CPS 590.2: Computation, Information, and Learning in Market Design

Instructor: Vincent Conitzer
conitzer@cs.duke.edu

Michael Albert
malbert@cs.duke.edu

Course web page:
http://www.cs.duke.edu/courses/spring17/compsci590.2/
ACM Transactions on Economics and Computation (TEAC)
John von Neumann

- Computer Science & Engineering
  - Computer architecture (von Neumann architecture)
- Economic Theory
  - Game theory (minimax theorem)
  - Linear programming (duality)
- Mathematical Optimization & Operations Research

Timeline:
- 1900
- 1950
- 2000
Who Are We?

We are a group of Duke University faculty, postdocs, and students interested in the intersection of computer science and economics (and the social sciences more broadly) and the impact of this interplay on decisions in information technology and digital business. This includes applying techniques from computer science and optimization to economics -- for example, using computation to design market clearing mechanisms and to implement efficient allocation and pricing in them -- as well as applying techniques from economics to computer science -- for example, designing incentives for users of networked computer systems and social networks.

Contacts

For organizational questions about the seminar series:
- Yuqian Li
- Catherine Moon

For other matters, contact the relevant faculty member(s):
- Atila Abdulkadiroglu (Econ)
- Vincent Conitzer (CS)
- Rachel Kranton (Econ)
- Ben Lee (ECE)
- Kamesh Munagala (CS)
Master’s of Science in Economics and Computation (MSEC)

The MSEC degree is a joint master’s program of the Departments of Economics and Computer Science. The joint field of economics and computation has recently emerged from two converging intellectual needs, which has created the opportunity for a truly interdisciplinary program.

The MSEC program is the outcome of exciting developments across the two fields:

- Computer science is becoming increasingly important for economists addressing complex questions on large repositories of data;

- The explosion of computer uses in all areas of life has made it necessary for computer scientists to understand the economics of computing systems; and,

- Computer scientists may now also analyze informational and financial transactions between people, businesses, governments, and electronic agents in economic terms.

"Macroeconomic problems are almost always analytically intractable and therefore require a computer to solve ... Having a background in computer science made it much easier to learn (computational) methods, as well as apply cutting-edge advances in technology to economic problems."

David Klemish
MSEM

"Tech giants such as Google, Microsoft, Yahoo, and Facebook are in many ways ahead of academia in getting computer scientists and economists to work together, and interest in these topics extends well beyond this group of companies. With this Master’s program, Duke is at the forefront of catching up with industry in breaking down boundaries."

Vince Conitzer
Department of Computer Science

Application Deadline

For Fall 2014: February 15, 2014

Charles Becker, MSEC Program Director
What is Economics?

• “Economics is a social science concerned with the factors that determine the production, distribution, and consumption of goods and services.” [Wikipedia, Jan. 2017]

• Some key concepts:
  – Economic agents or players (individuals, households, firms, …)
  – Agents’ current endowments of goods, money, skills, …
  – Possible outcomes ((re)allocations of resources, tasks, …)
  – Agents’ preferences or utility functions over outcomes
  – Agents’ beliefs (over other agents’ utility functions, endowments, production possibilities, …)
  – Agents’ possible decisions/actions
  – Mechanism that maps decisions/actions to outcomes
An economic picture

$v(\text{computer}) = 200$

$v(\text{monitor}) = 100$
$v(\text{laptop}) = 400$

$v(\text{server}) = 800$
$v(\text{network}) = 400$

$\$600$

$\$200$
After trade (a more efficient outcome)

$v(\text{desktop}) = 200$

$v(\text{computer}) = 100$
$v(\text{laptop}) = 400$

$v(\text{computer}) = 200$
$v(\text{laptop}) = 400$

$\$ 1100$

$\$ 400$

$\$ 100$

… but how do we get here? Auctions? Exchanges? Unstructured trade?
Some distinctions in economics

• **Descriptive vs. normative economics**
  – Descriptive:
    • seeks only to describe real-world economic phenomena
    • does not care if this is in any sense the “right” outcome
  – Normative:
    • studies how people “should” behave, what the “right” or “best” outcome is

• **Microeconomics vs. macroeconomics**
  – Microeconomics: analyzes decisions at the level of individual agents
    • deciding which goods to produce/consume, setting prices, …
    • “bottom-up” approach
  – Macroeconomics: analyzes “the sum” of economic activity
    • interest rates, inflation, growth, unemployment, government spending, taxation, …
    • “big picture”
What is Computer Science?

“Computer science [...] is the scientific and practical approach to computation and its applications. [...] A computer scientist specializes in the theory of computation and the design of computational systems.” [Wikipedia, Jan. 2017]

A computational problem is given by a function $f$ mapping inputs to outputs

- For integer $x$, let $f(x) = 0$ if $x$ is prime, 1 otherwise
- For an initial allocation of resources $x$, let $f(x)$ be the (re)allocation that maximizes the sum of utilities

An algorithm is a fully specified procedure for computing $f$

- E.g., sieve of Eratosthenes
- A correct algorithm always returns the right answer
- An efficient algorithm returns the answer fast

Computer science is also concerned with building larger artifacts out of these building blocks (e.g., personal computers, spreadsheets, the Internet, the Web, search engines, artificial intelligence, …)
Resource allocation as a computational problem

**input**

- \( v(\text{dollars}) = \$400 \)
- \( v(\text{computers}) = \$600 \)

**output**

- \( v(\text{dollars}) = \$800 \)
- \( v(\text{computers}) = \$400 \)
- \( v(\text{dollars}) = \$750 \)
- \( v(\text{computers}) = \$450 \)

Here, gains from trade (\$300) are divided evenly (not essential)
Combinatorial auctions

Simultaneously for sale:  

- bid 1: \( v( \text{Server, Printer} ) = $500 \)
- bid 2: \( v( \text{Server, Monitor} ) = $700 \)
- bid 3: \( v( \text{Laptop} ) = $300 \)

used in truckload transportation, industrial procurement, radio spectrum allocation, …
Financial securities

• Tomorrow there must be one of

• Agent 1 offers $5 for a security that pays off $10 if or

• Agent 2 offers $8 for a security that pays off $10 if or

• Agent 3 offers $6 for a security that pays off $10 if

• Can we accept some of these at offers at no risk?
Prediction markets

Which party will win the 2020 U.S. presidential election?

Market Type: Linked
End Date: N/A
Status: Open

Trade shares from this page by clicking any price in bold. For more information on an individual prediction, click on the name or image.

<table>
<thead>
<tr>
<th>Party</th>
<th>Latest</th>
<th>Buy Yes</th>
<th>Sell Yes</th>
<th>Buy No</th>
<th>Sell No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Republican</td>
<td>51¢</td>
<td>51¢</td>
<td>50¢</td>
<td>50¢</td>
<td>49¢</td>
</tr>
<tr>
<td>Democratic</td>
<td>50¢</td>
<td>50¢</td>
<td>49¢</td>
<td>51¢</td>
<td>50¢</td>
</tr>
<tr>
<td>Libertarian</td>
<td>3¢</td>
<td>4¢</td>
<td>3¢</td>
<td>97¢</td>
<td>96¢</td>
</tr>
<tr>
<td>Green</td>
<td>3¢</td>
<td>4¢</td>
<td>3¢</td>
<td>97¢</td>
<td>96¢</td>
</tr>
</tbody>
</table>
Economic mechanisms

**“true” input**

\[ v(\text{agent 1}) = 400 \]
\[ v(\text{agent 2}) = 600 \]

**agents’ bids**

\[ v(\text{agent 1}) = 500 \]
\[ v(\text{agent 2}) = 501 \]

**result**

\[ \text{Exchange mechanism (algorithm)} \]
\[ v(\text{result}) = 800 \]
\[ v(\text{result}) = 400 \]

Exchange mechanism designer does not have direct access to agents’ private information

Agents will selfishly respond to incentives
How to incentivize a weather forecaster

- Forecaster’s bonus can depend on
  - Prediction
  - Actual weather on predicted day
- Reporting true beliefs should maximize expected bonus
Peer prediction

I had a good experience with product X.

Each forecaster’s bonus depends only on how well it matches the other’s

I had an OK experience with product X.

5/5

4/5
What is game theory?

• “Game theory is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers." Game theory is mainly used in economics, political science, and psychology, as well as logic, computer science, and biology.” [Wikipedia, Jan. 2017]
What is game theory...

- Game theory studies settings where multiple parties (agents) each have
  - different preferences (utility functions),
  - different actions that they can take
- Each agent’s utility (potentially) depends on all agents’ actions
  - What is optimal for one agent depends on what other agents do
    - Very circular!
- Game theory studies how agents can rationally form beliefs over what other agents will do, and (hence) how agents should act
  - Useful for acting as well as predicting behavior of others
Penalty kick example

Is this a "rational" outcome? If not, what is?
Game playing & AI

**perfect information games:** no uncertainty about the state of the game (e.g. tic-tac-toe, chess, Go)

- Optimal play: value of each node = value of optimal child for current player (**backward induction**, minimax)
- For chess and Go, tree is too large
  - Use other techniques (heuristics, limited-depth search, alpha-beta, deep learning, ...)
- Top computer programs better than humans in chess, **not yet in Go**

**imperfect information games:** uncertainty about the state of the game (e.g. poker)

- Player 2 **cannot distinguish** nodes connected by dotted lines
  - Backward induction fails; need more sophisticated game-theoretic techniques for optimal play
- Small poker variants can be solved optimally
- Humans still better than top computer programs at full-scale poker (at least most versions?)
- Top computer (heads-up) poker players are based on techniques for game theory
Real-world security applications

Airport security
Where should checkpoints, canine units, etc. be deployed?

Federal Air Marshals
Which flights get a FAM?

US Coast Guard
Which patrol routes should be followed?

Wildlife Protection
Where to patrol to catch poachers or find their snares?
Questions and problems in (computational) game theory

• How should we represent games (=strategic settings)?
  – Standard game-theoretic representations not always concise enough

• What does it mean to solve a game?
  – Solution concepts from game theory, e.g., Nash equilibrium

• How computationally hard is it to solve games?
  – Can we solve them approximately?

• Is there a role for (machine) learning in games?

• What types of modeling problems do we face when addressing real-world games?
  – E.g., applications in security

• …
What is mechanism design?

• “Mechanism design is a field in economics and game theory that takes an engineering approach to designing economic mechanisms or incentives, toward desired objectives, in strategic settings, where players act rationally. […] Two distinguishing features of [mechanism design] are:
  – that a game “designer” chooses the game structure rather than inheriting one
  – that the designer is interested in the game’s outcome
• [Wikipedia, Jan. 2017]
Mechanism design...

- **Mechanism** = rules of auction, exchange, ...
- A **function** that takes reported preferences (bids) as input, and produces outcome (allocation, payments to be made) as output

\[
f(v) = \begin{cases} 
\text{The entire function } f \text{ is one mechanism} \\
\text{E.g., the mechanism from before: find allocation that maximizes (reported) utilities, distribute (reported) gains evenly} \\
\text{Other mechanisms choose different allocations, payments}
\end{cases}
\]
Example: (single-item) auctions

- **Sealed-bid** auction: every bidder submits bid in a sealed envelope
- **First-price** sealed-bid auction: highest bid wins, pays amount of own bid
- **Second-price** sealed-bid auction: highest bid wins, pays amount of second-highest bid

<table>
<thead>
<tr>
<th>Bid</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid 1</td>
<td>$10</td>
</tr>
<tr>
<td>Bid 2</td>
<td>$5</td>
</tr>
<tr>
<td>Bid 3</td>
<td>$1</td>
</tr>
</tbody>
</table>

- **First-price**
  - bid 1 wins, pays $10
- **Second-price**
  - bid 1 wins, pays $5
Which auction generates more revenue?

- Each bid depends on
  - bidder’s true valuation for the item (utility = valuation - payment),
  - bidder’s beliefs over what others will bid (→ game theory),
  - and... the auction mechanism used

- In a first-price auction, it does not make sense to bid your true valuation
  - Even if you win, your utility will be 0…

- In a second-price auction, (we will see later that) it always makes sense to bid your true valuation

Are there other auctions that perform better? How do we know when we have found the best one?
Mechanism design...

• Mechanism = game
• → we can use game theory to predict what will happen under a mechanism
  – if agents act strategically

• When is a mechanism “good”?
  – Should it result in outcomes that are good for the reported preferences, or for the true preferences?
  – Should agents ever end up lying about their preferences (in the game-theoretic solution)?
  – Should it always generate the best allocation?
  – Should agents ever burn money?(!?)

• Can we solve for the optimal mechanism?
Other kinds of private information in auctions?

my quality estimate
q( ) = 90

my need
n( ) = 70

my quality estimate
q( ) = 80

my need
n( ) = 75

How should the auctioneer use this information?
Sponsored search / ad auctions

- Choice of ads (if any) to show determined by:
  - Advertiser bid
  - Predicted likelihood of click
Many uses of linear programming, mixed integer (linear) programming in this area

<table>
<thead>
<tr>
<th>Game theory</th>
<th>Linear programming</th>
<th>Mixed integer linear programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dominated strategies</td>
<td>Nash equilibrium</td>
</tr>
<tr>
<td></td>
<td>Minimax strategies</td>
<td>Optimal mixed strategies to</td>
</tr>
<tr>
<td></td>
<td>Correlated equilibrium</td>
<td>commit to in more complex</td>
</tr>
<tr>
<td></td>
<td>Optimal mixed strategies to commit to</td>
<td>settings</td>
</tr>
<tr>
<td>Social choice, expressive marketplaces</td>
<td>Winner determination in auctions, exchanges, … with partially acceptable bids</td>
<td>Winner determination in: auctions, exchanges, … without partially acceptable bids; Kemeny, Slater, other voting rules; kidney exchange</td>
</tr>
<tr>
<td>Mechanism design</td>
<td>Automatically designing optimal mechanisms that use randomization</td>
<td>Automatically designing optimal mechanisms that do not use randomization</td>
</tr>
</tbody>
</table>
Learning & mechanism design

- Auctioneer may need to know **distribution over private information** to design optimal mechanism
  - Learn from repeated play?
  - Exploration/exploitation tradeoffs?
- Conversely: what if data for ML is generated by strategic agents?
  - Agent being classified herself
  - Data from multiple agents used to find social-welfare maximizing decision
Deferred Acceptance algorithm

[Gale & Shapley 1962]
Why should economists care about computer science?

• Finding efficient allocations of resources is a (typically hard) computational problem
  – Sometimes beyond current computational techniques
  – If so, unlikely that any market mechanism will produce the efficient allocation (even without incentives issues)
  – Market mechanisms must be designed with computational limitations in mind
  – New algorithms allow new market mechanisms
Why should economists care about computer science…

- **Agents** also face difficult computational problems in participating in the market
  - Especially acting in a game-theoretically optimal way is often **computationally hard**
  - Game-theoretic predictions **will not come true** if they cannot be computed
    - Sometimes bad (e.g., want agents to find right bundle to trade)
    - Sometimes good (e.g., do not want agents to manipulate system)
Why should computer scientists care about economics?

• Economics provides high-value computational problems

• Interesting technical twist: no direct access to true input, must incentivize agents to reveal true input

• Conversely: Computer systems are increasingly used by multiple parties with different preferences (e.g., Internet)

• Economic techniques must be used to
  – predict what will happen in such systems,
  – design the systems so that they will work well

• Game theory is relevant for artificial intelligence
  – E.g., computer poker
What is social choice?

• “Social choice theory or social choice is a theoretical framework for analysis of combining individual opinions, preferences, interests, or welfares to reach a collective decision or social welfare in some sense.” [Wikipedia, Jan. 2017]

• I.e., making decisions based on the preferences of multiple agents

• Largely, but not exclusively, focused on voting
Voting over outcomes

- Can vote over other things too
  - Where to go for dinner tonight, other joint plans, …
- Many different rules exist for selecting the winner
Kidney exchange

patient 1

patient 2

patient 3

patient 4

donor 1
(patient 1’s friend)

donor 2
(patient 2’s friend)

donor 3
(patient 3’s friend)

donor 4
(patient 4’s friend)

compatibilities
Problems in computational social choice

- **Winner determination problem**
  - For some voting rules, determining the winner is NP-hard
  - In a combinatorial auction, deciding which bids win is (in general) an NP-hard problem

- **Preference elicitation (communication) problem**
  - Can be impractical to communicate all of one’s preferences (e.g., valuation for every bundle)

- **Mechanism design problem**
  - How do we get the bidders to behave so that we get good outcomes?

- **These problems interact in nontrivial ways**
  - E.g. limited computational or communication capacity can limit mechanism design options
  - … but can perhaps also be used in a positive way