Special Topics at the Intersection of Social and Information Sciences, Fall 2011 Lecture 1

Katrina Ligett

Caltech

September 27

How should we sell my old cell phone?

What goals might we have?

How should we sell my old cell phone?

What goals might we have?

- maximize revenue
- get it to the person who has the most use for it

How should we (re)design network routing protocols?

What goals might we have?

How should we (re)design network routing protocols?

What goals might we have?

- minimize overall latencies
- distribute traffic "fairly" among ISPs

Economics and Computer Science

- selling cell phone = economics, traditional auction theory
- routing protocols = computer science, traditional networks

How should we sell my old cell phone?

Mature theory of auctions; computational perspective provides

- new applications
  - new scale
  - new challenges
- new tools/focus
  - complexity bounds
  - worst-case approximation

How should we (re)design network routing protocols?

(Less mature) science of networks; economic perspective provides

- focus on incentives
  - designers
  - participants
- focus on outcomes and metrics
  - equilibrium notions
  - concepts of welfare

This course

Will sample recent topics at active intersection between game theory and computer science, “algorithmic game theory”.

Research-focused.

Today

- find out about your background
- establish a common language
- preview some topics for the semester
- course mechanics

Introductions

For now:

- Name
- Background
- Experience
- Interests

Survey at end of class

Sources for today's lecture

- Nisan, Roughgarden, Tardos, and Vazirani (eds), *Algorithmic Game Theory*, Cambridge University, 2007. [AGT book]
- Tim Roughgarden's lecture notes from his "Topics in Algorithmic Game Theory" course

Outline

1. Algorithmic Mechanism Design
2. Defining and Quantifying the Effects of Selfishness
3. Algorithmic? Game Theory?
4. This term

Algorithmic Mechanism Design

Want to perform computation on data held by self-interested participants

Example: auction

- private information: how much each person values the cell phone
- want "mechanism": protocol that interacts with participants, determines outcome (winner and how much she pays)

First-price auction

Winner is highest bidder; selling price is her bid.

First-price auction

Winner is highest bidder; selling price is her bid.

- Bidders shade bids
- Difficult to know how to bid
- Difficult to predict outcome

Ascending auction

(Like an art auction)

Second-price auction

Winner is highest bidder; selling price is *second-highest* bid.

Second-price auction

Winner is highest bidder; selling price is *second-highest* bid.

- Famous result [Vickrey, 61]: every participant in a second-price auction may as well bid truthfully.

Auctions by eBay, Amazon, Google, Microsoft: similar to second-price.

Second-price auction

Winner is highest bidder; selling price is *second-highest* bid.

- Famous result [Vickrey, 61]: every participant in a second-price auction may as well bid truthfully.
- Easy to know how to bid

Auctions by eBay, Amazon, Google, Microsoft: similar to second-price.

Second-price auction

Winner is highest bidder; selling price is *second-highest* bid.

- Famous result [Vickrey, 61]: every participant in a second-price auction may as well bid truthfully.
- Easy to know how to bid
- Easy to predict outcome, given valuations

Auctions by eBay, Amazon, Google, Microsoft: similar to second-price.

Algorithmic Mechanism Design asks. . .

- How much does it matter that our mechanism doesn't have direct access to the participants' data?
- What is the impact of computational constraints?

Second-price (Vickrey) auction

- Each bidder i has private valuation v_i ("willingness to pay"); submits some bid b_i to auctioneer.
- Auction has two parts:
  - Allocation scheme: who gets what? (here, highest bidder gets item)
  - Payment scheme: who pays what to whom? (here, highest bidder pays second-highest bid to the auctioneer and nobody else pays anything)
- We'll assume if i loses, has utility 0; if i wins and pays p, has utility $v_i - p$ (quasilinear utility)

The Vickrey auction is *truthful/strategyproof*

Theorem

For every player i and every set $\{b_j\}_{j \neq i}$ of bids for the other players, player i maximizes her utility by choosing $b_i = v_i$.

That is, bidding truthfully is a *dominant strategy*, even if you know everyone else's bids!

The Vickrey auction is *truthful/strategyproof*

Proof.

Fix v_i, $b_j, \forall j \neq i$. TS: $b_i = v_i$ maximizes utility. Let $B = \max_{j \neq i} b_j$. Three cases:

- Case 1: $v_i < B$.
- Case 2: $v_i > B$.
- Case 3: $v_i = B$.

□

The Vickrey auction is *truthful/strategyproof*

Proof.

Fix v_i, $b_j, \forall j \neq i$. TS: $b_i = v_i$ maximizes utility. Let $B = \max_{j \neq i} b_j$. Three cases:

- Case 1: $v_i < B$.
  - Bidding truthfully or less than or equal to B (no matter how ties are broken): will get zero utility.
  - If bids more than B, will win and pay B, getting negative utility.
- Case 2: $v_i > B$.
- Case 3: $v_i = B$.

□

The Vickrey auction is *truthful/strategyproof*

Proof.

Fix v_i, $b_j, \forall j \neq i$. TS: $b_i = v_i$ maximizes utility. Let $B = \max_{j \neq i} b_j$. Three cases:

- Case 1: $v_i < B$.
- Case 2: $v_i > B$.
    - Bidding truthfully: i wins and gets $v_i - B > 0$ utility.
    - Bidding at or below B: loses and gets zero utility.
    - Bidding above B: price, allocation, and utility are always the same.
- Case 3: $v_i = B$.

□

The Vickrey auction is *truthful/strategyproof*

Proof.

Fix v_i, $b_j, \forall j \neq i$. TS: $b_i = v_i$ maximizes utility. Let $B = \max_{j \neq i} b_j$. Three cases:

- Case 1: $v_i < B$.
- Case 2: $v_i > B$.
- Case 3: $v_i = B$.
  - Bidding truthfully: may win or lose; get 0 utility.
  - Bidding above B: win and get negative utility.
  - Bidding below B: lose and get 0 utility.

□

The Vickrey auction is *truthful/strategyproof*

Proof.

Fix v_i, $b_j, \forall j \neq i$. TS: $b_i = v_i$ maximizes utility. Let $B = \max_{j \neq i} b_j$. Three cases:

- Case 1: $v_i < B$.
- Case 2: $v_i > B$.
- Case 3: $v_i = B$.

No false bid yields strictly higher utility! □

Vickrey: Regrets

Theorem

For every bid $b_i \neq v_i$, there exists a set of bids $\{b_j\}_{j\neq i}$ for the other players such that i's utility is strictly lower than it would have been at $b_i = v_i$.

Vickrey: Regrets

Theorem

For every bid $b_i \neq v_i$, there exists a set of bids $\{b_j\}_{j \neq i}$ for the other players such that i's utility is strictly lower than it would have been at $b_i = v_i$.

Proof.

- Case 1: $b_i < v_i$. Set B such that $b_i < B < v_i$, so i now loses.
- Case 2: $b_i > v_i$. Set B such that $b_i > B > v_i$, so i now wins and overpays.

□

This property makes the auction *strongly truthful*.

Additional properties of Vickrey auctions

Theorem

Truthtellers always get nonnegative utility.

So, we say the Vickrey auction is *individually rational*—players are willing to participate.

Why do we care about truthfulness?

- Bidders don't need to do market research
- Bidders find it easy to compute their bids
- Outcomes are predictable
- Auctioneer can try to solve optimization problem on the true valuations

What might the auctioneer try to optimize?

- *Social surplus*: $\max \sum_{i=1}^{n} v_i x_i$, where x_i is a binary indicator of the winner.
- Auctioneer revenue
- . . .

More properties of the Vickrey auction

Theorem

If all players bid truthfully, the outcome maximizes social surplus.

Thus, the Vickrey auction is *efficient* in the economic sense.

Theorem

The Vickrey auction works with general valuations.

Theorem

The Vickrey auction is polynomial (linear)-time.

Thus, is is *efficient* in the computational sense.

In more complex auction settings, the interplay between these three constraints and the incentive constraints is a main focus of study.

Outline

1. Algorithmic Mechanism Design

2. Defining and Quantifying the Effects of Selfishness

3. Algorithmic? Game Theory?

4. This term

Quantifying the Effects of Selfishness

In the auction setting we just discussed,

- the truthfulness property meant that we didn't have to consider the impact of player selfishness, and
- the best action for each player doesn't depend on others' actions.

Quantifying the Effects of Selfishness

In the auction setting we just discussed,

- the truthfulness property meant that we didn't have to consider the impact of player selfishness, and
- the best action for each player doesn't depend on others' actions.

In game settings where that's not true, we need to come up with a model for how players will act and of the resulting outcomes, along with a metric for outcome quality.

Selfish Routing

What route should a commuter take to campus tomorrow?

- Selfishly want to minimize travel time.
- Probably don't consider impact on other commuters.

What is effect of everyone acting selfishly?

Pigou's Example (1920)

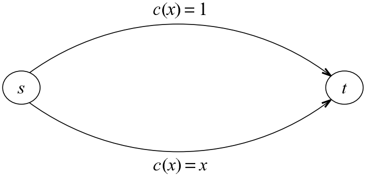
Players wish to drive from s to t
Two routes:

- Long but wide
- Short but narrow

Pigou's Example (1920)

Players wish to drive from s to t
Two routes:

- Long but wide
- Short but narrow



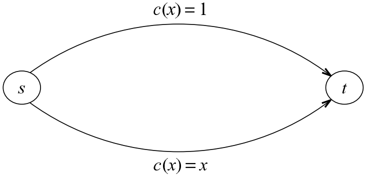
Continuous, non-decreasing *cost functions* $c(\cdot)$ represent latency as a function of the fraction of traffic using that edge (*congestion-dependent*).

Graphics in this section taken from Tim Roughgarden's book *Selfish Routing*.

Pigou's Example (1920)

Players wish to drive from s to t
Two routes:

- Long but wide: Time always 1 hour
- Short but narrow



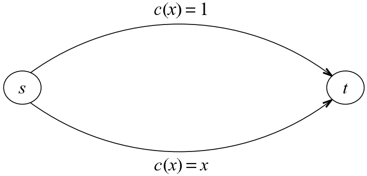
Continuous, non-decreasing *cost functions* $c(\cdot)$ represent latency as a function of the fraction of traffic using that edge (*congestion-dependent*).

Graphics in this section taken from Tim Roughgarden's book *Selfish Routing*.

Pigou's Example (1920)

Players wish to drive from s to t
Two routes:

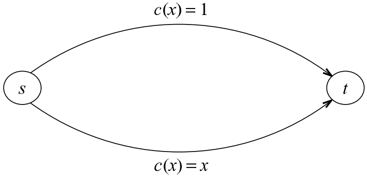
- Long but wide: Time always 1 hour
- Short but narrow: Time equals fraction of traffic



Continuous, non-decreasing *cost functions* $c(\cdot)$ represent latency as a function of the fraction of traffic using that edge (*congestion-dependent*).

Graphics in this section taken from Tim Roughgarden's book *Selfish Routing*.

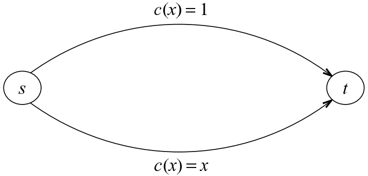
Pigou's Example



- If everyone wants to minimize their commute time, what will happen?

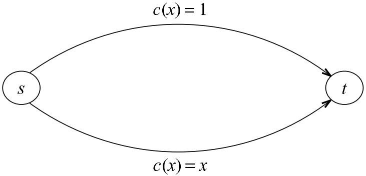
- Could we do better if we had centralized control?

Pigou's Example



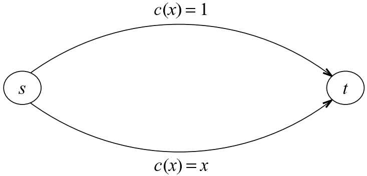
- If everyone wants to minimize their commute time, what will happen? **We expect all take bottom edge and thus take 1 hour.**
- Could we do better if we had centralized control?

Pigou's Example



- If everyone wants to minimize their commute time, what will happen? **We expect all take bottom edge and thus take 1 hour.**
- Could we do better if we had centralized control? **Yes! Force half and half, to get average drive time of 45 minutes.**

Pigou's Example

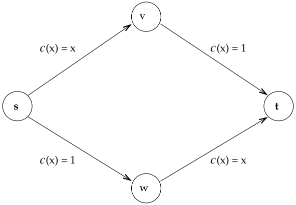


- If everyone wants to minimize their commute time, what will happen? **We expect all take bottom edge and thus take 1 hour.**
- Could we do better if we had centralized control? **Yes! Force half and half, to get average drive time of 45 minutes.**

How bad can the impact of selfish behavior be?

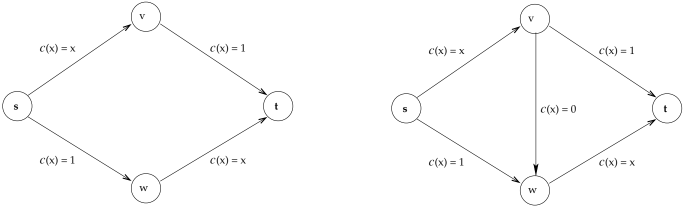
Braess' Paradox (1968)

New commuting example: Two routes, each with one short/narrow section and one long/wide section.
Traffic splits 50/50 and everyone has a 90-minute commute.



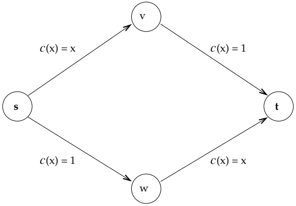
Braess' Paradox (1968)

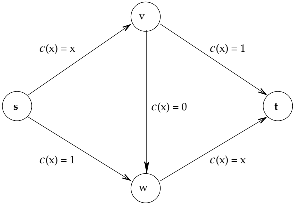
New commuting example: Two routes, each with one short/narrow section and one long/wide section.
Traffic splits 50/50 and everyone has a 90-minute commute.



What if we build a very efficient road to try to help?

Observations from Braess' Paradox





- Again, selfishness can lead to sub-optimal outcomes
- *Everyone* is hurt by the selfishness, in this case
- Counterintuitive outcomes

Outline

1. Algorithmic Mechanism Design
2. Defining and Quantifying the Effects of Selfishness
3. Algorithmic? Game Theory?
4. This term

Algorithmic? Game Theory?

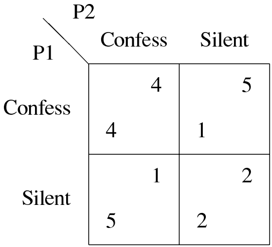
In routing examples, implicitly assumed that players would find a stable outcome, or *equilibrium*. Also assumed that they have the information and resources to *compute* such an outcome.

Game Theory 101

Game theory gives formal models for situations where multiple agents take actions that may affect one another's outcomes.

Game Theory 101

Game theory gives formal models for situations where multiple agents take actions that may affect one another's outcomes.

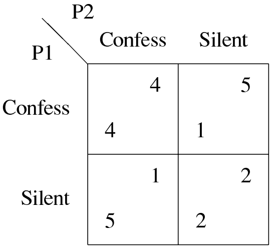


	P2	
P1	Confess	Silent
Confess	4 4	5 1
Silent	1 5	2 2

Figure: *Game matrix*, or *cost matrix* for the Prisoners' Dilemma game. (Figures in this section taken from *AGT*.)

Game Theory 101

Game theory gives formal models for situations where multiple agents take actions that may affect one another's outcomes.



P1 \ P2	Confess	Silent
Confess	4 4	5 1
Silent	1 5	2 2

Figure: *Game matrix*, or *cost matrix* for the Prisoners' Dilemma game. (Figures in this section taken from *AGT*.)

Only stable solution: both confess.

Matching Pennies

In Prisoners' Dilemma, there's an outcome such that no player would wish to unilaterally deviate from it. Not always the case!

Matching Pennies

In Prisoners' Dilemma, there's an outcome such that no player would wish to unilaterally deviate from it. Not always the case!

1 \ 2	H	T
H	-1 1	1 -1
T	1 -1	-1 1

Figure: Matching Pennies

Simultaneous-move games

- Each *player (agent)* i simultaneously picks a *strategy*, each from her own set S_i of possible *strategies*.
- The outcome for each player is fully determined by the vector $S = \times_i S_i$ of strategies.
- Each player has a preference ordering or *utility function* over outcomes.

Simultaneous-move games

- Each *player (agent)* i simultaneously picks a *strategy*, each from her own set S_i of possible *strategies*.
- The outcome for each player is fully determined by the vector $S = \times_i S_i$ of strategies.
- Each player has a preference ordering or *utility function* over outcomes.

Generally, we won't rely on explicit representations (like matrix)

Solution concepts

In Prisoners' Dilemma, exists *dominant strategy solution* $s \in S$: each player i is best off playing s_i, no matter what strategy vector $s' \in S$ the others choose:

$$u_i(s_i, s'_{-i}) \geq u_i(s'_i, s'_{-i})$$

As in Vickrey example, one goal of mechanism design is to design games with (good!) dominant strategy solutions.

Pure Strategy Nash Equilibria

Strategy vector $s \in S$ is a *Nash Equilibrium* if $\forall i$, $\forall s'_i \in S_i$,

$$u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$$

- need not be optimal
- can exist multiple equilibria with very different quality
- not clear how to compute, select, and coordinate. . .

Mixed Strategy Nash Equilibria

Allow randomizing, *risk-neutral* players who attempt to maximize *expected payoff*.

1 \ 2	H	T
H	-1 1	1 -1
T	1 -1	-1 1

Mixed Strategy Nash Equilibria

Allow randomizing, *risk-neutral* players who attempt to maximize *expected payoff*.

2 1	H	T
H	-1 1	1 -1
T	1 -1	-1 1

Theorem

Any game with a finite player set and finite strategy set has a mixed Nash equilibrium.

Outline

1. Algorithmic Mechanism Design
2. Defining and Quantifying the Effects of Selfishness
3. Algorithmic? Game Theory?
4. This term

Course goals

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

Course goals

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

I hope that by the end of the term, you will

- have the background (or know how to get the background) to read the emerging literature, engage in discussion, and attend talks in the area
- have thought deeply about some active topics of research
- have explored the interface between algorithmic game theory and your other areas of interest/expertise

Course content

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

Course content

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

Two main topics:

- Quantifying the cost of selfish behavior (with an emphasis on learning)
- Algorithmic Mechanism Design (with an emphasis on the Bayesian)

Also: guest lectures, special topics (as time allows), presentations on your projects

Course mechanics

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

Course mechanics

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

- NO "homework"
- NO exams
- Participation (20%): includes occasional surveys, self-assessments, assessments of classmates' presentations.
- Teaching (20%): required once; present core topics.
- Reaction paper (20%): 4-6 pages synthesizing, reflecting on, and engaging with literature.
- Project (40%): substantial engagement with a current topic of research. Deliverables: a short proposal, a 10-12 page report, and an in-class presentation.

Maturity and integrity

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

Maturity and integrity

The goal of this course is to engage with current topics of research at the interface between economics and computer science.

- Most of us will have to do extra outside reading. As needed, seek guidance on selecting these readings.
- You'll be selecting one or two areas to focus on. Again, seek guidance.
- Academic integrity: This course will require you to present and build on many existing sources. Proper attribution of ideas, text, and paraphrased text is required, in both oral and written presentation.

This is your course! Communicate about content/pace. And have fun!

Survey

Google “Katrina Ligett” to find course website (URL unstable)