Economics in the Design of Computer-Assisted Markets

Jeffrey K. MacKie-Mason

University of Michigan

Dept. of Economics and
School of Information

www-personal.umich.edu/~jmm/

Conference on Trends and Research in Electronic Commerce

Rüschlikon, Switzerland, November 2-3, 1998

Program for Research on the Information Economy

- **http://prie.si.umich.edu**
- About 20 faculty and grad students:
  - comp sci, econ, artificial intelligence, management, organization theory, cog psych
- Key projects:
  - **AuctionBot**: on-line configurable auction server (auction.eecs.umich.edu)
  - **MARX**:auction markets for real-time management of network resources (marx.ai.eecs.umich.edu/)
  - **PEAK**: bundling theory and field trial (with Elsevier) for e-commerce in information goods (www.lib.umich.edu/libhome/peak/)

Market design challenge:

- Designing interplay of **controlled** and **uncontrollable** agents to accomplish resource allocation goals
  - computer-assisted business-to-business negotiation
  - dynamic interaction between buyers and sellers of information goods
- Mechanism Space = Goods Space X **Rules Space**

Not all markets are created equal

- As a general matter,
  max revenue (profit)
  ≠ max efficiency
  ≠ max # trades
- Market performance depends on:
  - goods definitions
  - rule definitions
    - min bid
    - bid increment
    - single or multiple units
    - stopping rules
    - &c.

Practical Market Design

- Example: "free-rider" auction has **different outcomes** depending on when bidding closes
- FCC Spectrum Auctions (US)
  - full combinatorial auctions theoretically efficient
  - but exponentially complex
  - used: multiple simultaneous ascending bid
    - high revenue, but collusion discovered post-auction due to bidding rules (insufficient info hiding)

What's so important about auctions?

- Generic problem: Engineering **new transaction mechanisms** that exploit cheap silicon and sand
- **Auctions are the atomic primitive** of markets
- Auctions provide a formal language and design architecture for essentially all negotiation instances

Motivation for market mechanisms

- **First Welfare Theorem**: A competitive equilibrium is Pareto efficient: i.e., competition forces efficiency

- **Second Welfare Theorem**: Any Pareto efficient allocation can be supported by a competitive equilibrium

Harnessing market mechanisms

- Construct mechanisms that converge to competitive equilibrium
- (Same approach applies to, say, revenue or profit-maximizing design goal, rather than efficiency-maximizing)

Not so fast...

- Market mechanisms provide **poor solutions** for distressingly fundamental problems
- Many fundamental problems have distressingly high computational **complexity**

Poor solutions

- Market mechanisms often **aren't efficient**
  - externalities
  - transaction costs
  - increasing returns
  - integer constraints
  - asymmetric information
- **Negative results** for more general mechanisms

Defining good solutions

- What is the **goal**?
  - efficiency, or …
  - revenue maximization, …
- What are the **constraints**?
  - information
  - voluntary participation
  - assumed agent strategies are plausible
  - low computation and complexity costs
- **Solve** for set of mechanisms that achieve goal subject to constraints

Negative: Dominant agent strategies

- Gibbard-Satterthwaite (1973)
  - If agent preferences are ordinal but otherwise unrestricted, and
  - If there are at least three outcomes that are each sometimes socially desirable
  - **Then only dictatorial** mechanisms can implement Pareto efficiency in dominant strategies

Negative: Bayesian-Nash Strategies

- Defn: Agents play strategies that provide best expected payoff if other agents also play Bayesian-Nash strategies
- Myerson-Satterthwaite (1980):
  - If agent valuations drawn from overlapping intervals
  - And trade is bilateral
  - **Then no** mechanism can always be
    - efficient
    - non-coercive
    - no-deficit

Complexity

- Market mechanisms may be **intractably complex** for problem solving
- For many problems comp(ε,d) = o($(1/\varepsilon)^d$)
  - utility and expected utility maximization
  - finding competitive, Nash, and rational expectations equilibria
  - option valuation
    - (Rust 1997, and Traub et al. 1988)

What can we do?

- Markets work "well" in real world
- Strong theoretical properties in special cases
- But in practice hard to achieve ideal
- Should we just design ad hoc markets, and hope they are good enough?
- What is **generalizable**? What is a **principled basis** for design and extension?

Pragmatic mechanism design

- (1) What **relaxed set of constraints** admits optimal solutions?
  - E.g.: efficient bilateral trade with ex post voluntary participation is possible with subsidy (GVA: MacKie-Mason and Varian 1994)
- (2) Alternatively, how much efficiency can **admissible mechanisms** achieve?

Application: Scheduling Problems

- Walsh, Wurman, Wellman, MacKie-Mason (1998)
- Autonomous agents with:
  - conflicting scheduling requirements
  - private local information
  - messaging capability
- Scheduling optimization is hard:
  - discrete
  - complementarities
  - even with public information it's typically a knapsack problem

The tasks for pragmatic design

- Protocol design:
  - how to organize message passing, reach closure, determine outcome?
- Economic design:
  - how to elicit messages containing info needed for good outcomes?

Schedule example

A1
v=$10
L=2 hrs
by 12:00

A2
v=$16
L=2 hrs
by 11:00



A4
v=$6
L=1 hrs
by 11:00

A3
v=$14.5
L=4 hrs
by 16:00

Min price = $3

Markets can be good

- **Theorem 1**: If there exists a **p** such that *f* is a market equilibrium, then *f* is **Pareto optimal**
- But mkt equilib **may not exist**

	Length	Deadline	Value
Agent 1	2	2	3
Agent 2	1	2	2

What do we know in general?

- **Lemma**: The **single unit** problem always has a unique minimum equilibrium price vector
- **Theorem 3**: **Any** optimal solution to a single unit scheduling problem is supported by a price equilibrium
- But now: How to **find** good solutions even for single unit problems?

1. Ascending Single-Good Auction

- **Goods**: 1 auction for each slot
- **Rules**:
  - min. bid increment ε
  - no bid withdrawal
  - closure when bidding stops
- **Strategies**:
  - agent j bids for set of slots to max surplus at current p
  - or drop out if no set of slots has positive surplus
  - N.B. For single-slot problem, this is dominant strategy
  - N.B. For multi-slot, not regret proof

ASGA: Single-slot problems

- Thm 3 says **equilib exists**, but this **protocol not guaranteed to find it**

	1pm	2pm	3pm
A_1	6	6	
A_2	7	7	7
Min	4	3	3

A_2 first: could bid $b_2=3$
A_1 then could bid $b_1=4$: $s_1 \rightarrow A_1$, $s_2 \rightarrow A_2$, v=16
But at $p_2=3$ A_1 prefers s_2 so not an equilib

ASGA single-slot (cont)

- **Theorem 5**: p_j will differ from the min. unique equilib price by at most $\kappa\varepsilon$, where κ = min(#goods,#agents)
- **Theorem 6**: *v(f)* will differ from optimal by at most $\kappa\varepsilon(1+\kappa)$

Single-slot demand summary

- Equilibrium exists
- ASGA **deviates from optimal by bounded amount**, proportional to ε

AGSA: Multi-Slot Demands

- **Lemma:** p can differ from equilib by arbitrarily large amount
- **Lemma:** *v(f)* can differ from optimal by arbitrarily large amount

	Length	Deadline	Value
Agent 1	1	1	3
Agent 2	2	2	11

Min prices: (1,9)
Suppose A_2 bids: $b_1=1$, $b_2=9$
Suppose A_1 bids: $b_1=2$
Then: $s_1 \rightarrow A_1$, $s_2 \rightarrow A_2$, v(f)=3
But optimum: $s_1 \rightarrow A_1$, $s_2 \rightarrow \perp$, v(f)=12

Example cont.

- Can make inefficiency **arbitrarily large**
- Is it reasonable for agent 2 to stop bidding?
  - By bidding $b_1=3$, it can do better than if auction stopped
  - However, better still is someone else bids away s_2
  - **No dominant strategy exists** that makes A_2 always better off

2. Combinatorial Auctions

- Looking for price equilibria for multi-slot problems
- Redefine concerns: combinations of goods
  - Banks et al. RAND 1989
  - Rassenti et al. Bell 1982
  - Rothkopf et al. Mgmt Sci 1998
- Problem: **Exponential complexity** (2^n combinations)
  - Need feasible mechanisms that work "pretty well"

Combinatorial Auctions (cont.)

- New goods definition: pair(y,z) is y slots by deadline z, contains z
- Complexity = O(nL), L = max schedule length
- Finds equilibria for some problems not solved by ascending auction:

	Length	Deadline	Value
Agent 1	2	2	3
Agent 2	1	2	2

Consider: p(1,1)=p(1,2)=2.1, p(2,2)=2.9
A_1 gets (2,2), A_2 gets nothing
A_1 can't do better with single goods
A_2 doesn't want anything at these prices

Combinatorial Auctions Performance

- CA price equilibria are **not necessarily efficient**

	Length	Deadline	Value
Agent 1	2	2	3
Agent 2	1	2	2
Agent 3	1	2	2

Equilib: p(1,1)=p(1,2)=1.6, p(2,2)≥3.2
Another: p(1,1)=p(2,1)=2.1, p(2,2)=2.9
(same as previous example)
Second is **suboptimal** -- no useful bound on deviation from optimality!

Combinatorial Auction: Summary

- All single-good price equilibria are also (not nec. unique) combinatorial equilibria
- Some problems with no single-good price equilibria **do** have combinatorial equilibria
- But, combinatorial equilibria (that are not single-good equilibria are **not necessarily efficient**)

Generalized Vickrey Auction

- Varian and MacKie-Mason (1994)
- **Goods** definition: Identical to agent concerns
- **Rules**: Agents announce $r_j()$, alleged utility function
  - GVA computes optimal allocation for the r()
  - GVA computes payments to agents $P_j(r_{-j})$

GVA properties

- Pareto efficient
- If a price equilibrium exists, **GVA payments are no higher** than payments under minimum equilibrium prices (Gul & Stacchetti)
- If minimum reserve prices known, GVA for scheduling is participatory efficient in dominant strategies
  - If not, participatory efficient for M-1 agents (**market maker may run deficit**)

GVA computation cost

- **Single-slot** optimal solutions can be found by a simple centralized greedy algorithm (Neapolitan & Naimipour) in **Θ(m log m)**
- **GVA needs to solve m+1** of these: one for global optimum, plus one for each of agent payments, so $\Theta(m^2 \log m)$
- Same m-factor increase **for multi-slot**, but **the underlying optimization is now NP-complete** Integer Knapsack

SUMMARY

- We have explored a **spectrum of mechanisms** for scheduling problems:
- single good $\leftrightarrow$ combntrl $\leftrightarrow$ GVA
- **increasing** ---------->
  - **scope of concerns**
  - **cases in which (near) efficient outcome found**
  - **computational complexity**
- Proposed: "metamechanisms" for endogeneously selecting mechansisms from plausible set

Applying the theory

- Multi-stage network scheduling with multiple QoS
- Web caching
- Task networks
- Device or facility scheduling

Exploring the design space

- Design space is huge
- The set of existing instantiations -- real and artificial -- has measure zero
- Need to **systematically map** more of the design space
- Economics provides **principled framework** for guiding and summarizing the mapping