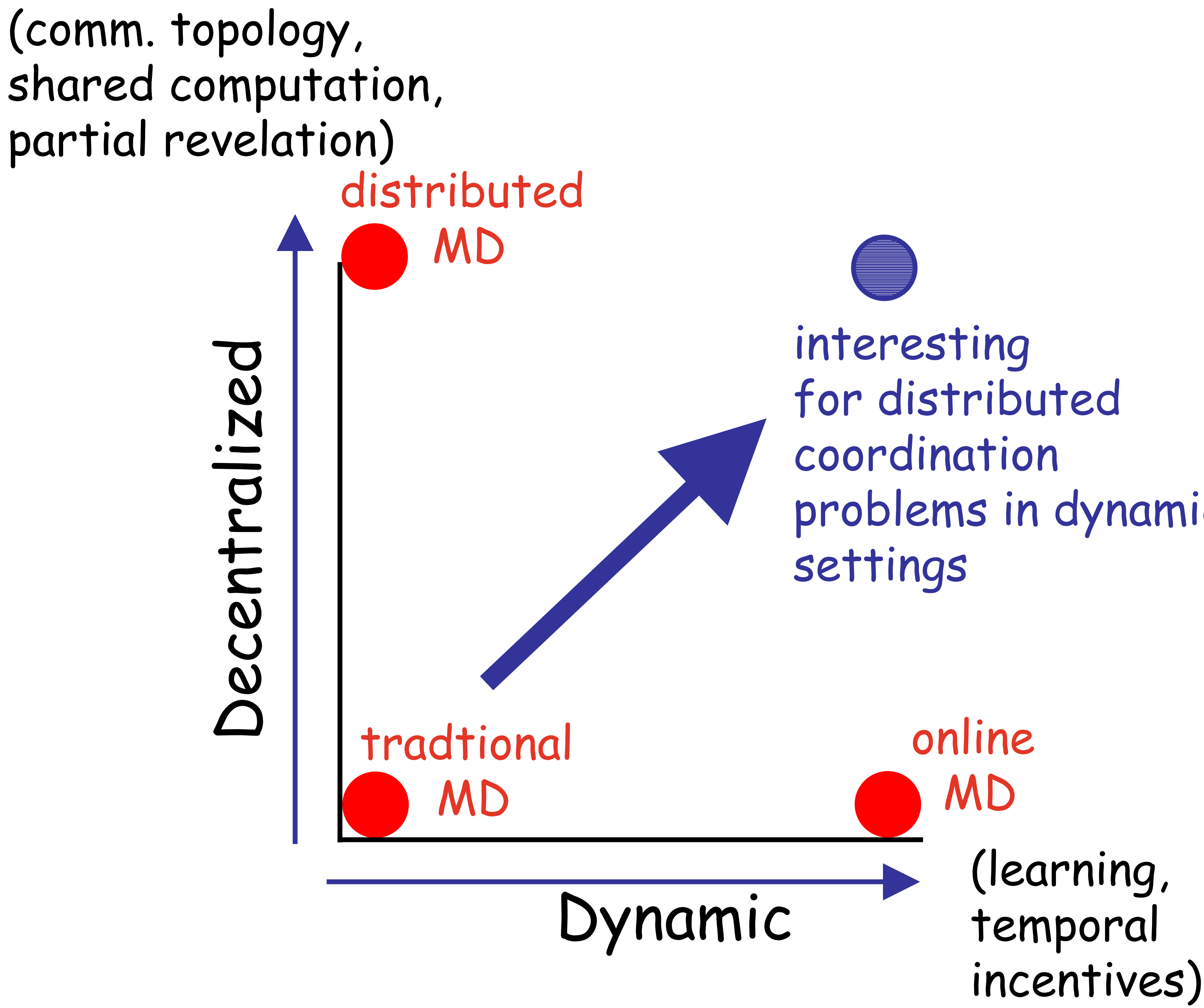
- Agents are cooperative
- Main concern is computational and communication

- Agents are self-interested
- Main concern is incentives

"Computational Mechanism Design"
- brings both together…

CMD: Problem Features




(comm. topology,
shared computation,
partial revelation)
distributed
MD
Decentralized
interesting
for distributed
coordination
problems in dynamic
settings
tradtional
MD
online
MD
Dynamic
(learning,
temporal
incentives)

Outline

- Brief review of GT concepts
- Static & Centralized MD
  - Characterization of truthful mechanisms
  - VCG: a case study
- Static & Decentralized MD
  - Iterative mechanisms: comm & value complexity
  - Distributed implementations
- Dynamic & Centralized MD
  - online mechanism design
- Adaptive & Decentralized MD
  - "market of minds"

Basic Game Theory Model

- Normal Form Game $G = \langle N, (Ac_i), (u_i) \rangle$:
  - $N = \{1,\ldots,n\}$: set of agents
  - Ac_i: set of available actions for agent i
  - u_i: $Ac_1 \times \ldots \times Ac_n \in R$
- Agent i selects a mixed strategy:
  - s_i: $Ac_i \rightarrow [0,1]$, s.t. $\sum_{a_i \in Ac_i} s_i(a_i) = 1$
- Utility function:
  - $u_i(s_1,\ldots,s_n) = \sum_{a \in Ac} u_i(a_1,\ldots,a_n) \prod_{i \in N} s_i(a_i)$
- A strategy profile s^* is a Nash Equilibrium if:
  - $u_i(s'_i, s^*_{-i}) \leq u_i(s^*_i, s^*_{-i})$, $\forall\, i \in N$, $\forall\, s'_i$

Notation

- Vector notation
  - notation $s=(s_1,\ldots,s_n)$
  - notation $s_{-i}=(s_1,\ldots,s_{i-1},s_{i+1},\ldots,s_n)$
- Equilibrium notation:
  - $s^*=(s^*_1,\ldots,s^*_n)$ denotes an equilibrium

Example: Payoff Matrix game

(x , y)
row column

column

row	L	M	R
U	4,3	5,1	6,2
M	2,1	8,4	3,6
D	3,0	9,6	2,8

Pure NE: (U,L) by iterated elimination of dominated strategies

Example: Payoff Matrix Game

(adapted from Porter et al., 2004)

(x , y)
row column

	0	2/7	3/7	2/7	0
0	2,3	-1,4	2,4	5,2	1,-1
0	2,2	3,0	4,1	-2,4	1,3
2/11	4,6	7,2	2,-2	4,9	2,1
4/11	9,0	-2,6	6,3	7,0	0,5
5/11	3,2	6,1	2,5	5,3	1,0

Mixed NE: [0, 0, 2/11, 4/11, 5/11] maximizes expected payoff, given [0, 2/7, 3/7, 2/7, 0], vice-versa for column player.

Incomplete Information Games

- Private Type: $(\theta_1, \ldots, \theta_n)$
- Strategy: $s_i(\theta_i) \in Ac_i$ (often an infinite action space)
  - function of an agent's type
- Utility: $u_i(s(\theta); \theta_i) \in R$
  - function of an agent's type

- Bayes-Nash equilibrium:
  - joint distribution $f(\theta)$ on agent types
  - agent i plays best-response that maximizes expected-utility, given strategy s_{-i} and model $f(\theta_{-i} | \theta_i)$
- Strategy profile, s^*, is a BNE when:

$$E_{\theta_{-i}}\{u_i(s^*_i(\theta_i), s^*_{-i}(\theta_{-i}); \theta_i)\} \geq E_{\theta_{-i}}\{u_i(s'_i(\theta_i), s^*_{-i}(\theta_{-i}); \theta_i)\}, \ \forall i, \forall \theta_i, \forall s'_i$$

Example: First Price Auction

Value v_i. Agents bid $b_i(v_i)$, receive utility:

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

In the special case of $v_i \sim \text{Uniform}(0,1)$, then a symmetric BNE is $b^*_i(v_i) = \frac{(n-1)}{n} v_i$

Stronger solution concepts

- ex post Nash: s^*_i is best-response whatever the type of other agents:

  $u_i(s^*_i(\theta_i),s^*_{-i}(\theta_{-i});\theta_i)$

  $\geq u_i(s'_i(\theta_i),s^*_{-i}(\theta_{-i});\theta_i), \ \forall\theta_{-i}, \forall\theta_i, \forall i, \forall s'_i$

- Dominant strategy eq.: s^*_i is best-response whatever the type & whatever the strategy of other agents:

  $u_i(s^*_i(\theta_i),s_{-i}(\theta_{-i});\theta_i)$

  $\geq u_i(s'_i(\theta_i),s_{-i}(\theta_{-i});\theta_i), \ \forall s_{-i}, \forall\theta_{-i}, \forall\theta_i, \forall i, \forall s'_i$

DSE $\subseteq$ ex post $\subseteq$ BNE

Example: Second price auction

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

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

The (weakly) dominant strategy is to bid $b^*_i(v_i)=v_i$

Note: don't need to know values, or strategies of other agents.

Vickrey's (1961) celebrated auction.

We'll see examples of ex post Nash eq. later…

Additional Equilibrium concepts…

- Perfect Bayesian Nash Equilibrium (see Osborne & Rubinstein 94)
  - multi-stage game
  - Bayesian-updating about types
  - play rational strategy from each subgame
- Coalitional-proof Nash (Bernheim, Peleg, Whinston 87)
  - a coalition deviates if all members in the coalition are at least as well off, and at least one of them is better off
- False-name bid proof Nash (Yokoo et al. 03)
  - no agent can benefit from participating in a mechanism under multiple identities
- Group Strategyproof (Moulin & Shenker 96)
  - no coalition can usefully deviate, *whatever* the strategies and types of other agents

Outline

- Brief review of GT concepts
- Static & Centralized MD
  - Characterization of truthful mechanisms
  - VCG: a case study
- Static & Decentralized MD
  - Iterative mechanisms: comm & value complexity
  - Distributed implementations
- Dynamic & Centralized MD
  - online mechanism design
- Adaptive & Decentralized MD
  - "market of minds"

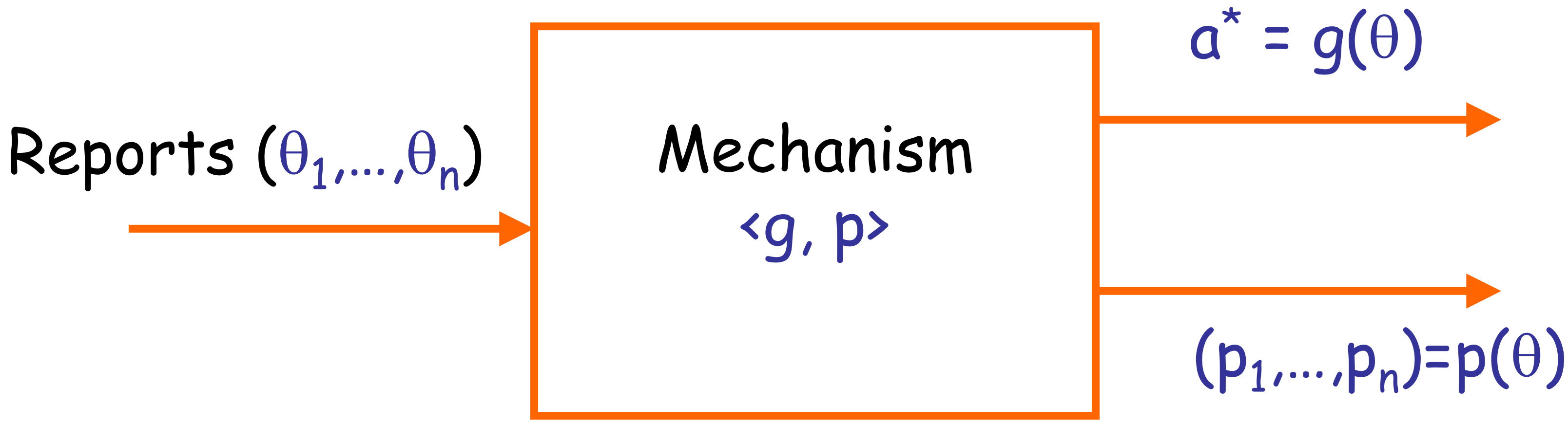
Formal Model of MD

- Set of alternatives $A = \{a,b,\ldots\}$
- Agents $N = \{1,2,\ldots\}$
- Agent i has private information, θ_i
  - the "type" of the agent
  - defines a value function $v_i(a;\theta_i) \in R$ for alternative $a \in A$
- Utility $u_i(a,p)=v_i(a;\theta_i)-p$ for alternative a at price p

- Goal: implement social choice function, $scf(\theta) \in A$
  - e.g. choose a^* to max $\sum_i v_i(a;\theta_i)$

(sometimes care about revenue as well)

Truthful Mechanisms



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

truthful = truth-revelation is a **dominant-strategy** equilibrium
aka "strategyproof"

(relax sometimes)

Suppose a Center:

- Agents announce private type θ_i, perhaps untruthfully
- Center chooses a choice $a=g(\theta)\in A$
- Center chooses payments $(p_1,\ldots p_n)=p(\theta)\in R^n$
- Center enforces outcome.

Canonical Problem: Combinatorial Auction

- Goods G
- 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:
  - $EFF(\theta)$, allocation to solve $\max_S \sum_i v_i(S_i,\theta_i)$

- Applications:
  - logistics, MBA course scheduling, wireless spectrum, school lunches in Chile, …

The Celebrated VCG Mechanism

(Vickrey 61, Clarke 71, Groves 73)

VCG mechanism:

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

Theorem. The VCG mechanism is truthful and implements $EFF(\theta)$.

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

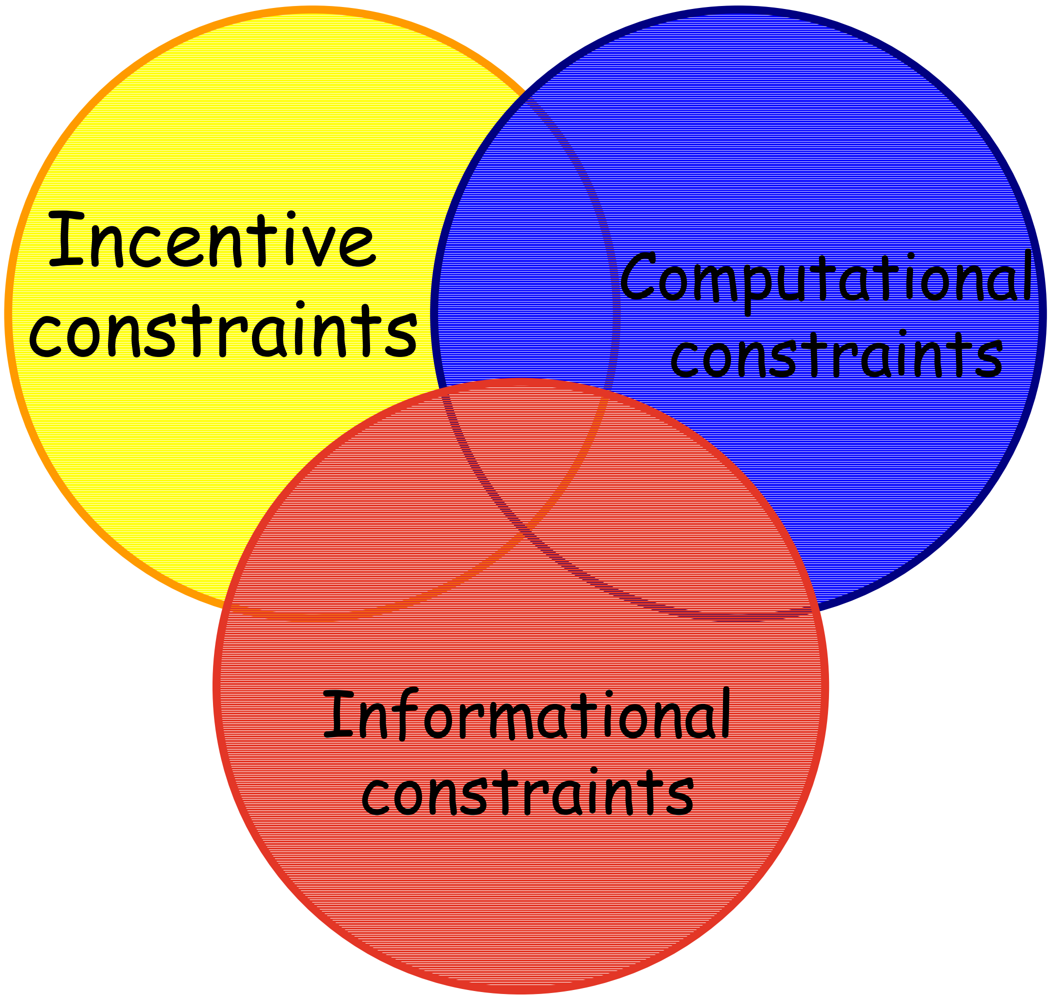
agents

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

- Truthful:

$$u_i(\theta_i,\theta_{-i})=v_i(a^*;\theta_i)+\sum_{j\neq i}v_j(a^*;\theta_j)-\sum_{j\neq i}v_j(a^{-i};\theta_j)$$

CMD: Problem Features



Tension: Truthful & Tractable

- VCG is a very general method for MD.
- But, $g(\theta) = \arg\max_{a \in A} \sum_i v_i(a;\theta_i)$ is often intractable.
- Approximate $g'(\theta) \neq g(\theta)$ causes incentives to unravel.
- No longer finds a^* to maximize utility to agent.
  - (for "price-based" interpretation, prices still agent-independent but no longer chooses agent-maximal alternative).

Example: Single-Minded CAs

(still NP hard)

- Single-minded: $\theta_i = \langle w_i, S_i \rangle$ 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
  - use VCG-based payment scheme

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

Implement (A, ∅, B).

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

Payment by 2: 0 ← should overstate value!

Payment by 3: 10 - 10 = 0

Not truthful!

Price-Based Characterization

(e.g. Segal 02, Bartal et al. 03, Lavi et al. 03, Yokoo 03)

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

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

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

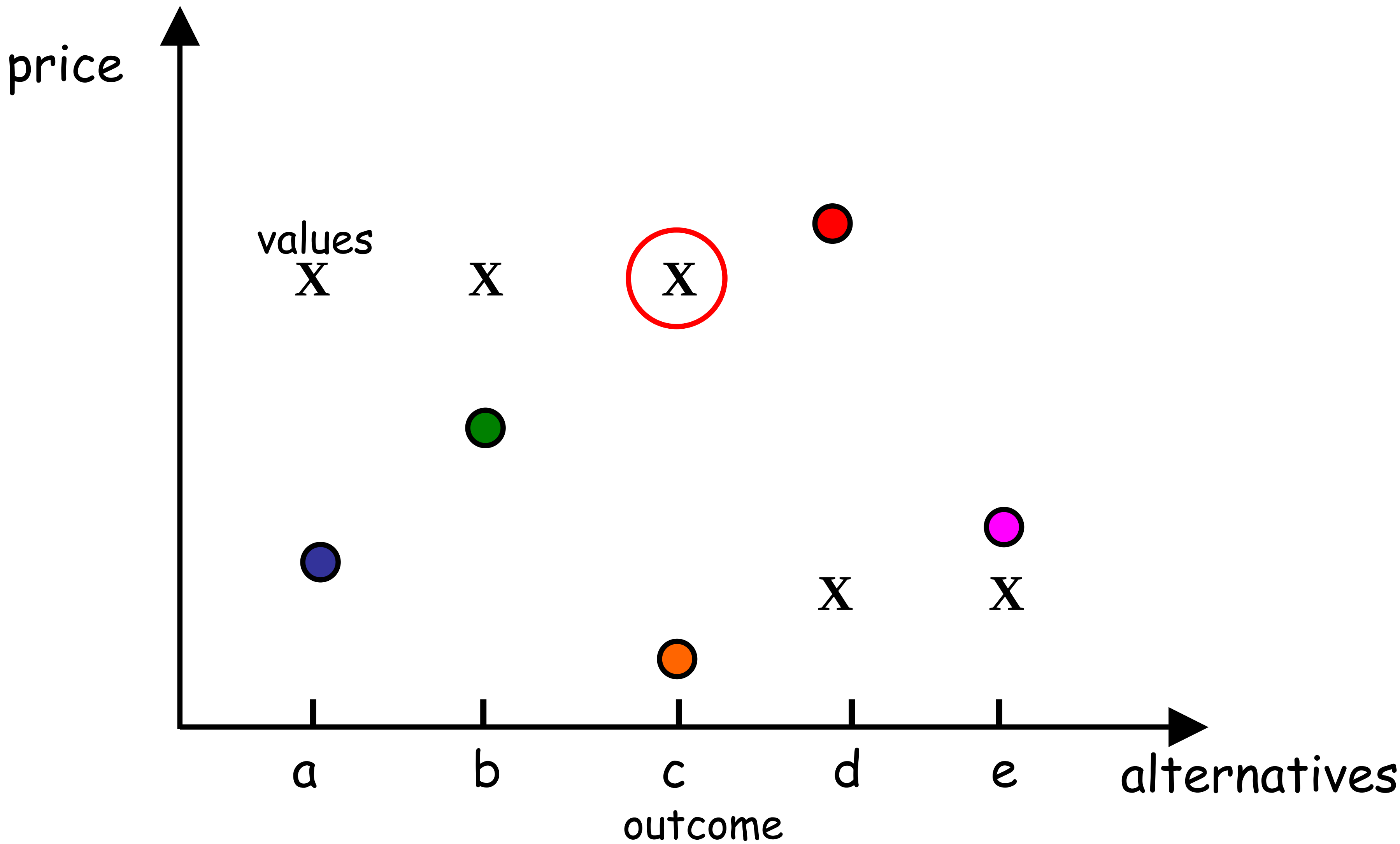
Proof:

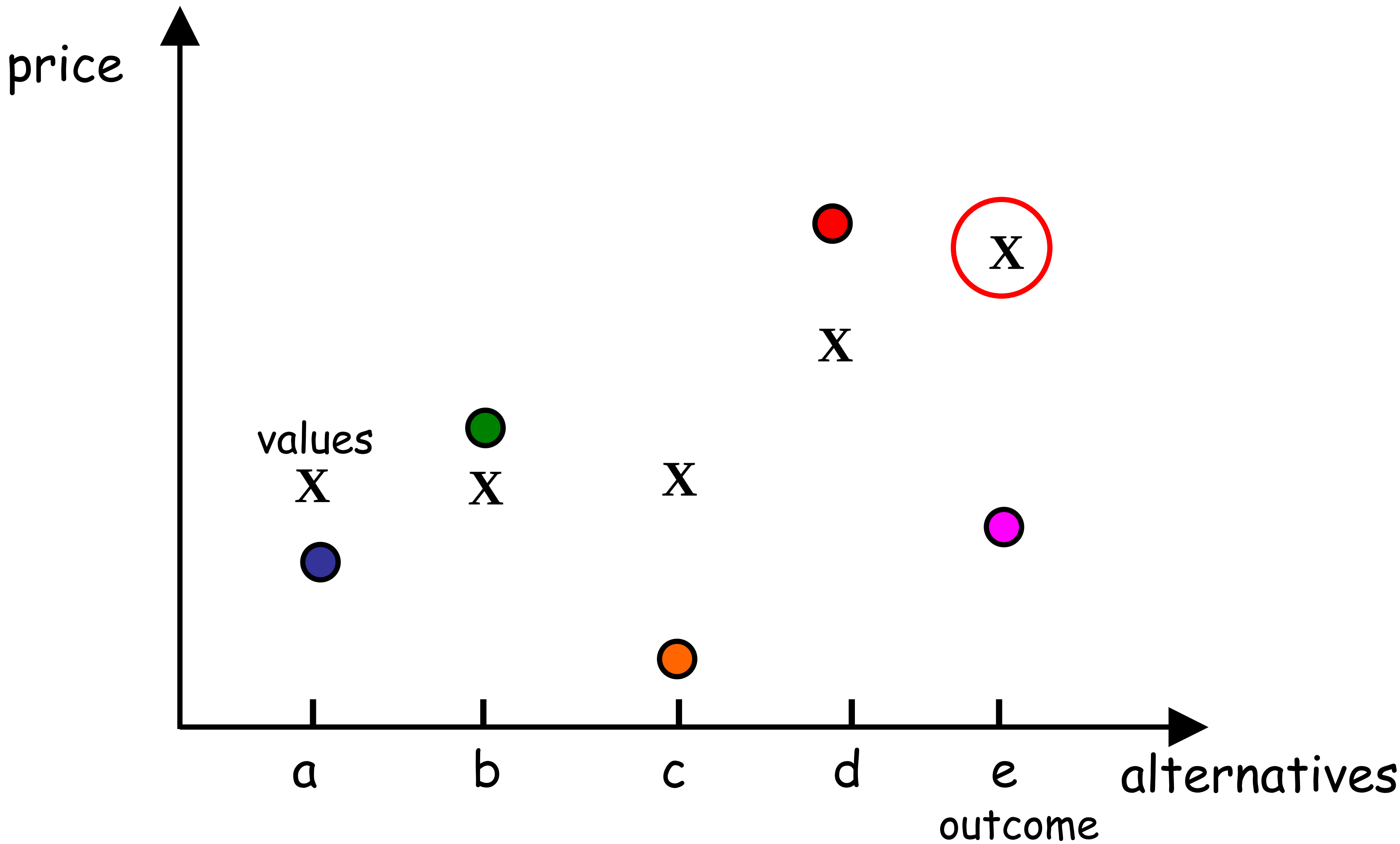
(sufficient) Agent i cannot change π_i, and maximizes utility $u_i(a,\pi_i(a,\theta_{-i}))$ by reporting true θ_i

(necessary)

1) Show $p_i(\theta)$ is agent-independent. Suppose some θ, some $\theta'_i \neq \theta_i$, with $g(\theta)=g(\theta'_i,\theta_{-i})=a$, but $p_i(\theta)\neq p_i(\theta'_i,\theta_{-i})$. Wlog, $p_i(\theta)> p_i(\theta'_i,\theta_{-i})$, and agent should declare θ'_i. Contradiction.

2) 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. Now, 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 $b\neq a$. Contradiction with truthfulness, agent should declare θ'_i for which $g(\theta'_i,\theta_{-i})=b$.


price
values
X
X
X
X
X
a
b
c
d
e
alternatives
outcome


price
values
X
X
X
X
X
a
b
c
d
e
alternatives
outcome

E.g. Price-based Interpretation of VCG

Define: Agent-independent price:

$$\pi_i(a,\theta_{-i}) = \min \{v_i(a;\theta'_i) \mid \theta'_i \in \Theta_i,\ g(\theta'_i,\theta_{-i}) = a\},\ \text{(if possible)}$$

$$= \infty,\ \text{otherwise}$$

Proof. Let $a^*=g(\theta)$.

(1) Agent-independent. Show $p_{vcg,i}(\theta)=\pi_i(g(\theta),\theta_{-i})$, $\forall\ \theta$. First, suppose $\exists\ \theta_{-i}$, θ_i s.t. $v_i(a^*;\theta_i)<\sum_{j\neq i}v_j(a^{-i};\theta_j)-\sum_{j\neq i}v_j(a^*;\theta_j)$. But, then $\sum_j v_j(a^*;\theta_j)<\sum_j v_j(a^{-i};\theta_j)$ and contradiction with $a^*=g(\theta)$. Then, define $\delta=\sum_j v_j(a^*;\theta_j)-\sum_{j\neq i}v_j(a^{-i};\theta_j)$. Construct θ'_i s.t. $v_i(a;\theta'_i)=\max\{0,v_i(a;\theta_i)-\delta\}$, for all $a\in A$. Optimal choice $g(\theta'_i,\theta_{-i})=a^*$.

(2) Agent-maximal. Suppose $b\neq a^*$ s.t. $v_i(b;\theta_i)-\pi_i(b,\theta_{-i}) > v_i(a^*;\theta_i)-\pi_i(a^*,\theta_{-i})$. Then, $v_i(b;\theta_i)-\sum_{j\neq i}v_j(a^{-i};\theta_j) + \sum_{j\neq i}v_j(b;\theta_j) > v_i(a^*;\theta_i) -\sum_{j\neq i}v_j(a^{-i};\theta_j) + \sum_{j\neq i}v_j(a^*;\theta_j)$, and $v_i(b;\theta_i)+\sum_{j\neq i}v(b;\theta_j) > v_i(a^*;\theta_i)+\sum_{j\neq i}v(a^*;\theta_j)$, a contradiction with $a^*=g(\theta)$.

Single-minded: Truthful Mechanism

(Lehmann et al. 2003)

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

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

  $= 0$, otherwise
- 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

agent-independent and agent-maximal $\Rightarrow$ Truthful!

Price-based Interpretation of LOS

- Let $g(\theta)=(S_1,\ldots,S_n)$ denote allocation.
- Price-Based Demonstration of truthfulness:

  (1) Agent-independent payment.

  Price is independent of bid price, the MINIMAL bid to still win.

  $p_i(\theta)=\pi_i(S_i,\theta_{-i})=\min\{w'_i \in R : \theta'_i=\langle w'_i,S_i\rangle, g_i(\theta'_i,\theta_{-i}) = S_i\}$ when $g(\theta)=S$.

  (2) Agent-maximal.

  For a **winner**, $\pi_i(S_i,\theta_{-i})=|S_i|\cdot(w_j/S_j)\leq w_i$, where j is the displaced bid. (because greedy algorithm considers bids in order of per-unit price.)

  For a **loser**, must have $\pi_i(S_i,\theta_{-i}) > w_i$ because greedy algorithm is **monotonic**, and continues to allocate for all $w'_i\geq\pi_i(S_i,\theta_{-i})$.

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.

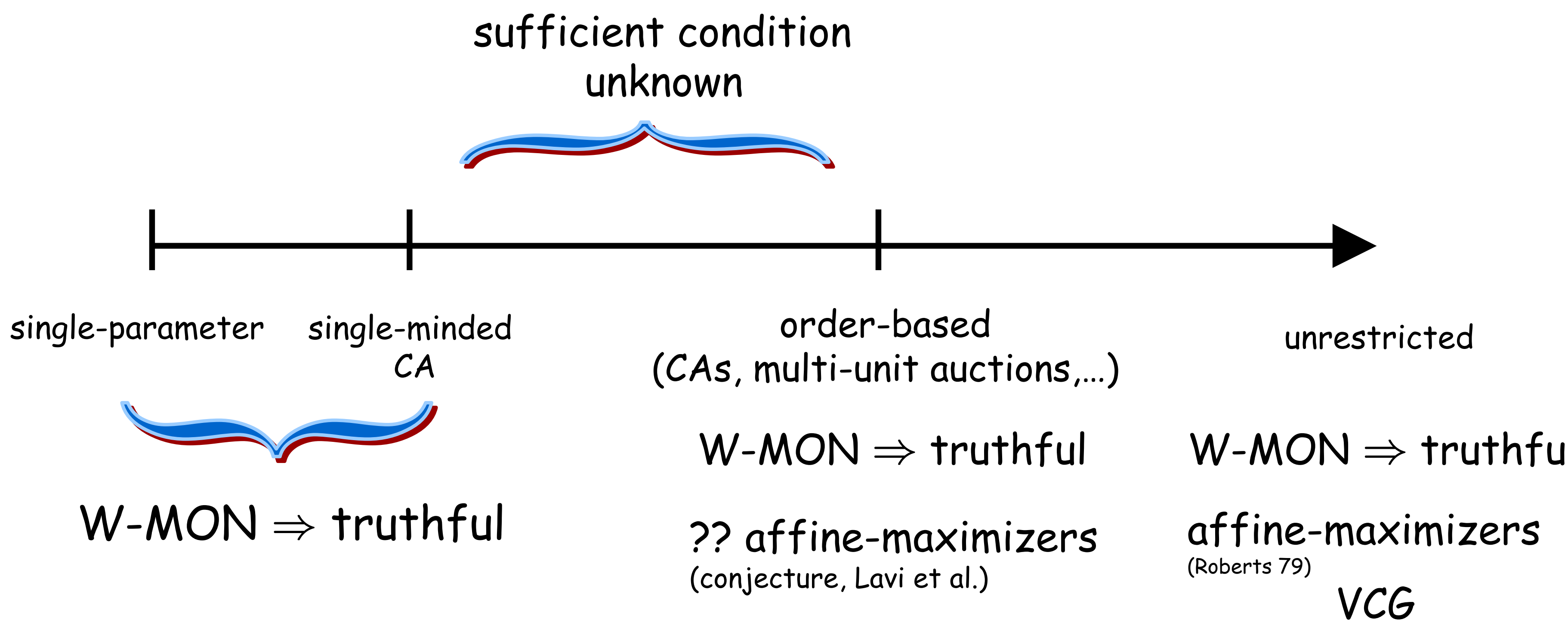
- All single-minded greedy allocation rules $g(\cdot)$ that sort by $w_i / |S_i|^k$ for $k \geq 0$ are monotonic.

- For all such allocation rules g, can use "critical value" **payment rule**, $p_i(\theta) = \pi_i(S_i, \theta_{-i}) = \min \{ w'_i \in R : \theta'_i = \langle w'_i, S_i \rangle, g_i(\theta'_i, \theta_{-i}) = S_i \}$, for $g(\theta) = (S_1, \ldots, S_n)$, and will be **truthful**.

Characterization of truthful $g(\theta)$

"Weak-monotonicity" is necessary for truthfulness

(Muller & Vohra'03; Lavi, Mu'alem and Nisan'03)



Affine-maximizer: $g(\theta) = \arg\max_a \sum_i c_i \cdot v_i(a,\theta_i) + c(a)$

Order-based Domains

- W-MON: $g(v_i,v_{-i})=a$, $g(u_i,v_{-i})=b \Rightarrow u_i(b)-u_i(a) \geq v_i(b)-v_i(a)$
  - cannot change from a to b unless value on b increases
  - Necessary for truthful. (Follows from simple IC argument)
  - Suppose $g(v_i,v_{-i})=a$ and $g(u_i,v_{-i})=b$. Then, $v_i(a)-\pi_i(a,v_{-i}) \geq v_i(b)-\pi_i(b,v_{-i})$ and $u_i(b)-\pi_i(b,v_{-i}) \geq u_i(a)-\pi_i(a,v_{-i})$, which gives $u_i(b)-u_i(a) \geq v_i(b)-v_i(a)$.

- Order based: Domain of types Θ defined in terms of $R_i(a,b) \in \{=,<,\leq,>,\geq,\perp\}$ and $Null \subset A$ s.t. $\theta_i \in \Theta_i \Leftrightarrow$
  - $v_i(a;\theta_i)=v_i(b;\theta_i)$ $\forall a, b$ s.t. $R_i(a,b)$ = "="
  - $v_i(a;\theta_i)<v_i(b;\theta_i)$ $\forall a, b$ s.t. $R_i(a,b)$ = "<"
  - …
  - $v_i(a;\theta_i)=0 \ \forall a \in Null$
  - $v_i(a;\theta_i)$ and $v_i(b;\theta_i)$ can be in any relation when $R_i(a,b)$="$\perp$"

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 $S_i^a \subseteq S_i^b$

- **Theorem.** (Lavi, Mu'alem & Nisan). W-MON is necessary and sufficient for truthfulness in order-based domains.
  - Typically, set $\pi_i(a,\theta_{-i})=\min\{ v_i(a;\theta'_i) \mid g(\theta'_i,\theta_{-i})\in E_i(a)\}$
    - $E_i(a)$ is set of equivalent alternatives to a, i.e. with $R_i\{a,b\}$= "=".
    - "critical value" for a

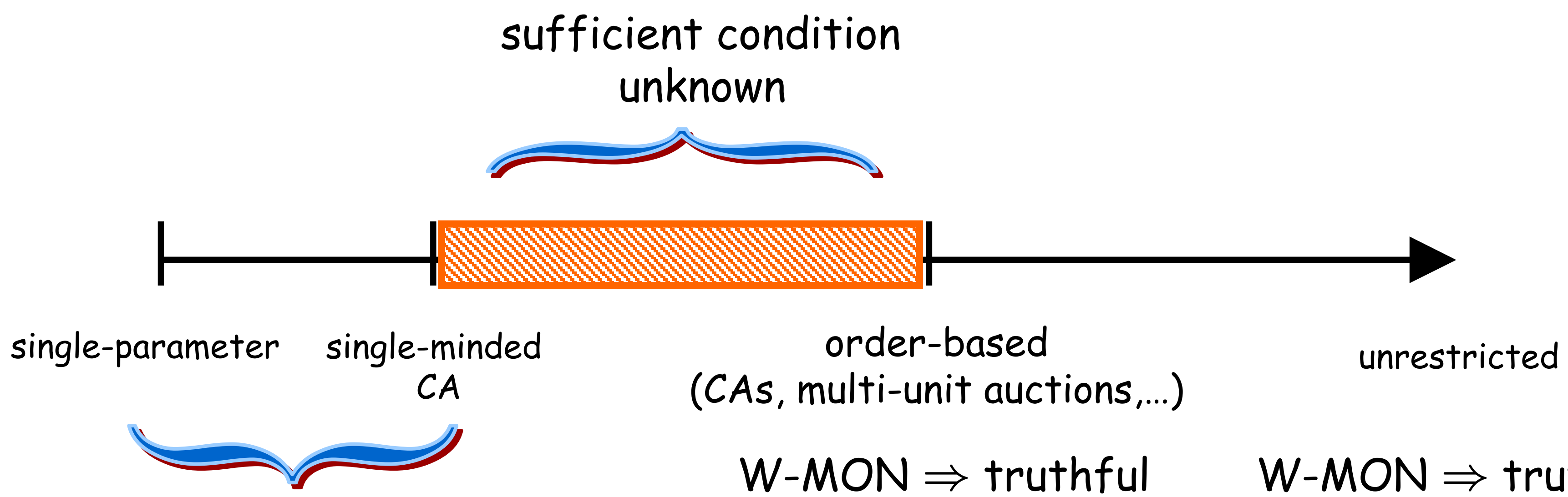
- Order-based domains include:
  - combinatorial auctions, set $R_i(a,b)$= "$\leq$" for $a_i \subseteq b_i$ and $R_i(a,b)$ = "=" for $a_i=b_i$ and $a \in Null$ for $a_i=\emptyset$.
  - multi-unit auctions

$\Rightarrow$ can construct truthful & tractable mechanisms for order-based domains by designing W-MON algorithms.

Characterization of truthful $g(\theta)$

"Weak-monotonicity" is necessary for truthfulness

(Muller & Vohra'03; Lavi, Mu'alem and Nisan'03)



W-MON ⇒ truthful

W-MON ⇒ truthful

?? affine-maximizers

(conjecture, Lavi et al.)

W-MON ⇒ truthful

affine-maximizers

(Roberts 79)

VCG

Affine-maximizer: $g(\theta) = \arg\max_a \sum_i c_i \cdot v_i(a,\theta_i) + c(a)$

Gaps in Characterization

- Double-minded CA:
  - agents have value for two different bundles
- Online auctions:
  - agents have value to win an item during a time period $a_i \leq t \leq d_i$
  - recent paper by Hajiaghayi et al. fills in gap
- Majority of structured allocation problems:
  - e.g. settings where bidders need resources in some range, e.g. "ratio of CPU allocation and Memory allocation must be between 0.8 and 1.2"

$\Rightarrow$ interesting to close these gaps, and develop tractable & truthful mechanisms for practical domains

Outline

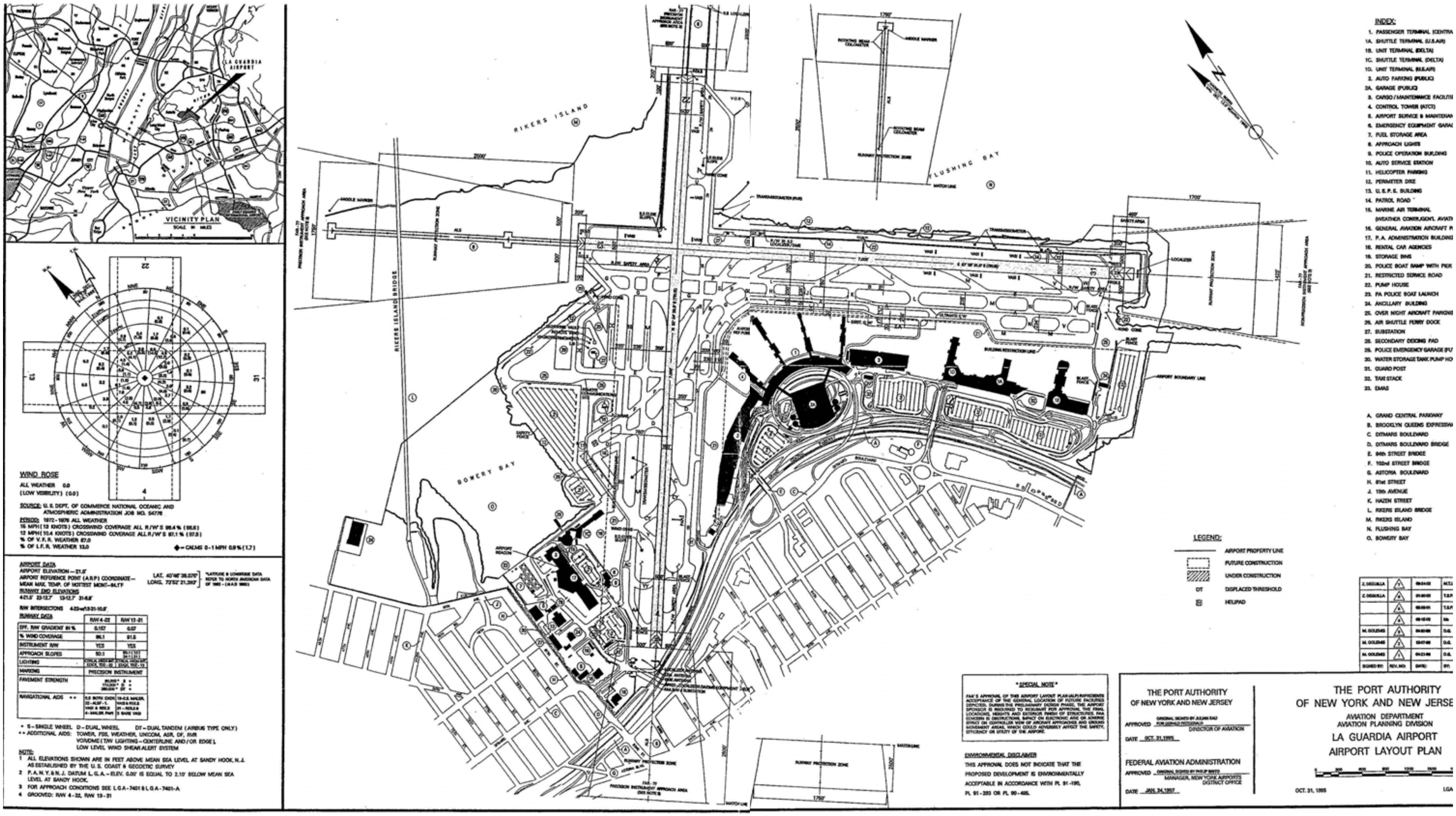
- Brief review of GT concepts
- Static & Centralized MD
  - Characterization of truthful mechanisms
  - VCG: a case study
- Static & Decentralized MD
  - Iterative mechanisms
  - Distributed implementations
- Dynamic & Centralized MD
  - online mechanism design
- Adaptive & Decentralized MD
  - "market of minds"

Decentralized Mechanisms

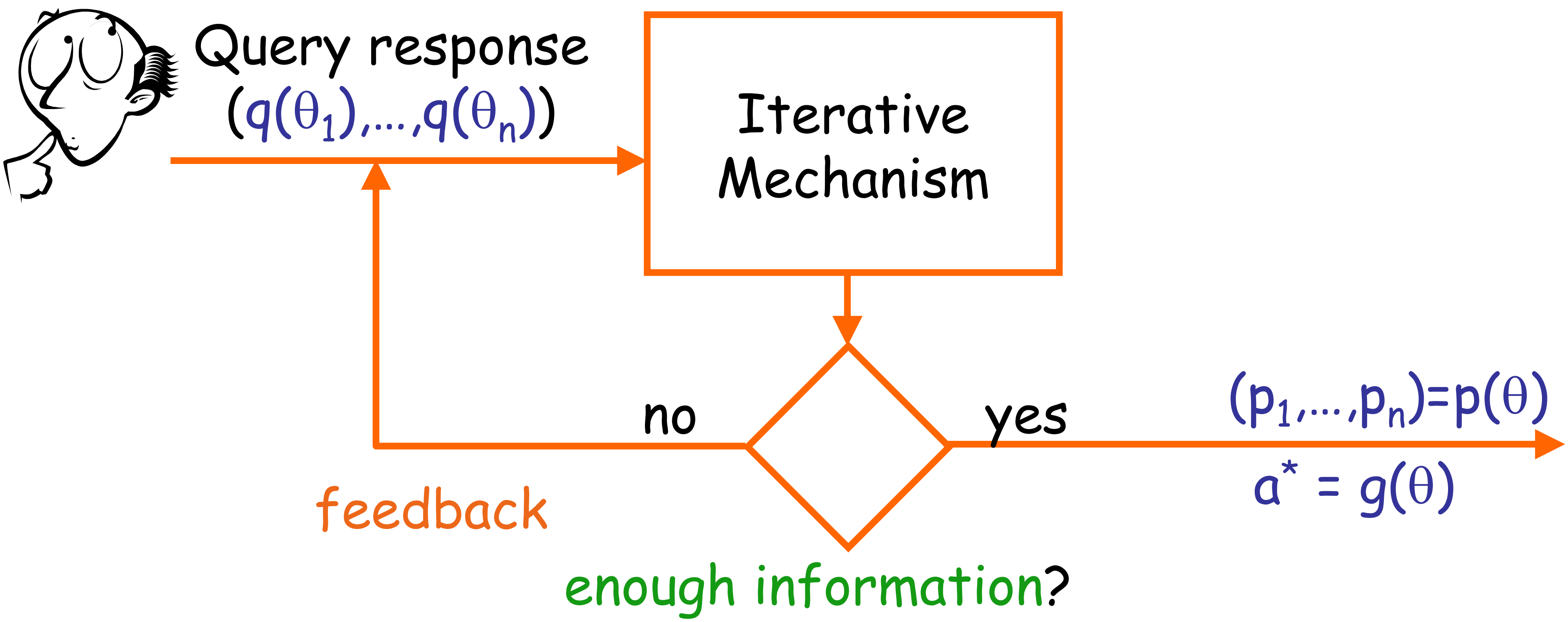
- **Goal**: Determine outcomes without complete type elicitation, perhaps also engage agents in computation.
- Iterative mechanisms:
  - Adaptive elicitation by the center.
  - Terminate early, when $g(\theta')$ and $p(\theta')$ constant for all $\theta' \in \mathbf{F}(\theta)$, given information that $\theta \in \mathbf{F}(\theta) \subseteq \Theta$
- Distributed implementations:
  - use self-interested agents in computation to determine $g(\theta)$ and $p(\theta)$
  - bring computation, communication, and information-revelation into an equilibrium
  - expands strategy space

Example: Laguardia Airport

(elicitation for all bundles of slots is too costly)



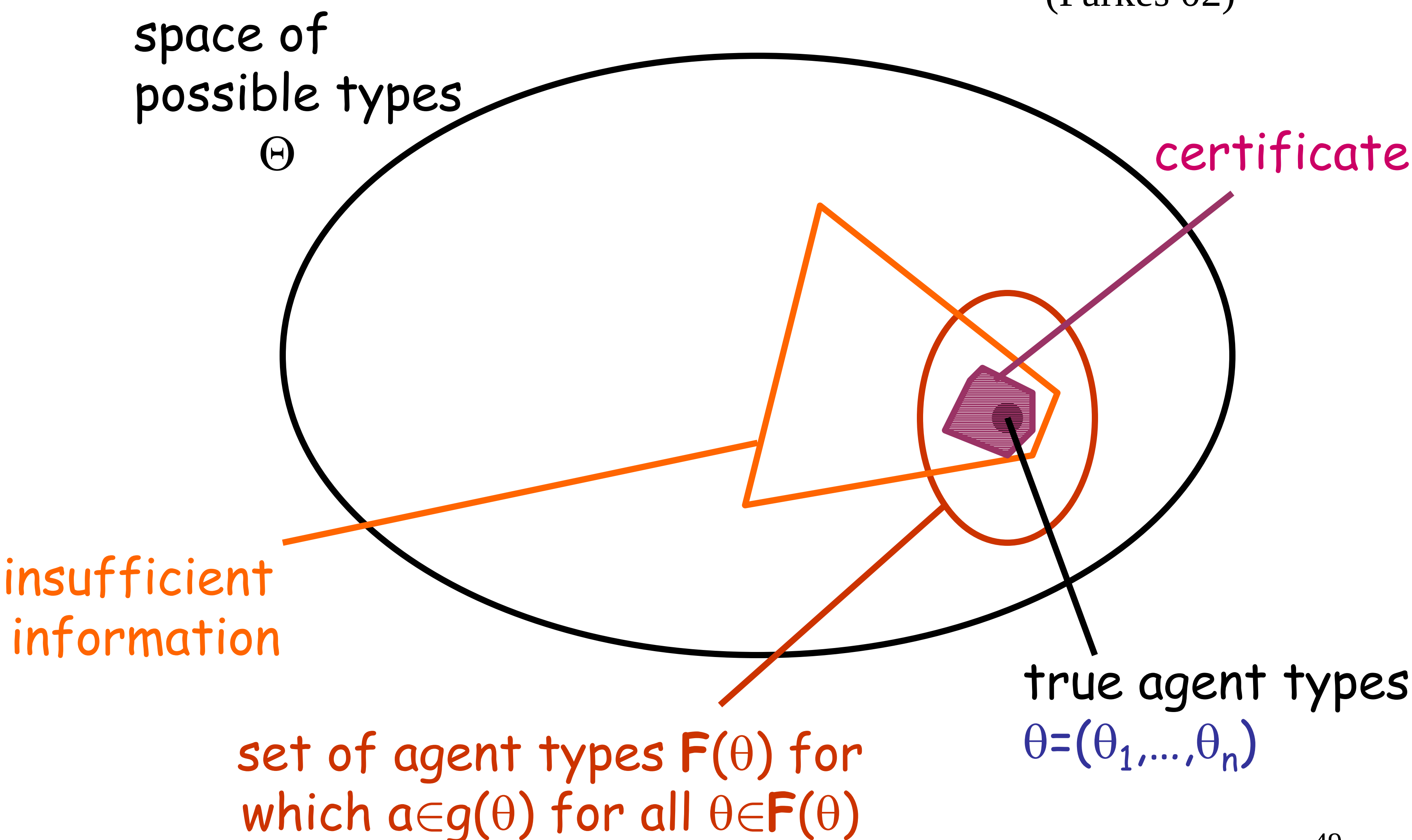
Iterative Mechanisms



truthful = truth-revelation is an ex post Nash equilibrium
"best-response, given other agents play the equilibrium..."

Information Certificates

(Parkes 02)



Example: Single item allocation

- Values. $v_1 = 10$. $v_2 = 6$. $v_3 = 4$.
- Efficient allocation: $g(v)=(A, \emptyset, \emptyset)$.

- Sufficient information:
  - $v_1 \geq 7$; $v_2, v_3 \leq 7$

- For VCG outcome: $p_1(v)=6$, $p_2(v)=p_3(v)=0$
- Sufficient information:
  - $v_1 \geq 6$; $v_2=6$; $v_3 \leq 6$

Queries

- Value(a): what is $v_i(a)$?
- Demand(p): given prices $p=(p(a),p(b),\ldots)$ return some $a' \in \arg\max_{a \in A} \{ v_i(a)-p(a) \}$
- Bounds(a,w): is $v_i(a) \geq w$? is $v_i(a) \leq w$?
- Ordinal(a,b): is $v_i(a) \geq v_i(b)$?
- Rank(k): which alternative is in k-th rank?
- …

Style of Research Question

- For a given scf(θ), what is the minimal information certificate?
  - what is the additional cost of truthfulness?
- For a given scf(θ), what is the minimal elicitation procedure?
  - what is the additional cost of truthfulness?
  - can elicitation be solved with a polynomial number of queries?
- What is the best mechanism given a limited number of queries? (Blumrosen, Segal & Nisan)
- …

Minimal Certificates for CAs are Price-Based

(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 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 $EFF(\theta)$ elicits enough information to determine CE prices.

(Also sufficient).

Query: announce (p, S), where $S=(S_1,\ldots,S_n)$ is seller-optimal. If all agents say "yes", done.

Minimal VCG Certificates

(Lahaie & Parkes 04)

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 $E(N\backslash 1),\ldots, E(N\backslash n)$

Theorem. Any mechanism that implements $EFF(\theta)$ and $VCG(\theta)$ must elicit enough information to determine UCE prices.

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, $p_{uce}=6$.

Style of Research Question

- For a given scf(θ), what is the minimal information certificate?
  - what is the additional cost of truthfulness?
- For a given scf(θ), what is the minimal elicitation procedure?
  - what is the additional cost of truthfulness?
  - can elicitation be solved with a polynomial number of queries?
- What is the best mechanism given a limited number of queries? (Blumrosen, Segal & Nisan)
- …

Iterative CA Paradigms

- Ascending price:
  - report prices, get demand-set information
  - adjust,
  - terminate when “supply=demand”
- Direct elicitation methods:
  - ask queries of bidders
  - continue until know enough information

Ascending price CAs

- Initialize: all prices p set to zero
- While (demand>supply)
  - collect demand sets from each bidder
  - solve WD to maximize revenue
  - increase prices based on bids
- Finally (terminate with CE prices):
  - implement allocation consistent with bids
  - charge final price, or adjusted price

iBundle (Parkes & Ungar 00): efficient for general valuations

ascending-Vickrey auctions (Parkes,Mishra & Ungar 04; Parkes & Mishra 03; de Vries et al. 03)

- terminate with UCE prices

Linear-Programming Based Design

(de Vries et al., Parkes, ...)

- 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...

Iterative CA Paradigms

- Ascending price:
  - report prices, get demand-set information
  - adjust,
  - terminate when "supply=demand"
- Direct elicitation methods:
  - ask queries of bidders
  - continue until know enough information

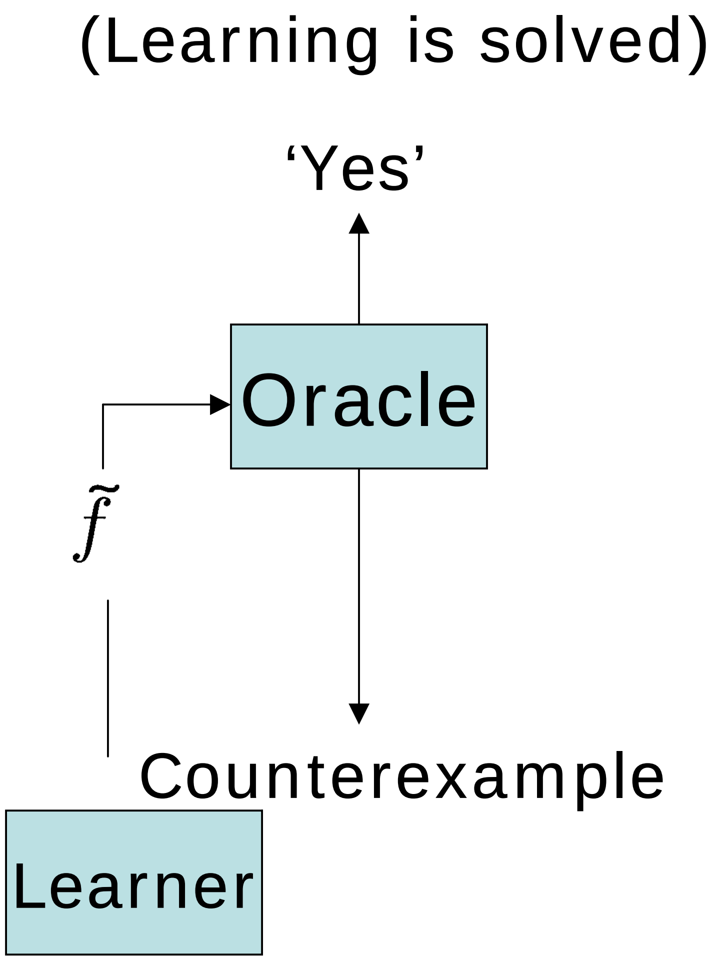
Direct Elicitation

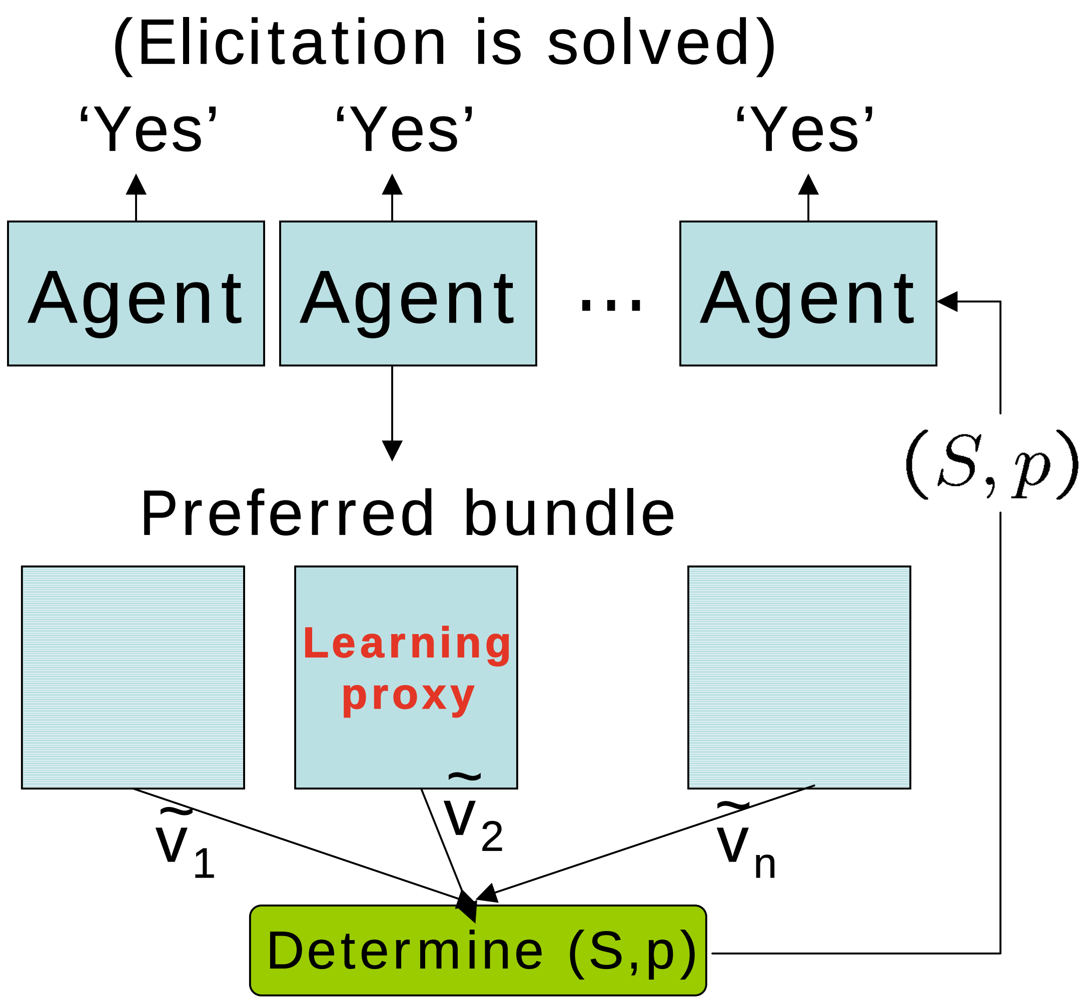
[Hudson & Sandholm, Conen & Sandholm, Parkes, Zinkevich et al., Blum et al., Lahaie & Parkes]

- Can use any interactive algorithm to compute the outcome of a truthful mechanism.
  - need not be an ascending-price auction
- Computational learning theory approach:
  - define a representation language (e.g. XOR, OR*, polynomial, linear-threshold)
  - define a query class (e.g. demand queries, value queries)
- Learn value function v_i in # queries poly(m,t), where t is size of minimal representation
  - and stop early, when have enough for efficient allocation…

Eliciting not Learning: Early-stopping

(Lahaie & Parkes, EC'04)





Stop early when PRICE($\tilde{v}_1$,…$\tilde{v}_n$) are actual CE prices.

Details: Learning with Early-Stop

- Solve WD($v''_1,\ldots,v''_n$), Compute prices($v''_1,\ldots,v''_n$)
- Manifest allocation S'', Manifest CE prices p''
- Query type: Demand(p'',S''_i)
  - **Yes**, done. **Else**, get counterexample (T).
  - Query to find one of {S''_i,T} for which $v''_i \neq v_i$
  - Given counterexample, have efficient method to find a new atomic (as before for XOR; will describe for OR)

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
  $$\tilde{f}(x) = f(x), \quad \forall x \in X$$
- Else presents some input x' such that:
  $$\tilde{f}(x') \neq f(x')$$

Elicitation

- Demand query $\tilde{v}_1, \ldots, \tilde{v}_n$
- Maintain manifest valuations
- Present allocation $(x_1, \ldots, x_n)$ 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)

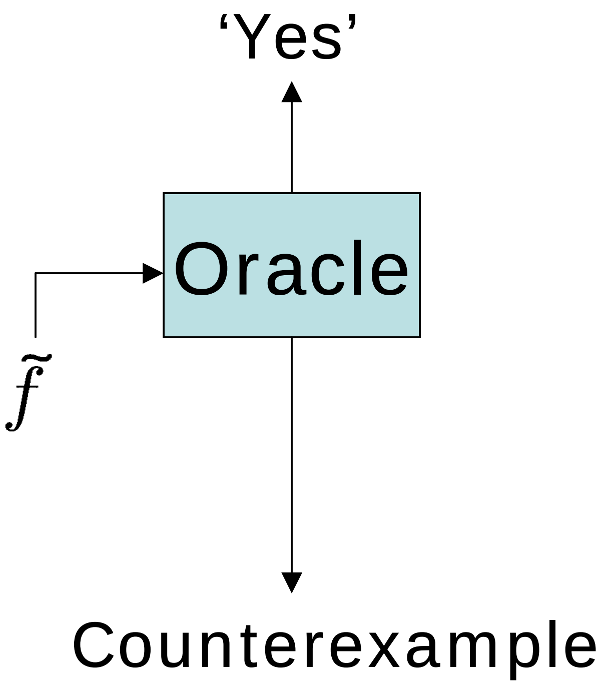
Elicitation

- 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

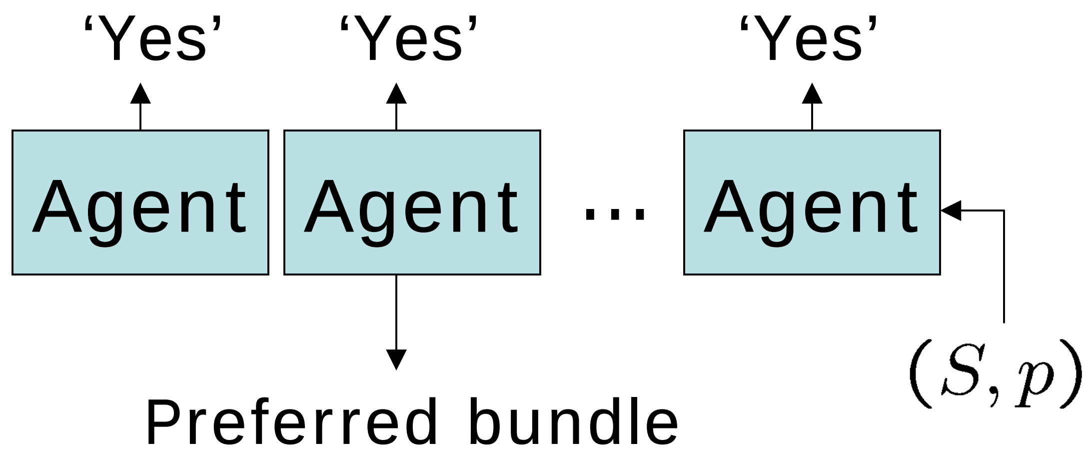
Equivalence

(Learning is solved)



Demand

(Elicitation is solved)

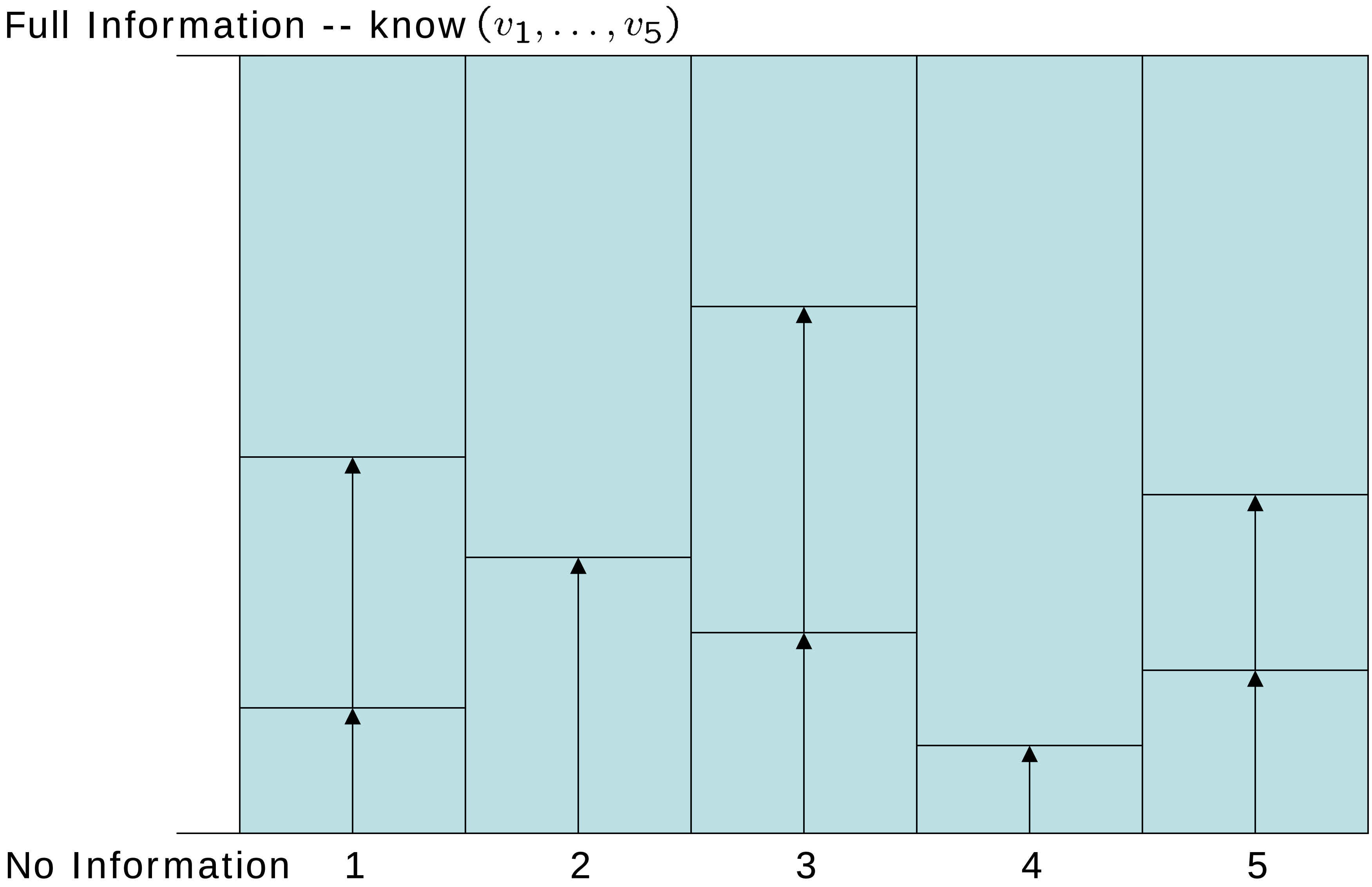


Then

- Preferred bundle, or
- Proposed bundle

is a counterexample.

The Algorithm



Modification: Universal Queries

- Universal Demand Queries <p, $\{S_1, S^{-2}_1, \ldots, S^{-n}_1\}$ >
  - 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

⇒ terminate with UCE prices, and implement VCG outcome

General Characterization

(Lahaie & Parkes)

- Definition. Class $V_1 \times \ldots \times V_n$ can be polynomial-query elicited with value- and demand-queries if EFF(v) can be determined in poly(n,m,size($v_1, \ldots, v_n$)) queries.

Theorem. Valuations can be polynomial-query elicited if they can each be polynomial-query learned.

Theorem. Valuations can be elicited with polynomial communication if they can each be polynomial-time learned.

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
- [Lahaie & Parkes 2004] Relationship between non-linear price demand queries and equivalence queries
  - Fast elicitation for XOR (t+1 demand, mt value), polynomial, and linear-threshold.

Applications

Polynomials: t terms, m goods, n agents

- Schapire & Sellie (93): efficient learning algorithm for polynomials.
- $v_i(S)=a_0 \cdot x_1 + a_1 \cdot (x_1 x_3) - a_2 \cdot (x_1 x_5) + \dots$
- Concise for valuations "almost substitutes"
- $O(nmt)$ demand queries, $O(nmt^3)$ value queries

XOR bids: t terms, m goods, n agents

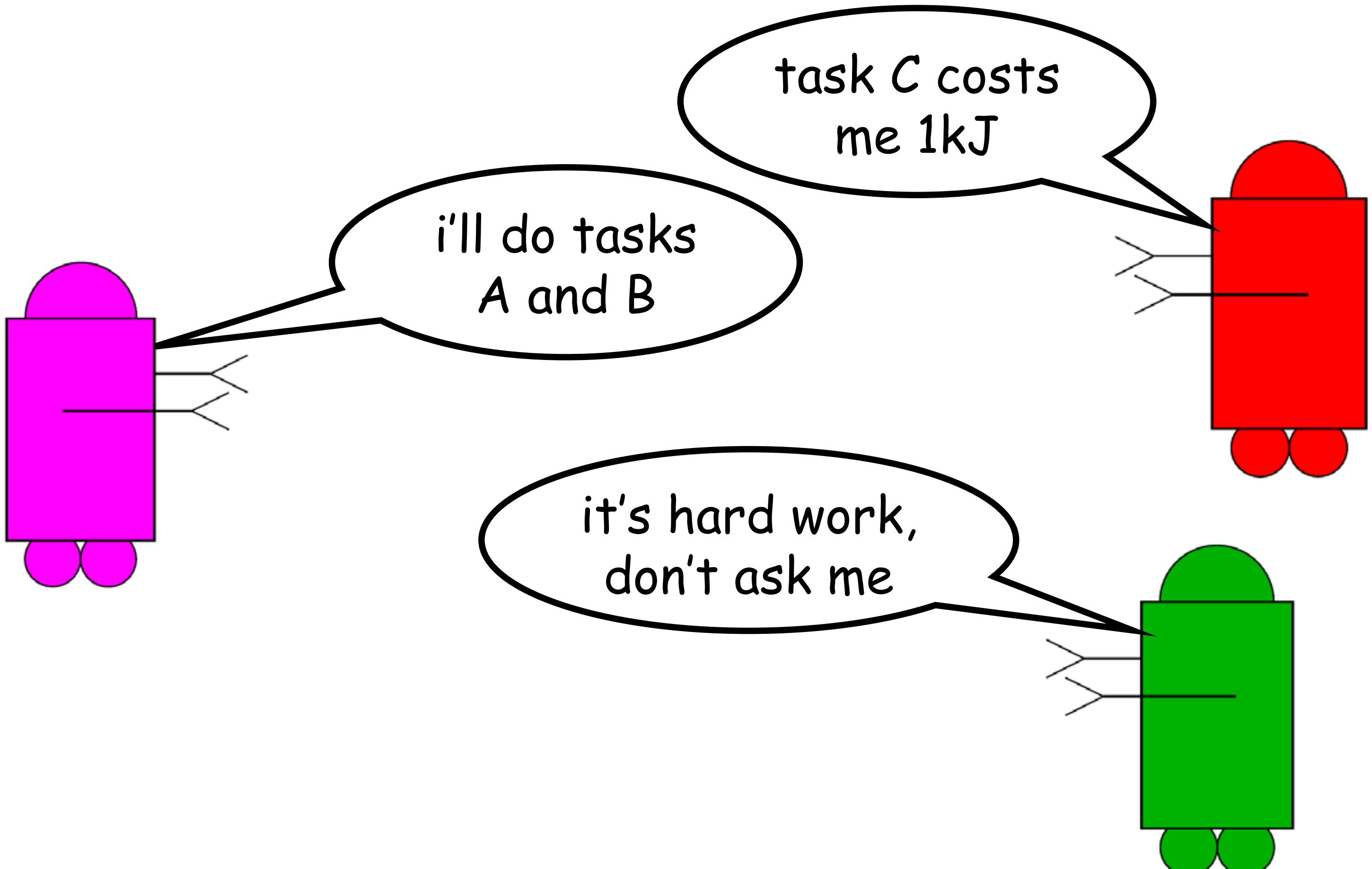
- XOR bids can be efficiently learned, generalizing a learning algorithm for monotone DNF (Angluin 87).
- Values: $\{ \langle b_1, L_1 \rangle, \langle b_2, L_2 \rangle, \dots, \langle b_t, L_t \rangle \}$; $v_i(S)=\max_{L \subseteq S} b_i(L)$
- compact for valuations "almost complements"
- worst-case $t+1$ demand queries, mt value queries

What about involving agents in the computation itself?

- Determine outcomes without complete type elicitation, involve agents in computation.
- Iterative mechanisms:
  - Adaptive elicitation by the center.
  - Terminate early, when $g(\theta')$ and $p(\theta')$ constant for all $\theta' \in F(\theta)$, given information that $\theta \in F(\theta)$
- Distributed implementations:
  - use self-interested agents to compute $g(\theta)$ and $p(\theta)$
  - bring computation, communication, and information-revelation into an equilibrium
  - expands strategy space

Example: MultiAgent Planning

(static and distributed MD problem)



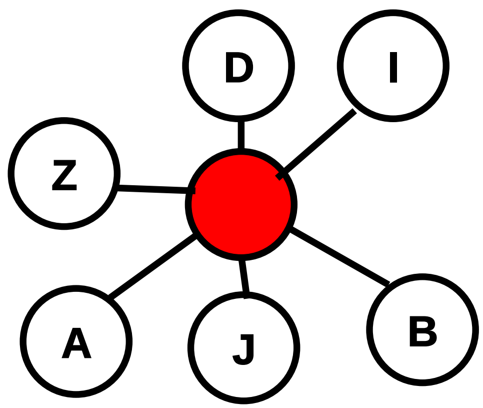
also, p2p networks, Internet algorithms, …

Distributed Implementation

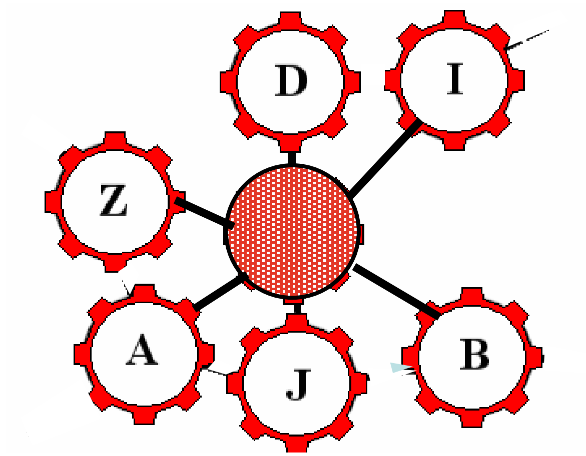
(Monderer & Tennenholtz 99; Feigenbaum et al.02; Feigenbaum & Shenker 02; Parkes & Shneidman 04; Shneidman & Parkes 04)

- Need to bring computation, message-passing, and information-revelation into an equilibrium.
  - expands strategy space!!

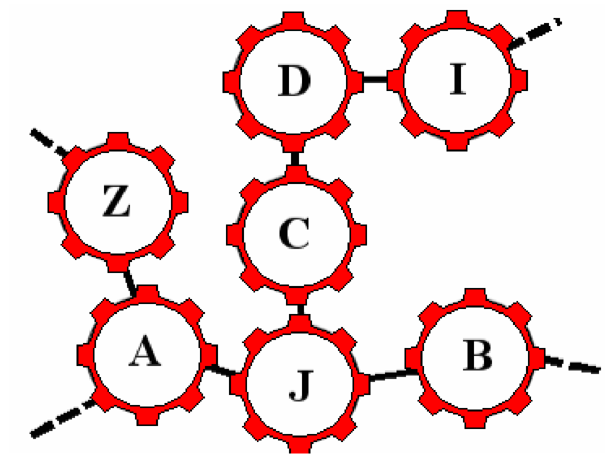
- centralized



- trusted comm, light center



- fully distr.

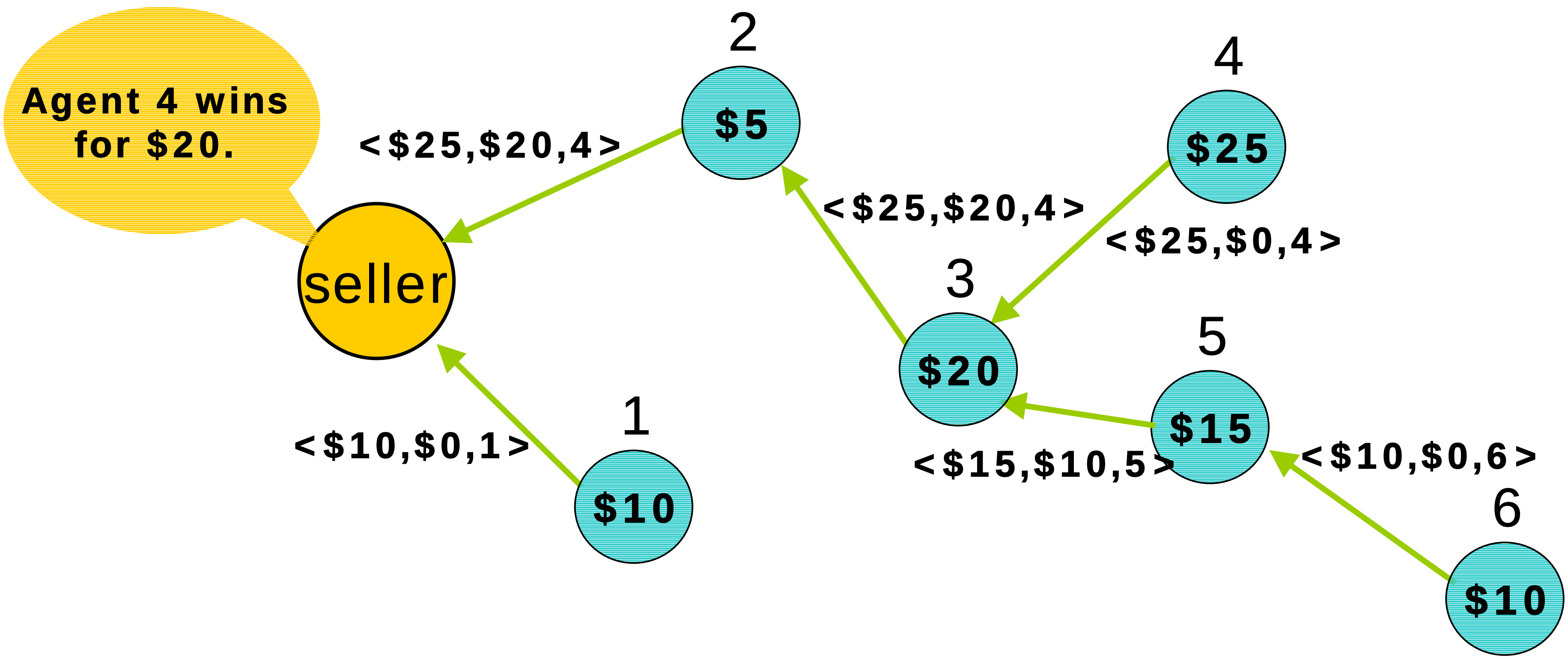


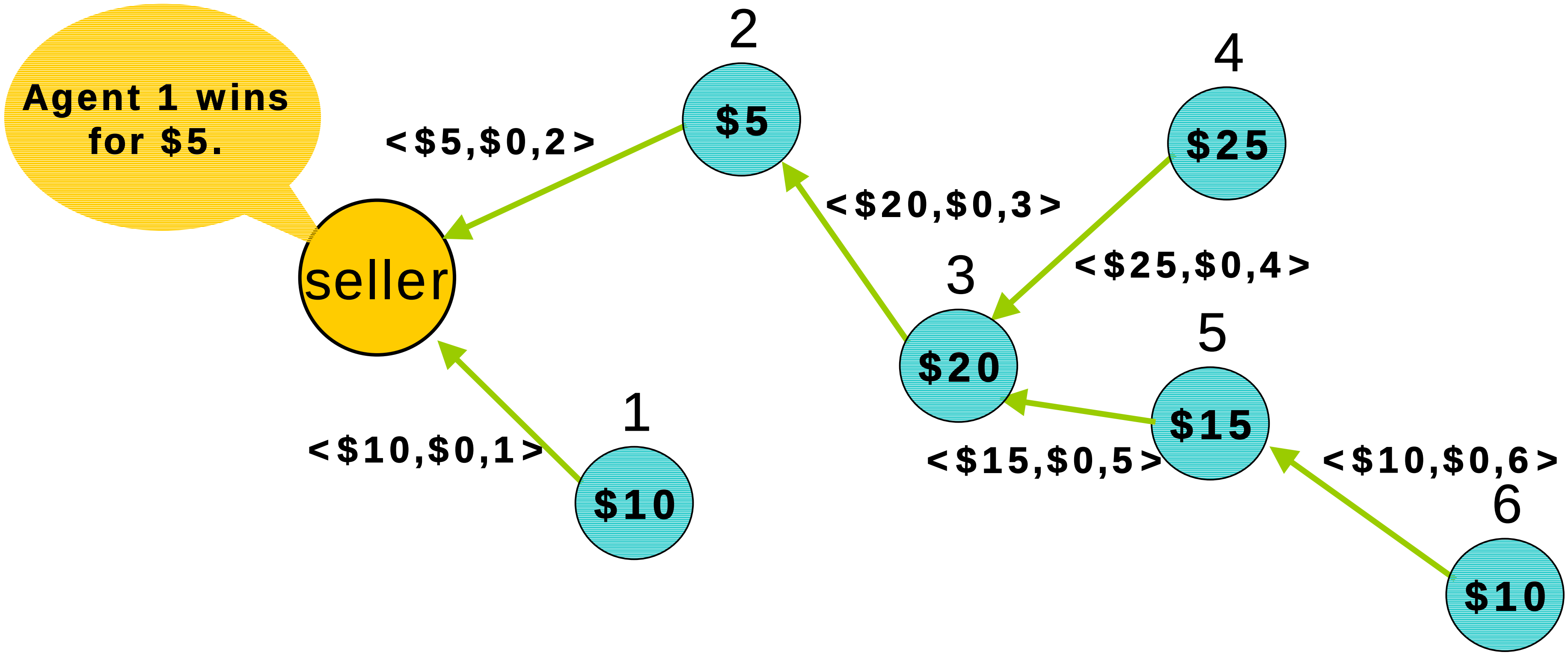
Two Problems

- Computation
  - determining the outcome of the mechanism without a center to receive types and calculate $g(\theta)$ and $p(\theta)$
  - write $f = \langle g, p \rangle$ so that $f(\theta) \in A \times R^n$
- Execution
  - enforcing the outcome of the mechanism, once computed
  - making sure agents follow-through, and make payments

Scenario: Distributed Auction

Example: a second-price auction across a peer-to-peer network.




Agent 1 wins
for $5.
seller
2
$5
<$5,$0,2>
4
$25
<$20,$0,3>
<$25,$0,4>
3
$20
5
$15
1
$10
<$10,$0,1>
<$15,$0,5>
<$10,$0,6>
6
$10

Distr. Implementation

- Private type, $\theta_i \in \Theta_i$
- Utility $u_i(a;\theta_i)$, on alternatives $a \in A$
- SCF, $scf(\theta) \in A$

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

"intended implementation"

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

strategy
$s_i \in \Sigma_i$

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

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

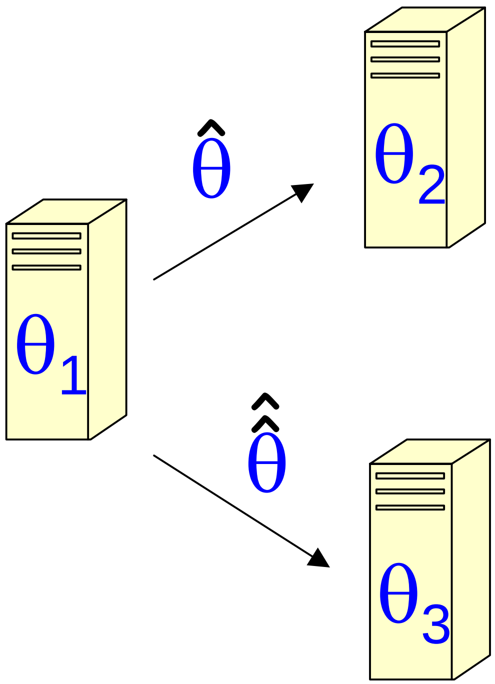
comput. action
"action can affect outcome rule"
(not just info-rev)

(new)

message-passing action, "send a message, unchanged"

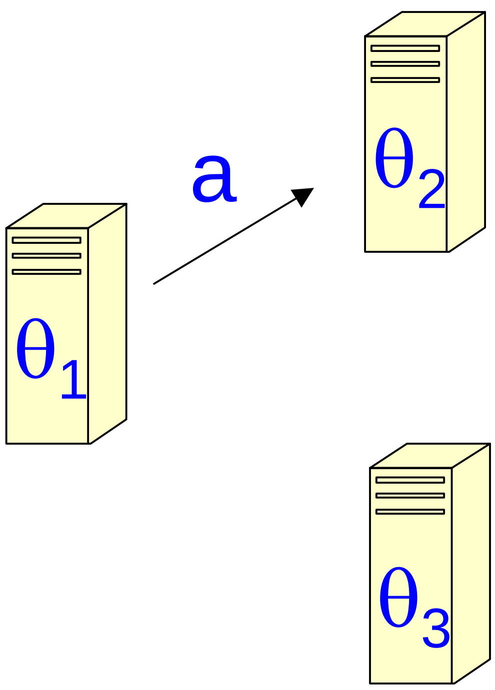
(new)

Information Revelation Action, r_i



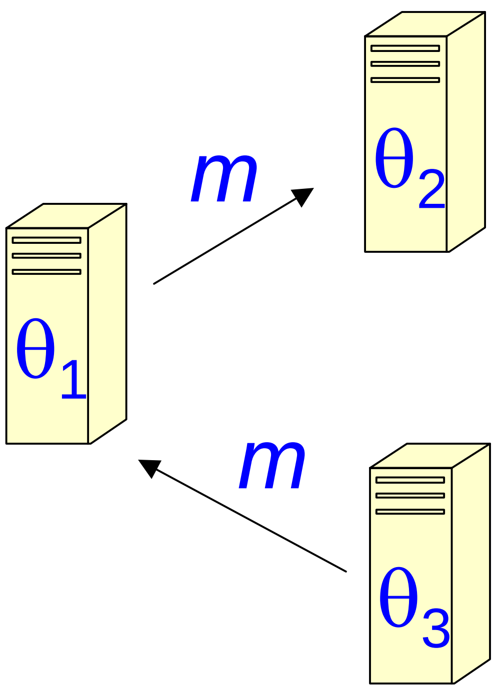
- r_i: reveal private type information to neighbors

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.

i.e., no agent can usefully deviate from s^m_i when all other agents follow s^m_{-i}, whatever θ_i and whatever θ_{-i}

Definition. Incentive compatibility (IC).

- will perform all information-revelation actions truthfully in equilibrium

Definition. Algorithm compatibility (AC).

- will choose to implement the specified computational actions faithfully in equilibrium.

Definition. Communication compatibility (CC).

- will choose to implement the specified communication actions faithfully in equilibrium

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

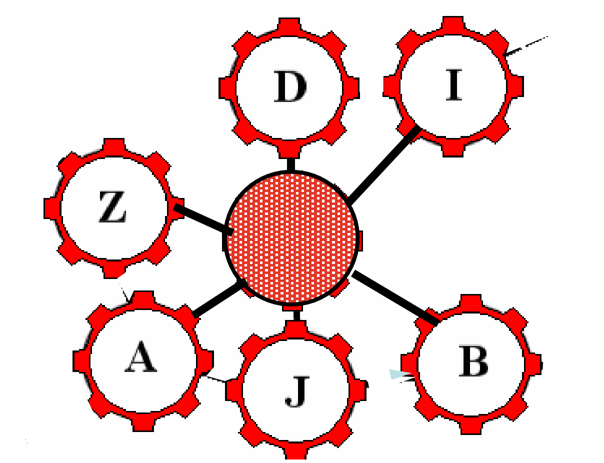
Basic Techniques

(Parkes & Shneidman 04)

- Partitioning: Divide the computation so that agents cannot benefit from deviating from the part of $g(\theta)=f(s^m(\theta))$ for which they are responsible.
- Incentives: Use incentives so that it is in an agent's best interest to faithfully compute $g(\theta)=f(s^m(\theta))$.
- Redundancy: Use multiple agents for same computational step, and go with the "quorum" result. Perhaps punish agents found to deviate.

A Couple of Examples

- Assume communication with the center:
  - No communication-compatibility problems
- Use agents in computation of $g(\theta)$ and $p(\theta)$.



"light center"

Example I: Distributed VCG

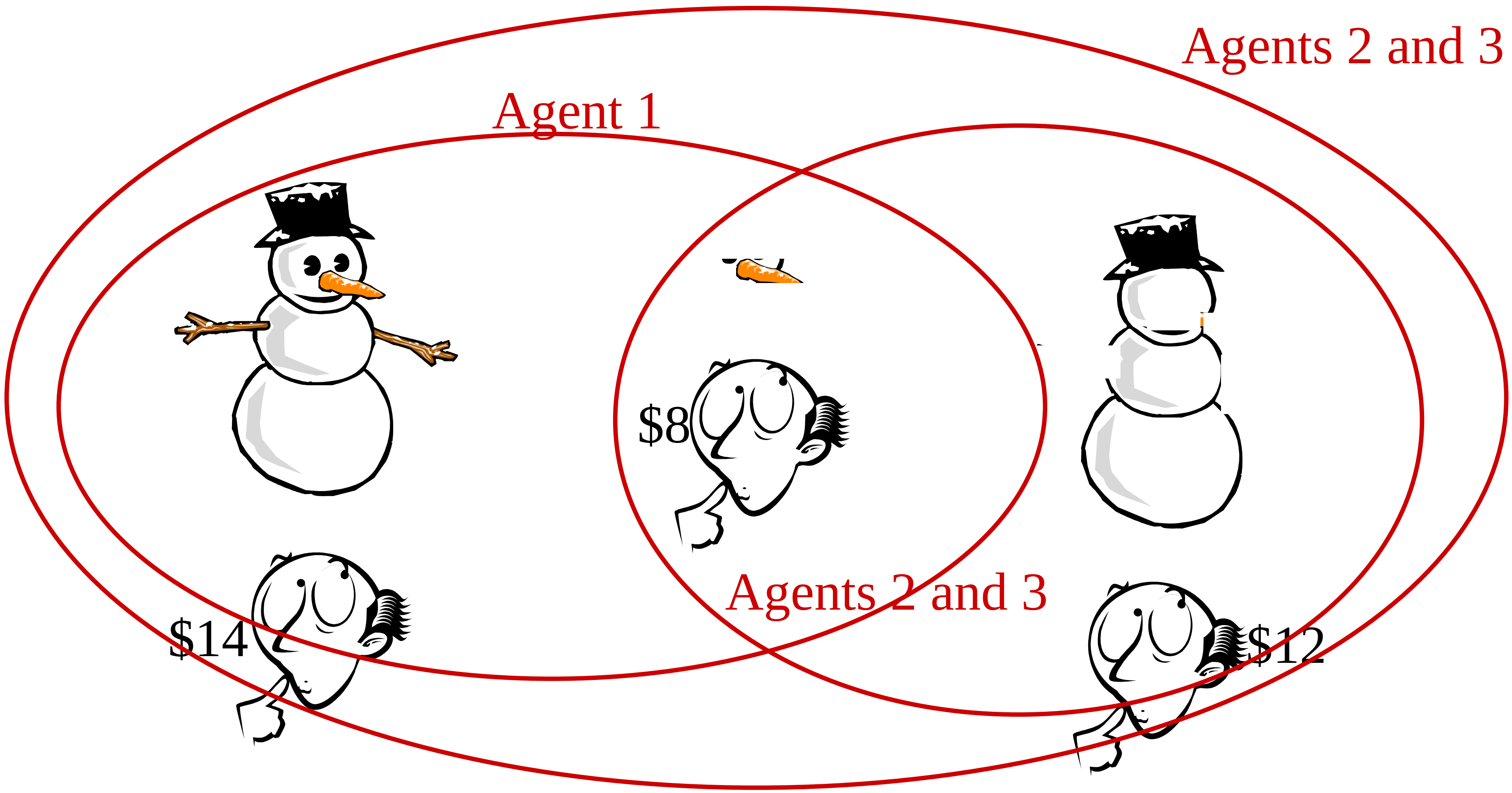
Partition principle:

Theorem. d_M is a faithful VCG implementation when:

(i) the distr. algorithm s^m is correct

(ii) the distr. algorithm s^m solves $Eff(\theta_{-i})$ whatever actions of agent i.

(iii) there are no CC issues, with values reported to center


Agents 2 and 3
Agent 1
$8
$14
Agents 2 and 3
$12

Proof (sketch)

Agent 1 wants the mechanism to choose the optimal solution to V(N).

Agent 1 cannot affect its price $\pi_1(a,\theta_{-1})$, for outcome a.

Agent 1 indifferent between solving $V(N\backslash 2)$ and not solving.

Example II: Quorums

- 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

Theorem. d_M is a faithful impl. of a **truthful** scf when

(i) the distributed algorithm s^m is correct

(ii) a quorum approach is used for computation

(iii) there are no CC issues, with values reported to center

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

- 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

- 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

- 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 04a)

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

Future Directions for Distr. Impl.

- Distributed implementations for Nash equilibrium algorithms (e.g. Kearns, Littman & Singh 01)
  - bring the *algorithm itself* into an equilibrium
- General purpose methods for both *AC & CC*, in systems without a center:
  - unification between methods from Byzantine Fault Tolerance and Distributed MD?
- Methods for Execution enforcement:
  - ensure that agents choose to follow-through with the outcome that is computed

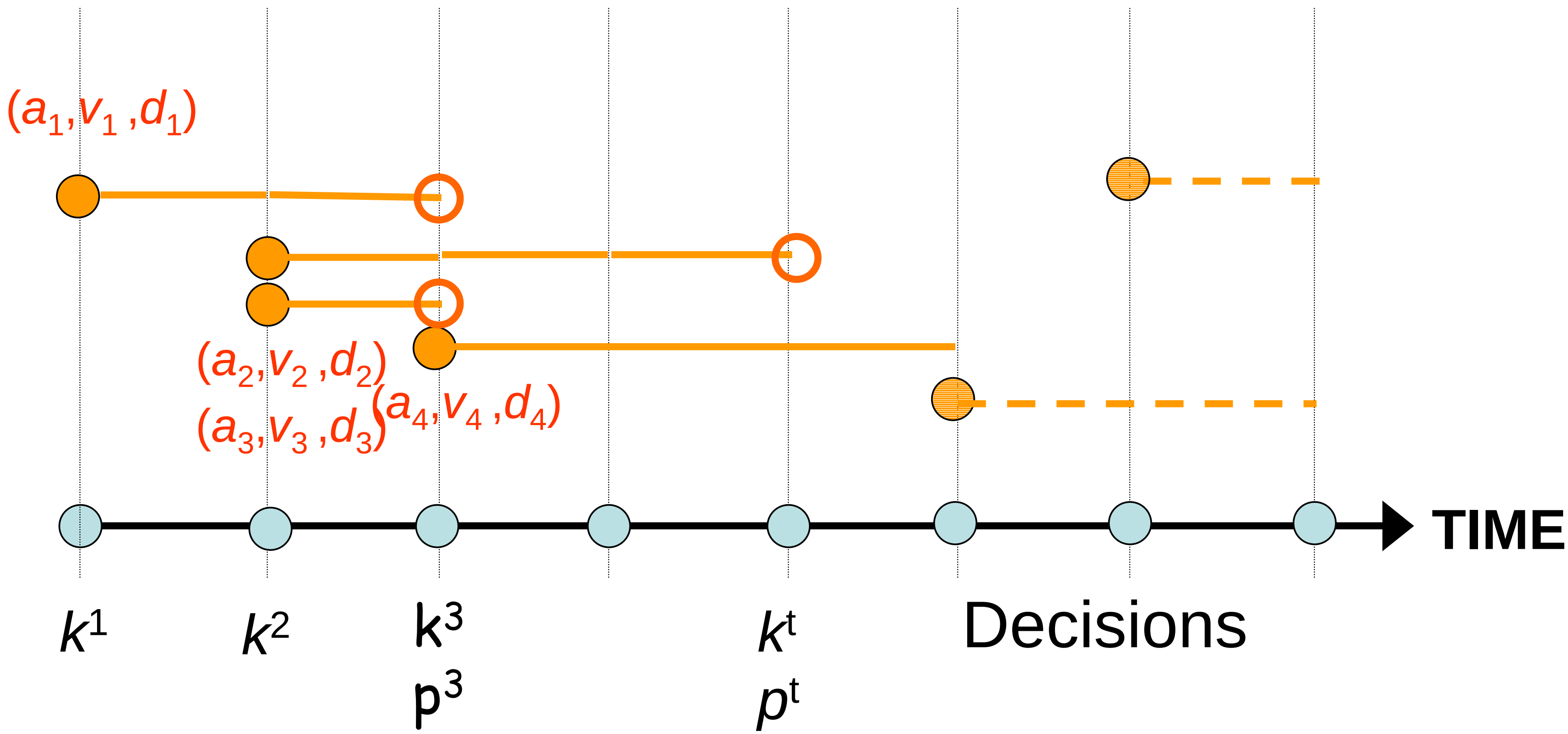
Outline

- Brief review of GT concepts
- Static & Centralized MD
  - Characterization of truthful mechanisms
  - VCG: a case study
- Static & Decentralized MD
  - Iterative mechanisms: comm & value complexity
  - Distributed implementations
- Dynamic & Centralized MD
  - online mechanism design
- Adaptive & Decentralized MD
  - "market of minds"

Example: WiFi @ Starbucks
(dynamic and centralized MD problem)

Online Mechanism Design

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



Online Mechanism Design

- 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; Hajiaghayi et al; Blum et al.)
  - reusable items (Hajiaghayi et al., Porter)
  - single-minded agents (Awerbuch et al.)
  - bounded-demand (Gonen et al.)
- Bayesian-Nash truthful:
  - more general sequential decision problem (Parkes & Singh 03, NIPS 04)
  - take an MDP approach
- Applications: Ad Auctions, WiFi at Starbucks, data storage in Grid computing, …

MDP Model of Online Mechanisms

NIPS'04: 10:50 am, Tuesday.

- Sequence of decisions $k_1,\ldots,k_{t-1}$
- Agent of type $\theta_i=(a_i,v_i,d_i)$ has reward

  $R^i(h_t,k_t) = v_i(k_{\leq t};\ \theta_i) - v_i(k_{\leq t-1};\ \theta_i)$ for $a_i \leq t \leq d_i$

  $= 0,$ otherwise
- State: history of agent types, decisions made

  $h_t=(\theta_1,\ldots,\theta_t;k_1,\ldots,k_{t-1})$
- System dynamics: $\mathrm{Prob}(h_{t+1} | h_t,k_t)$
- Payoff $R(h_t,k_t) = \sum_i R^i(h_t,k_t)$

- **Mechanism**: $\langle\pi, t\rangle$

  policy $\pi_t : H_t \to K_t$

  payments $t_{i,t}: H_t \to R$

achieve an ε-BNE

An Online VCG Mechanism

- Policy $\pi=\{\pi_1,\ldots,\pi_T\}$ where $\pi_t: H_t \rightarrow K_t$
- $V^\pi(h_t)=E_\pi\{R(h_t,\pi(h_t))+R(h_{t+1},\pi(h_{t+1}))+\ldots+R(h_T,\pi(h_T))\}$
- Optimal policy π^* maximizes value in all states
  - maximize current reward plus expected future reward
- $V^*(h \in H_t)$ denotes the optimal MDP value function.

Online VCG Mechanism:

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

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

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

total reported value — expected positive effect on value system

Key Property: Stalling

- Stalling: The space of policies allows the mechanism to delay the decision about the value to agent i.

- Formally, in state h_t in which agent i has arrived,

  $V^*(h_t) \geq R(h_t,k') + \sum_{h'} Prob(h'|h_t,k')V^*(h')$, where $k'=\pi^*(h_t^{-i})$

Main Result.

(Parkes & Singh '03)

Theorem. Online VCG mechanism with an optimal policy π^* for a correct MDP model that satisfies stalling is:

- BNE incentive-compatible
- No-deficit (w/ no positive externalities)
- Satisfies voluntary participation (w/ value-monotonicity)
- Implements an expected-value maximizing policy

NB. Bayes-Nash and not dominant-strategy equilibrium because optimality of MDP policy relies on correctness of model, which relies on equilibrium assumption…

Proof Sketch.

- VCG Payment: $R^{i}_{\leq T}(\theta_i;\pi^*)-[V^*(h_{ai})-V^*(h_{ai}^{-i})]$

- First two terms align agent's incentives with implementing π^* wrt its true type and reported types of other agents
- Agent wants the mechanism to have an accurate view of the state of the world.

Subtle: $V^*(h_{ai}^{-i})$ depends on reported arrival time.
-- but, effect cancels out in expectation with effect on $V^*(h_{ai})$...

Solving very large MDPs approximately

- 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: (Parkes & Singh, NIPS'04)

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

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

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

Directions for Online MD

- Learning in online mechanisms:
  - current work assumes that we have a model of the world
  - interesting to learn a model, but keep IC
- Strengthening to dominant-strategy implementation…
- Extensions to settings when online VCG fails budget-balance,
  - for instance when the presence of an agent makes the other agent's better off…
  - i.e. agents bring skills rather than consume resources…

Outline

- Brief review of GT concepts
- Static & Centralized MD
  - Characterization of truthful mechanisms
  - VCG: a case study
- Static & Decentralized MD
  - Iterative mechanisms: comm & value complexity
  - Distributed implementations
- Dynamic & Centralized MD
  - online mechanism design
- Adaptive & Decentralized MD
  - "market of minds"

Example: Hubble Telescope Scheduling

(adaptive and decentralized MD problem)

sequential decision problem

RL, coordination, self-interest

"Market of Minds"

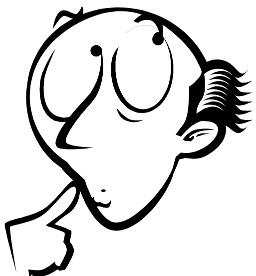
(Cavallo, Parkes, Singh)

- Sequential decision problem, *a la* online MD.
- New components of problem:
  - persistent agents
  - unknown reward functions
  - unknown model
- Agents receive **local rewards** for actions.

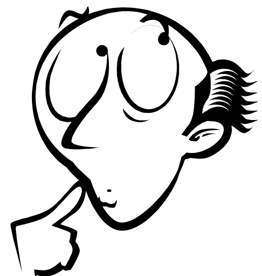
- **Goal**: learn and implement an optimal joint policy despite agent self-interest.

$\Rightarrow$ Design side-payments to "shape rewards", to that multi-agent learning will promote good overall performance.

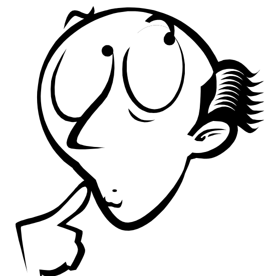
Eating and Drinking and Sleeping



eating makes
me most happy



drinking makes
me most happy



sleeping makes
me most happy

receives reports about the
value to each "subagent" for each action

- “Researchers at four universities found two areas of the brain that appear to compete for control over behavior...”

  -Harvard Gazette, describing [McClure et al, 2004]

MoM: Related work

- Q-decomposition [Russell & Zimdars, 2003]
  - agents receive local rewards
  - distributed SARSA learning
  - central arbitrator takes actions
  - no self-interest
- Action selection methods [Humphrys, 1997]
  - reinforcement learning for action-selection between self-interested subminds
  - no equilibrium analysis
- Intelligence as an "economy of agents" [Baum, 1999]
  - evolutionary computation
  - markets-based method to do credit assignment
  - evolve a system of problem solvers

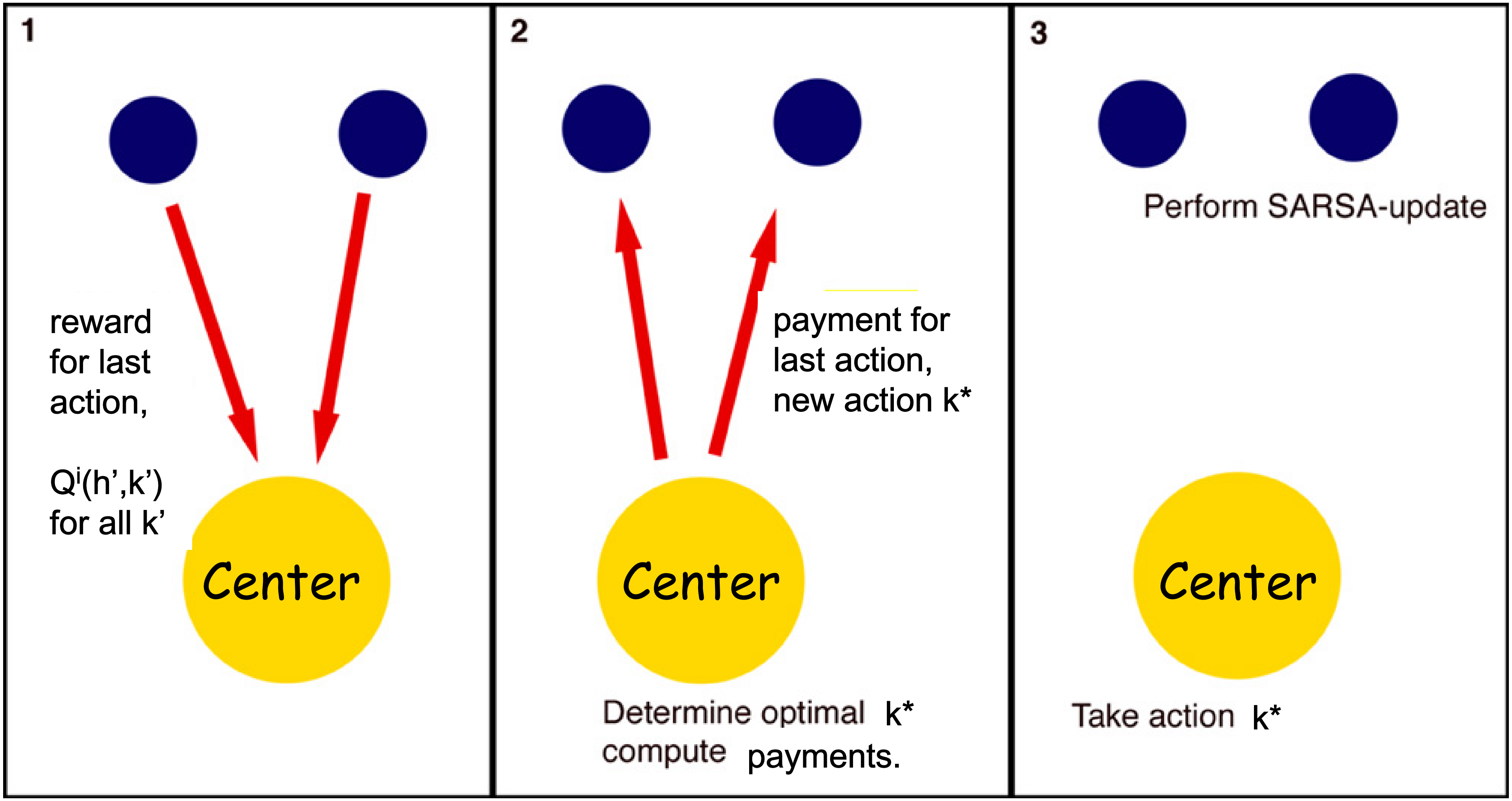
Review: Q-learning

- state $h_t \in H_t$
- action $k_t \in K_t$
- system dynamics: $Prob(h_{t+1} | k_t, h_t)$
- agent reward: $R^i(h_t, k_t)$
- system reward: $R(h_t, k_t) = \sum_i R^i(h_t, k_t)$
- policy $\pi = \{\pi_1, \ldots, \pi_t\}$; $\pi_t : H_t \rightarrow K_t$
- discount factor γ

- Optimal π^*: maximizes discounted system reward in all states
- $V^*(h_t)$: discounted system reward from h_t, given π^*
- $Q^*(h_t, k_t)$: discounted system reward from action k_t in state h_t, followed by optimal policy.

Basic Set-up for MoM

- Each agent learns its individual Q-value for each action in each state
  - $\hat{Q}^i(h_t,k_t)$ for action k_t, conditioned on system-wide policy
  - where reward sequence is intrinsic + payments
- Center elicits value information from each agent in each period.
- Center chooses $k^*_t \in \arg\max_k \sum_i \hat{Q}^i(h_t,k)$,
- Agents observe individual reward, world state, action taken, and receive payment $p_i(h_t,k^*_t)$ from center.
- Agents update value estimates.



Center plays $\max_{k'}\sum_i Q^i(h',k')$ with prob $1-\varepsilon$,
uniform-random exploration with prob ε.

Agents: SARSA update $Q^\pi(h^t,k^t)=(1-\alpha)\cdot Q^\pi(h^t,k^t)+\alpha\cdot(\text{reward}+\gamma\cdot Q^\pi(h^{t+1},k^{t+1}))$

On-policy learning.

Off-policy learning fails because of "illusion of control" (Russell & Zimdars)

Shaping Rewards to Provide Truthfulness

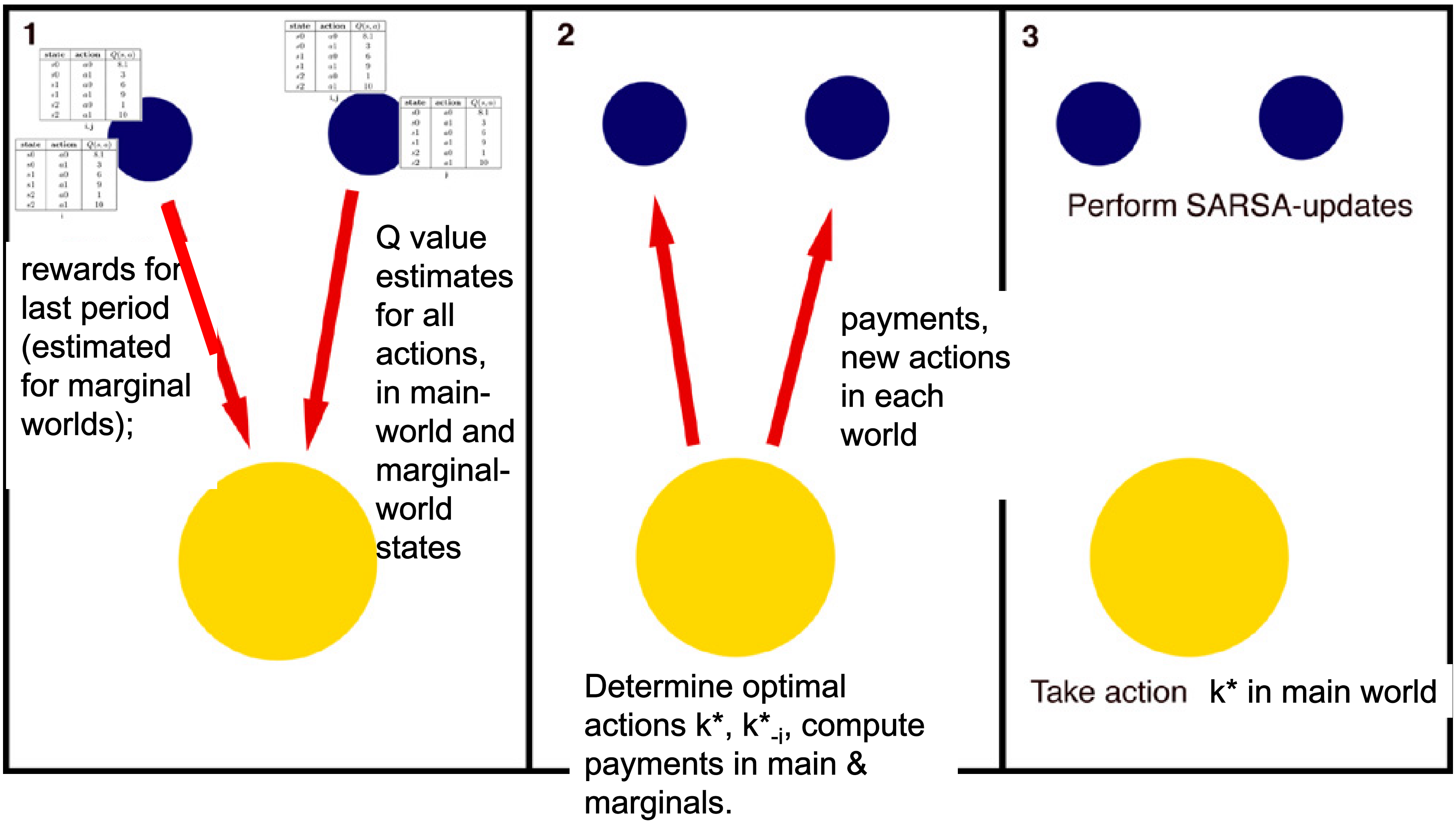
Marginal Worlds VCG:

- At each time-step t, pay agent i sum of agent, minus sum of rewards agents j≠i would have received without i

$$p_i(h_t,k_t) = \underbrace{\sum_{j\neq i} R^j(h_t,k_t)}_{\text{aligns incentives with optimal joint policy}} - \underbrace{\sum_{j\neq i} R^j(h^{-i}_t,k^{-i}_t)}_{\text{must track "marginal worlds" (know what action would be taken)}}$$

- **Center** must simulate dynamics in marginal worlds to compute second term, and must learn an transition model.
- **Agents** must compute Q*-values for all worlds:
  - Perform SARSA update for marginal world when $(h_t,k_t,h_{t+1},k_{t+1})$ would be selected in optimal marginal-world policy
  - Center can also do "explicit exploration" for marginal worlds

MoM: Marginal-Worlds VCG



Properties

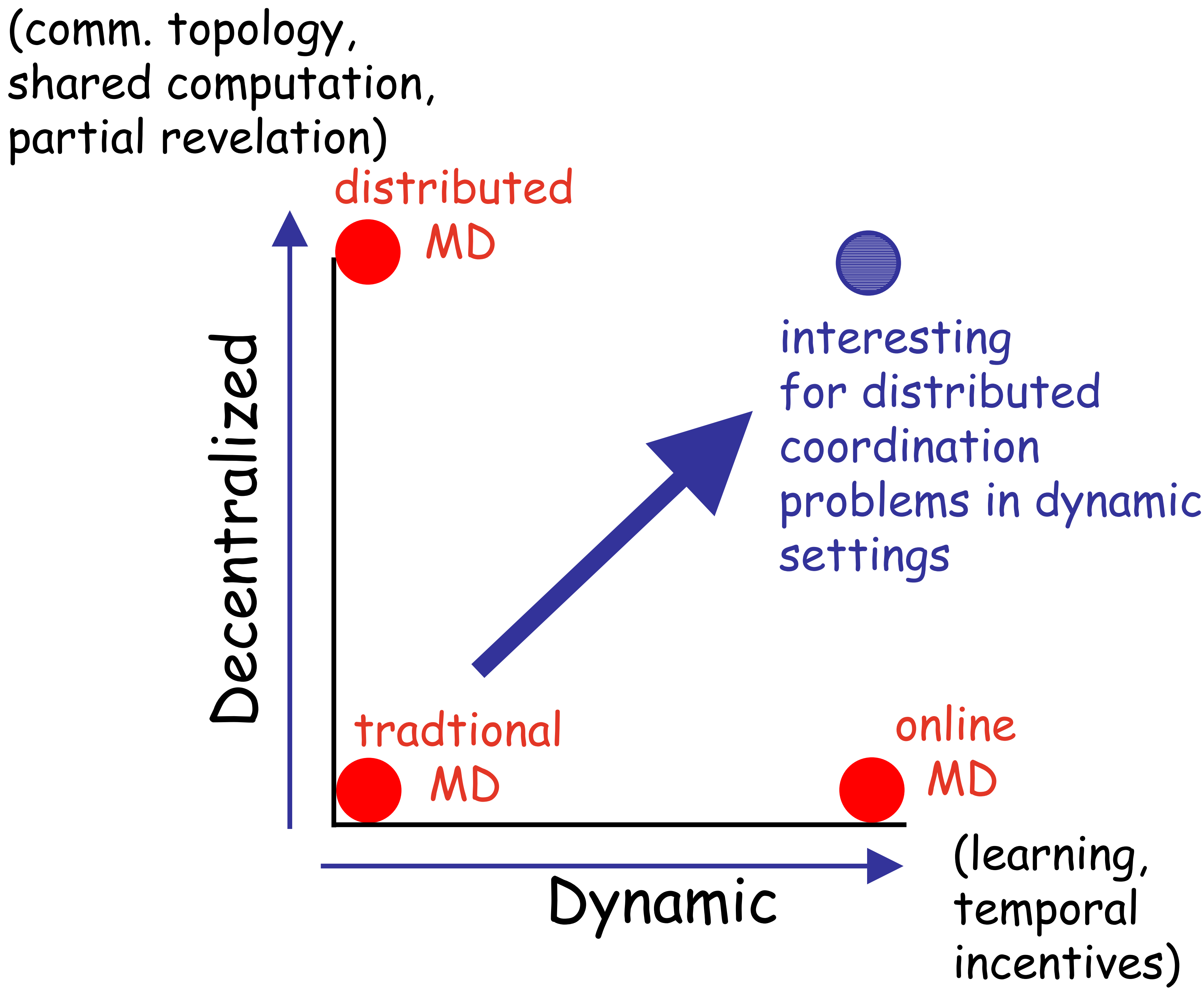
Optimal once converged:

- **Theorem**: When all agents know the true Q*-values:
  - honest reporting of both $Q^{*i}(h,k)$ and $R^i(h,k)$, $\forall h \in H$, $k \in K$ is an ex post (& perfect) equilibrium
  - each agent i is guaranteed to receive non-negative payoff
  - (weak) budget-balanced: net payments to center

Learning Phase

- Faithful SARSA updates not in equilibrium in learning phase. Try to speed-up convergence of system to the optimal policy.
- Can achieve "algorithm compatibility" by using the center to do checking:
  - replicate SARSA updates for one of the agents
  - punish if $Q^*(h,k)$ reports show deviation from SARSA (e.g. by removing the agent from consideration in center's decision for a while).

- *But, no general characterization of a method to bring truthful reports of $R^i(h,k)$ into equilibrium.*


(comm. topology,
shared computation,
partial revelation)
distributed
MD
Decentralized
interesting
for distributed
coordination
problems in dynamic
settings
tradtional
MD
online
MD
Dynamic
(learning,
temporal
incentives)

Summary

- Mechanism design provides methods to design worlds so that the "right outcomes" are implemented despite private-information and agent self-interest
- Large and rich literature in economics, but mostly focused on "direct-revelation" with centralized computation
- Agenda in CMD is to systematically relax assumptions:
  - Static & Centralized
  - Static & Decentralized
  - Dynamic & Centralized
  - Adaptive & Decentralized
- Get good economic and good comput./info. properties
- Lots to do!

Thankyou!

parkes@eecs.harvard.edu

http://eecs.harvard.edu/econcs

http://eecs.harvard.edu/~parkes