CS 886: Game-theoretic methods for computer science

Kate Larson

Computer Science
University of Waterloo

September 11, 2006

Outline

1. Introduction
   - Introduction
   - Two Communities
2. This Course
3. Examples
   - Selfish Routing
   - London Bus System

Outline

1. Introduction
   - Introduction
   - Two Communities
2. This Course
3. Examples
   - Selfish Routing
   - London Bus System

Introduction

- Growth in settings where there are multiple self-interacting parties
  - Networks
  - Electronic marketplaces
  - Game playing
  - . . .
- For participants to act optimally in such settings, they must take into account how other agents are going to act.
- We want to be able to
  - Understand the ways agents will interact and behave
  - Provide incentives so that agents behave the way **we** would like them to

Outline

Two Communities

Economics

- Traditional emphasis on game theoretic rationality
- Describing how agents should behave
- Multiple self-interested agents

Computer Science

- Traditional emphasis on computational and informational constraints
- Building agents
- Individual or cooperative agents

New Research Problems

- How do we use game theory and mechanism design in computer science settings?
- How do we resolve conflicts between game-theoretic and computational constraints?
- Development of new theories and methodologies

New Research Area

Explosion of research in the area (Algorithmic game theory, computational mechanism design, Distributed algorithmic mechanism design, computational game theory,...)

- Papers appearing in AAAI, AAMAS, UAI, NIPS, PODC, SIGCOMM, INFOCOMM, SODA, STOC, FOCS, ...
- Papers by CS researchers appearing in Games and Economic Behavior, Journal of Economic Theory, Econometrica,...
- Numerous workshops and meetings,...

This Course

The goals of this course

- Introduction to game theory and mechanism design
- Study how they are used in computer science
- Study computational issues that arise

Course structure

- Introductory lectures
- Current research papers

Topics

- Game theory
  - Normal form and extensive form games
  - Dominance and iterated dominance
  - Minimax strategies, Nash equilibrium, correlated equilibrium, backward induction, subgame perfect equilibrium, Bayesian games
  - Repeated games, Folk theorems
  - Coalitional game theory
- Social choice
  - Arrow's theorem, voting
- Mechanism Design
  - Incentive compatibility, individual rationality, positive and negative results, Revelation principle, VIckrey-Clarke-Groves mechanisms
  - Auctions (both single-item and combinatorial)
- Applications

Topics

- Game theory
  - Normal form and extensive form games
  - Dominance and iterated dominance
  - Minimax strategies, Nash equilibrium, correlated equilibrium, backward induction, subgame perfect equilibrium, Bayesian games
  - Repeated games, Folk theorems
  - Coalitional game theory
- Social choice
  - Arrow's theorem, voting
- Mechanism Design
  - Incentive compatibility, individual rationality, positive and negative results, Revelation principle, VIckrey-Clarke-Groves mechanisms
  - Auctions (both single-item and combinatorial)
- Applications

Topics

- Game theory
  - Normal form and extensive form games
  - Dominance and iterated dominance
  - Minimax strategies, Nash equilibrium, correlated equilibrium, backward induction, subgame perfect equilibrium, Bayesian games
  - Repeated games, Folk theorems
  - Coalitional game theory
- Social choice
  - Arrow's theorem, voting
- Mechanism Design
  - Incentive compatibility, individual rationality, positive and negative results, Revelation principle, VIckrey-Clarke-Groves mechanisms
  - Auctions (both single-item and combinatorial)
- Applications

Topics

- Game theory
  - Normal form and extensive form games
  - Dominance and iterated dominance
  - Minimax strategies, Nash equilibrium, correlated equilibrium, backward induction, subgame perfect equilibrium, Bayesian games
  - Repeated games, Folk theorems
  - Coalitional game theory
- Social choice
  - Arrow's theorem, voting
- Mechanism Design
  - Incentive compatibility, individual rationality, positive and negative results, Revelation principle, VIckrey-Clarke-Groves mechanisms
  - Auctions (both single-item and combinatorial)
- Applications

Prerequisites

- No formal prerequisites
- Students should be comfortable with mathematical proofs
- Some familiarity with probability
- Ideally students will have an AI course and background in algorithms and complexity
- **I will cover the game theory and mechanism design required**

Grading

- 2-3 assignments on game theory and mechanism design: 10%
- In class presentation(s): 20%
  - Peer-reviewed
- Class participation: 20%
- Research project: 50%

Presentations

Every student is responsible for presenting a research paper in class

- Short survey + a critique
- Everyone in class will provide feedback on the presentation
- Marks given on coverage of material + organization + presentation

Class Participation

You must participate!

- Before each class (before 6:00 am the day of the presentation) you must email me a list of comments on the paper to be presented[1]
  - What is the main contribution?
  - Is it important? Why?
  - What assumptions are made?
  - What applications might arise from the results?
  - How can it be extended?
  - What was unclear?
  - ...

[1] Plain text please. No attachments!

Projects

The goal of the project is to develop a deep understanding of a topic related to the course.

- The topic is open
  - Theoretical, experimental, *in-depth* literature review,...
  - Can be related to your own research
  - If you have trouble coming up with a topic, come and talk to me
- Proposals due October 18
- Final projects due December 6th[2]
- Students will present projects in class

[2] I will likely be flexible with this.

Projects

The goal of the project is to develop a deep understanding of a topic related to the course.

- The topic is open
  - Theoretical, experimental, *in-depth* literature review,...
  - Can be related to your own research
  - If you have trouble coming up with a topic, come and talk to me
- Proposals due October 18
- Final projects due December 6th[2]
- Students will present projects in class

[2] I will likely be flexible with this.

Projects

The goal of the project is to develop a deep understanding of a topic related to the course.

- The topic is open
  - Theoretical, experimental, *in-depth* literature review,...
  - Can be related to your own research
  - If you have trouble coming up with a topic, come and talk to me
- Proposals due October 18
- Final projects due December 6th[2]
- Students will present projects in class

[2] I will likely be flexible with this.

Projects

The goal of the project is to develop a deep understanding of a topic related to the course.

- The topic is open
  - Theoretical, experimental, *in-depth* literature review,...
  - Can be related to your own research
  - If you have trouble coming up with a topic, come and talk to me
- Proposals due October 18
- Final projects due December 6th[2]
- Students will present projects in class

[2] I will likely be flexible with this.

Projects

The goal of the project is to develop a deep understanding of a topic related to the course.

- The topic is open
  - Theoretical, experimental, *in-depth* literature review,...
  - Can be related to your own research
  - If you have trouble coming up with a topic, come and talk to me
- Proposals due October 18
- Final projects due December 6th[2]
- Students will present projects in class

[2] I will likely be flexible with this.

Other Information

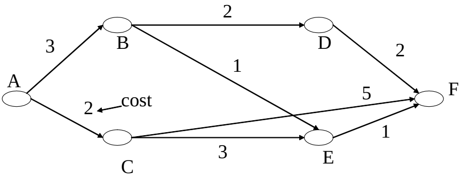
- Class times: Monday-Wednesday 10:00-11:30
- Office Hours: Mondays 1:30-2:30[3]
- Course website
  - `http://www.cs.uwaterloo.ca/~klarson/teaching/F06-`

[3] Except Sept 18, Oct 16, and Nov 20.

Outline

1. Introduction
   - Introduction
   - Two Communities
2. This Course
3. Examples
   - Selfish Routing
   - London Bus System

Selfish Routing



- We want to find the least-cost route from S to T.
- Costs are private information – we do not know them
- We do know that agents (nodes) are interested in maximizing revenue
- How can we use this to figure out the least-cost route?

Outline

London Bus System[4]

- 5 million passengers daily
- 7500 buses
- 700 routes
- The system has been privatized since 1997 by using competitive tendering
- *Idea*: Run an auction to allocate routes to companies

[4] As of April 2004

Auction Protocol

- Let G be set of all routes, I be the set of bidders
- Agent i submits bid $v_i(S)$ for all bundles $S \subseteq G$
- Compute allocation S^* to maximize sum of reported bids

$$V^*(I) = \max_{(S_1,\ldots,S_n)} \sum_i v_i(S_i)$$

- Compute best allocation without each agent

$$V^*(I \setminus i) = \max_{(S_1,\ldots,S_n)} \sum_{j \neq i} v_j^*(S_j)$$

- Allocate each agent S_i^*, each agent pays

$$P(i) = v_i^*(S_i^*) - [V^*(I) - V^*(I \setminus i)]$$

Auction Protocol

- Let G be set of all routes, I be the set of bidders
- Agent i submits bid $v_i(S)$ for all bundles $S \subseteq G$
- Compute allocation S^* to maximize sum of reported bids

$$V^*(I) = \max_{(S_1,\ldots,S_n)} \sum_i v_i(S_i)$$

- Compute best allocation without each agent

$$V^*(I \setminus i) = \max_{(S_1,\ldots,S_n)} \sum_{j \neq i} v_j^*(S_j)$$

- Allocate each agent S_i^*, each agent pays

$$P(i) = v_i^*(S_i^*) - [V^*(I) - V^*(I \setminus i)]$$

Auction Protocol

- Let G be set of all routes, I be the set of bidders
- Agent i submits bid $v_i(S)$ for all bundles $S \subseteq G$
- Compute allocation S^* to maximize sum of reported bids

$$V^*(I) = \max_{(S_1,\ldots,S_n)} \sum_i v_i(S_i)$$

- Compute best allocation without each agent

$$V^*(I \setminus i) = \max_{(S_1,\ldots,S_n)} \sum_{j \neq i} v_j^*(S_j)$$

- Allocate each agent S_i^*, each agent pays

$$P(i) = v_i^*(S_i^*) - [V^*(I) - V^*(I \setminus i)]$$

Auction Protocol

- Let G be set of all routes, I be the set of bidders
- Agent i submits bid $v_i(S)$ for all bundles $S \subseteq G$
- Compute allocation S^* to maximize sum of reported bids

$$V^*(I) = \max_{(S_1,\ldots,S_n)} \sum_i v_i(S_i)$$

- Compute best allocation without each agent

$$V^*(I \setminus i) = \max_{(S_1,\ldots,S_n)} \sum_{j \neq i} v_j^*(S_j)$$

- Allocate each agent S_i^*, each agent pays

$$P(i) = v_i^*(S_i^*) - [V^*(I) - V^*(I \setminus i)]$$

London Bus System

- Mechanism: Generalized Vickrey Auction
  - Specifies the rules
  - Describes how outcome will be determined
- Strategies
  - Policies which specify what actions to take
  - Agents are self-interested and rational
- GVA is efficient and strategy-proof

Computational Issues

- Winner determination problem: Select bids to maximize sum of reported values
  - Maximum weighted set packing (NP-hard)
  - Solve this problem $I + 1$ times
- Agent valuation problem
- Communication complexity
  - Each agent has to communicate 2^{700} bids to the auctioneer

Computational Issues

- Winner determination problem: Select bids to maximize sum of reported values
  - Maximum weighted set packing (NP-hard)
  - Solve this problem $I + 1$ times
- Agent valuation problem
- Communication complexity
  - Each agent has to communicate 2^{700} bids to the auctioneer

Computational Issues

- Winner determination problem: Select bids to maximize sum of reported values
  - Maximum weighted set packing (NP-hard)
  - Solve this problem $l + 1$ times
- Agent valuation problem
- Communication complexity
  - Each agent has to communicate 2^{700} bids to the auctioneer

Computational Issues

- Winner determination problem: Select bids to maximize sum of reported values
  - Maximum weighted set packing (NP-hard)
  - Solve this problem $l + 1$ times
- Agent valuation problem
- Communication complexity
  - Each agent has to communicate 2^{700} bids to the auctioneer