CS599: Algorithm Design in Strategic Settings
Fall 2012
Lecture 1: Introduction and Class Overview

Instructor: Shaddin Dughmi

Outline

Outline

We live in a world with scarce resources...

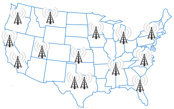
We live in a world with scarce resources...

we want to allocate these resources “optimally”.

We live in a world with scarce resources...

we want to allocate these resources “optimally”.

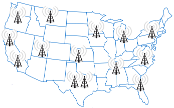
Electromagnetic Spectrum



We live in a world with scarce resources...

we want to allocate these resources "optimally".

Electromagnetic Spectrum



verizon at&t Sprint

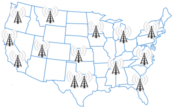
Google

Advertising space

We live in a world with scarce resources...

we want to allocate these resources “optimally”.

Electromagnetic Spectrum



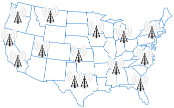
Advertising space

Content Distribution Networks

We live in a world with scarce resources...

we want to allocate these resources “optimally”.

Electromagnetic Spectrum



Advertising space

Content Distribution Networks

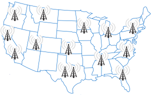
Take-off / landing slots

What is a “good” allocation?

- Utilitarian: maximize social welfare
- Maximize revenue
- Fairness
- ...

What is a “good” allocation?

- Utilitarian: maximize social welfare
- Maximize revenue
- Fairness
- . . .



verizon at&t Sprint

Licenses to companies best positioned to serve their customers with them.

Google

Space to advertisers most likely to generate business (clicks)

Akamai

Place servers/files on the Internet to best serve content providers' distribution needs.

Divide slots to maximize total air traveler satisfaction (on time flights)

Challenges

Economic Challenge

- Agents receiving goods/services/resources are self-interested.
- Quality of an allocation depends on private data of agents.
- Agents may strategically misrepresent this data.

Challenges

Economic Challenge

- Agents receiving goods/services/resources are self-interested.
- Quality of an allocation depends on private data of agents.
- Agents may strategically misrepresent this data.

Computational Challenge

Need to compute the desired outcome efficiently (i.e. in polynomial time)

Challenges

Economic Challenge

- Agents receiving goods/services/resources are self-interested.
- Quality of an allocation depends on private data of agents.
- Agents may strategically misrepresent this data.

Computational Challenge

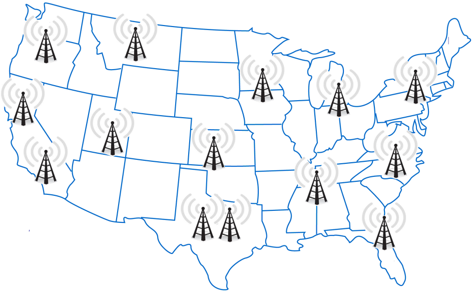
Need to compute the desired outcome efficiently (i.e. in polynomial time)

Motivating Question

Can we allocate resources in a desirable manner in the presence of self-interested behavior and limited computational power?

The field concerned with this question has come to be called algorithmic mechanism design

Example: Spectrum Auctions



verizon at&t Sprint

- Each telecom has a private value in $$ for each bundle of licenses
- Dependencies: Some of the licenses are substitutes/complements

Spectrum Auctions

FCC Statute

Design spectrum auctions that promote “efficient and intensive use” of the electromagnetic spectrum.

Spectrum Auctions

FCC Statute

Design spectrum auctions that promote "efficient and intensive use" of the electromagnetic spectrum.

Formal interpretation: Maximize social welfare of the allocation.

Definition (Social Welfare)

Sum of values of telecoms for the bundles they get.

Spectrum Auctions

FCC Statute

Design spectrum auctions that promote "efficient and intensive use" of the electromagnetic spectrum.

Formal interpretation: Maximize social welfare of the allocation.

Definition (Social Welfare)

Sum of values of telecoms for the bundles they get.

Can be defined more generally for abstract resource allocation problems.

Spectrum Auctions

Spectrum auctions in USA, UK, Germany, Sweden ...

- 1994-2001: More than $100 billion worth of licenses sold

Spectrum Auctions

Spectrum auctions in USA, UK, Germany, Sweden ...

- 1994-2001: More than $100 billion worth of licenses sold

FCC Auction 700MHz Band (2008)

- 1099 licenses
- 261 bidders
- $19 Billion in revenue

Spectrum Auctions

Spectrum auctions in USA, UK, Germany, Sweden ...

- 1994-2001: More than $100 billion worth of licenses sold

FCC Auction 700MHz Band (2008)

- 1099 licenses
- 261 bidders
- $19 Billion in revenue

> *...computing just one efficient allocation can be an inhumanly hard problem, and getting participants to reveal the information about their values necessary to do that computation is probably impossible.*
>
> \- Paul Milgrom

Spectrum Auctions

Spectrum auctions in USA, UK, Germany, Sweden ...

- 1994-2001: More than $100 billion worth of licenses sold

FCC Auction 700MHz Band (2008)

- 1099 licenses
- 261 bidders
- $19 Billion in revenue

> *...computing just one efficient allocation can be an inhumanly hard problem, and getting participants to reveal the information about their values necessary to do that computation is probably impossible.*
>
> \- Paul Milgrom

Economic Challenge

Mechanisms use incentives to extract private data.

Mechanism

1. Solicit preferences
2. Compute “good” allocation
3. Charge payments

Example: eBay Auction

1. Submit bids
2. Give to highest bidder
3. Charge second highest bid

Economic Challenge

Mechanisms use incentives to extract private data.

Mechanism

1. Solicit preferences
2. Compute “good” allocation
3. Charge payments

Example: eBay Auction

1. Submit bids
2. Give to highest bidder
3. Charge second highest bid

Truthfulness

A mechanism is truthful (aka incentive-compatible) if players maximize their utility by reporting their true preferences in the first step.

Economic Challenge

Mechanisms use incentives to extract private data.

Mechanism

1. Solicit preferences
2. Compute “good” allocation
3. Charge payments

Example: eBay Auction

1. Submit bids
2. Give to highest bidder
3. Charge second highest bid

Truthfulness

A mechanism is truthful (aka incentive-compatible) if players maximize their utility by reporting their true preferences in the first step.

Fact [Vickrey, Clarke, Groves]

Ignoring computational constraints, there is a truthful mechanism that computes an optimal allocation for any welfare maximization problem.

Computational Challenge

Computational Challenge

Need to compute allocation in polynomial time.

Computational Solution

- A rich theory of design and analysis of algorithms enables polynomial-time algorithms for some resource allocation problems.
- When problems NP-hard, theory of approximation algorithms enables polytime computation of “near optimal” allocations.
  - Approximation ratio: Percentage of optimal welfare on worst-case input.

Computational Challenge

Computational Challenge

Need to compute allocation in polynomial time.

Computational Solution

- A rich theory of design and analysis of algorithms enables polynomial-time algorithms for some resource allocation problems.
- When problems NP-hard, theory of approximation algorithms enables polytime computation of "near optimal" allocations.
  - Approximation ratio: Percentage of optimal welfare on worst-case input.

Frequently, we know the "optimal" approximation algorithm.

Combinatorial Auctions [Vondrak '08, Khot et al '05]

When valuations are submodular, there is a 63% approximation algorithm, and this is optimal assuming $P \neq NP$.

So Whats the Problem?

So Whats the Problem?

Difficulty

There seems to be tension between the economic goal of incentive-compatibility, and the computational goal of polynomial time.

So Whats the Problem?

Difficulty

There seems to be tension between the economic goal of incentive-compatibility, and the computational goal of polynomial time.

This tension has been a major focus of algorithmic mechanism design in recent years.

So Whats the Problem?

Difficulty

There seems to be tension between the economic goal of incentive-compatibility, and the computational goal of polynomial time.

This tension has been a major focus of algorithmic mechanism design in recent years.

This Class

We will study this tension, and algorithmic techniques developed to ameliorate it, using fundamental resource allocation problems as examples.

Outline

Course Goals

- Appreciate interplay between economic and computational considerations in algorithm design.
- Exposure to powerful algorithmic techniques and economic concepts
- Preparation for research in the burgeoning intersection of CS and Econ/Game theory

This class is NOT . . .

This class is NOT . . .

- an economics class,

This class is NOT . . .

- an economics class,
- a game theory class,

This class is NOT . . .

- an economics class,
- a game theory class,
- or even a mechanism design class!

This class IS . . .

. . . a theoretical CS class on algorithmic mechanism design.

- Focus will be on the interplay between computational goals (mainly, polynomial time) and economic goals (mainly incentive compatibility).
- Incentive compatibility will reduce to a combinatorial constraint on the algorithm, akin to restricted computational models (online, streaming, etc).
- Lectures and assignments will be mathematical proof-based.

Prerequisites

- Mathematical maturity: Be good at proofs
- Algorithms and Optimization at the graduate level:
  - CS670 or equivalent
  - Exposure to approximation algorithms
  - Exposure to LP
- Don’t worry, I will teach you all the econ/gt/md you need to know

Administrivia

- Lecture time: Fridays 2 pm - 4:50 pm
- Lecture place: KAP 145
- Instructor: Shaddin Dughmi
  - Email: shaddin@usc.edu
  - Office: SAL 234
  - Office Hours: Tuesday 1:30 - 3:30pm (subject to change)
- Course Homepage (to appear): www.cs.usc.edu/people/shaddin/cs599fa12
- References: AGT book (Nisan et al, editors), and Hartline's approximation in economic design book. Both available online, linked on website. Also, we will refer to research papers.

Requirements and Grading

- This is an advanced grad class, so grade is not the point.
  - I assume you want to learn this stuff.
  - If you can take pass/fail, please do.
- 3-4 homeworks, 70% of grade.
  - Proof based.
  - Challenging.
  - Discussion allowed, even encouraged, but must write up solutions independently.
- Problems in-class, 10% of grade.
- Research project or final, 20% of grade. Suggestions will be posted on website.
- One late homework allowed, 2 days. (too harsh?)

A Note on Lecture Length / Time

I don't want to listen to me talk for 3 hours on Friday late afernoon either

- Lecture portion will be ≈ 2 hours
- Remainder will be discussion and problem solving
- We can sometimes leave early (shhhh!)

Survey

- Undergrad, Ms, PhD?
- Grad algorithms class?
- Grad theory class?
- Exposure to approximation algorithms?
- Exposure to LP?
- Research project vs final?

Outline

Single-item Allocation

Single-item Allocation

First Price Auction

1. Collect bids
2. Give to highest bidder
3. Charge him his bid

Single-item Allocation

Second-price (Vickrey) Auction

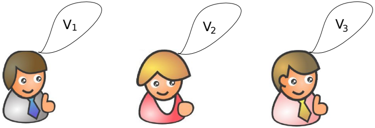
1. Collect bids
2. Give to highest bidder
3. Charge second highest bid

Single-item Allocation

Vickrey Auction with Reserve

1. Choose a reserve price r
2. Collect bids
3. If nobody bids above reserve, then cancel the auction, otherwise
4. Give to highest bidder
5. Charge the second highest bid or r, whichever is bigger

Example: Combinatorial Allocation

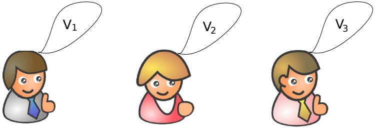


n players, m items.

Private valuation v_i : set of items $\to \mathbb{R}$.

- $v_i(S)$ is player i's value for bundle S.

Example: Combinatorial Allocation



n players, m items.

Private valuation v_i : set of items $\rightarrow \mathbb{R}$.

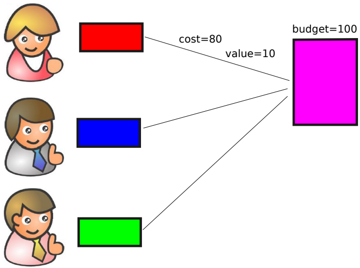
- $v_i(S)$ is player i's value for bundle S.

An auction would partition items into sets $S_1, \ldots, S_n$, possibly charging payments $p_1, \ldots, p_n$

Goals

- Welfare: Maximize $v_1(S_1) + v_2(S_2) + \ldots v_n(S_n)$
- Revenue: Maximize $p_1 + \ldots + p_n$
- Fairness: Maximize the minimum $v_i(S_i)$

Example: Knapsack Allocation



- n players, each player i with a task requiring c_i time
- Machine has total processing time B (public)
- Player i has (private) value v_i for his task

Must choose a feasible subset $S \subseteq [n]$ of the tasks to process, possibly charging players

Goals

- Welfare: maximize $\sum_{i \in S} v_i$
- Revenue

Commonalities

- There is a set of possible allocations
  - Single-item Allocation: The n different choices of winning player.
- There is a set of players, each of which has a private valuation function
  - Maps allocations to real numbers
  - Single item allocation: Player i's value for all allocations is 0, except for that in which he wins, where his value is some private quantity v_i.
- Want to choose a "good" outcome (allocation+payments), as a function of the private data.

Commonalities

- There is a set of possible allocations
  - Single-item Allocation: The n different choices of winning player.
- There is a set of players, each of which has a private valuation function
  - Maps allocations to real numbers
  - Single item allocation: Player i's value for all allocations is 0, except for that in which he wins, where his value is some private quantity v_i.
- Want to choose a "good" outcome (allocation+payments), as a function of the private data.

Challenges

- Economic: Agents invested in outcome and may have incentive to manipulate the input? (their reported valuation)
- Computational: The usual "can we do it in polynomial time" question

Mechanism Design

Mechanism Design

The study of computing with data owned by selfish agents.

Mechanism Design Problem

- Set Ω of allocations.
- Set of n players, each with private valuation $v_i : \Omega \to \mathbb{R}$. (aka type)

Mechanism Design

Mechanism Design

The study of computing with data owned by selfish agents.

Mechanism Design Problem

- Set Ω of allocations.
- Set of n players, each with private valuation $v_i : \Omega \to \mathbb{R}$. (aka type)

- Combinatorial allocation (n players, m items)
  - Ω is set of allocations of items $(S_1, \ldots, S_n)$
  - $v_i(S_1, \ldots, S_n)$ is player i's value for his bundle S_i (shorthand $v_i(S_i)$)

Mechanism Design

Mechanism Design

The study of computing with data owned by selfish agents.

Mechanism Design Problem

- Set Ω of allocations.
- Set of n players, each with private valuation $v_i : \Omega \to \mathbb{R}$. (aka type)

- Combinatorial allocation (n players, m items)
  - Ω is set of allocations of items $(S_1, \ldots, S_n)$
  - $v_i(S_1, \ldots, S_n)$ is player i's value for his bundle S_i (shorthand $v_i(S_i)$)
- Knapsack Allocation
  - Ω is family of subsets of tasks that fit in the knapsack
  - Value of a player i for a subset S is v_i if $i \in S$, otherwise 0

Mechanisms

We focus on the design of direct-revelation mechanisms in a setting where we may supplement allocation with a payment from each player.

Mechanism

1. Solicit valuations $v_1, \ldots, v_n$
2. Compute "good" allocation $\omega \in \Omega$
3. Charge payments $p_1, \ldots p_n$

Example: Vickrey Auction

1. Collect bids
2. Give to highest bidder
3. Charge second highest bid

Mechanisms

We focus on the design of direct-revelation mechanisms in a setting where we may supplement allocation with a payment from each player.

Mechanism

1. Solicit valuations $v_1, \ldots, v_n$
2. Compute "good" allocation $\omega \in \Omega$
3. Charge payments $p_1, \ldots p_n$

Example: Vickrey Auction

1. Collect bids
2. Give to highest bidder
3. Charge second highest bid

Helpful to separate a mechanism into:

- Allocation rule $\mathcal{A}$ mapping $(v_1, \ldots, v_n)$ to allocations $\omega \in \Omega$
- Payment rule p mapping $(v_1, \ldots, v_n)$ to payments $(p_1, \ldots, p_n)$.

Mechanisms and Games

If players knew each other's valuations, we get a game of complete information

Vickrey Auction

A painting is being sold in a second price auction. There are two players, with public values $v_1 = \$1$ and $v_2 = \$2$. Bids may either be $\$1$ or $\$2$. What are the stable bid profiles?

Mechanisms and Games

If players knew each other's valuations, we get a game of complete information

Vickrey Auction

A painting is being sold in a second price auction. There are two players, with public values $v_1 = \$1$ and $v_2 = \$2$. Bids may either be $\$1$ or $\$2$. What are the stable bid profiles?

Assume: Quasilinear utility

Winning player has utility $v_i - p_i$, losing player has utility 0.

Mechanisms and Games

If players knew each other's valuations, we get a game of complete information

Vickrey Auction

A painting is being sold in a second price auction. There are two players, with public values $v_1 = \$1$ and $v_2 = \$2$. Bids may either be $\$1$ or $\$2$. What are the stable bid profiles?

Assume: Quasilinear utility

Winning player has utility $v_i - p_i$, losing player has utility 0.

Write down the game matrix.

		P2: 1	P2: 2
P1	1		
	2		

Mechanisms and Games

If players knew each other's valuations, we get a game of complete information

Vickrey Auction

A painting is being sold in a second price auction. There are two players, with public values $v_1 = \$1$ and $v_2 = \$2$. Bids may either be $\$1$ or $\$2$. What are the stable bid profiles?

Assume: Quasilinear utility

Winning player has utility $v_i - p_i$, losing player has utility 0.

Write down the game matrix.

P2

P1		1	2
	1	(0,1/2)	(0,1)
	2	(0,0)	(-1/2,0)

Mechanisms and Games

If players knew each other's valuations, we get a game of complete information

Vickrey Auction

A painting is being sold in a second price auction. There are two players, with public values $v_1 = \$1$ and $v_2 = \$2$. Bids may either be $\$1$ or $\$2$. What are the stable bid profiles?

Assume: Quasilinear utility

Winning player has utility $v_i - p_i$, losing player has utility 0.

Write down the game matrix.

Two Pure Nash equilibria.

		P2	
		1	2
P1	1	(0,1/2)	(0,1)
	2	(0,0)	(-1/2,0)

Mechanisms and Games

P2

		1	2
P1	1	(0,1/2)	(0,1)
	2	(0,0)	(-1/2,0)

Two critiques of the full-information Nash equilibrium as the prediction:

- Informational: Players can't play at equilibrium because they don't know the game they are playing!
- Equilibrium selection: Which one is a "better" prediction of reality?

P2

P1		1	2
	1	(0,1/2)	(0,1)
	2	(0,0)	(-1/2,0)

One equilibrium stands out,

Fact

The Vickrey mechanism is dominant-strategy incentive-compatible (DSIC): no matter what other players do, a player never loses by bidding his value. And in fact, truth-telling is the only dominant strategy.

In other words, truth-telling is a "very stable" equilibrium, robust to uncertainty in other player's actions, and is the only such equilibrium.

Dealing with Incomplete Information

In general, two main approaches to dealing with these problems:

1. Prior-free:
   - No assumption on what agents know about each other.
   - Dominant strategy equilibrium is a choice, for each i and v_i, of an action $\widehat{v}_i$, such that $\widehat{v}_i$ is a best response regardless of $\widehat{v}_{-i}$
   - Design mechanisms that have a "good" DSE

Dealing with Incomplete Information

In general, two main approaches to dealing with these problems:

1. Prior-free:
   - No assumption on what agents know about each other.
   - Dominant strategy equilibrium is a choice, for each i and v_i, of an action $\widehat{v}_i$, such that $\widehat{v}_i$ is a best response regardless of $\widehat{v}_{-i}$
   - Design mechanisms that have a "good" DSE

Example: Vickrey Auction

Truth-telling is a dominant strategy equilibrium in the Vickrey Auction. Moreover, it is a "good" equilibrium for a utilitatrian auctioneer because the player who most values the item gets it.

Dealing with Incomplete Information

In general, two main approaches to dealing with these problems:

2. Bayesian common prior:
   - Player types are drawn from a publicly known distribution (say independent for now)
   - Bayesian Nash equilibrium is a choice, for each player i and each type v_i of his, of a report (bid) $\widehat{v}_i$, such that $\widehat{v}_i$ is a best response to $\widehat{v}_{-i}$ in expectation over draws of v_{-i}.
   - Design mechanisms where there is a "good" BNE in expectation

Dealing with Incomplete Information

In general, two main approaches to dealing with these problems:

2. Bayesian common prior:
   - Player types are drawn from a publicly known distribution (say independent for now)
   - Bayesian Nash equilibrium is a choice, for each player i and each type v_i of his, of a report (bid) $\widehat{v}_i$, such that $\widehat{v}_i$ is a best response to $\widehat{v}_{-i}$ in expectation over draws of v_{-i}.
   - Design mechanisms where there is a "good" BNE in expectation

Example: All-pay auction

- n players with values i.i.d from $[0, 1]$.
- All-pay auction: Give to highest bidder, charge each player i the amount $(1 - 1/n)v_i^n$
- Fact: truth-telling is a BNE, resulting in the utilitarian allocation.

Mechanism Design and Game Theory

- Whichever worldview you choose (Bayesian or Prior-free), you have an equilibrium concept (BNE or DSE).

Task of Mechanism design

Design a mechanism which guarantees a “good” equilibrium

- Single-item auction: Welfare, revenue
- Knapsack auction: welfare, revenue
- Combinatorial auction: welfare, revenue, fairness

Mechanism Design and Game Theory

- Whichever worldview you choose (Bayesian or Prior-free), you have an equilibrium concept (BNE or DSE).

Task of Mechanism design

Design a mechanism which guarantees a “good” equilibrium

- Single-item auction: Welfare, revenue
- Knapsack auction: welfare, revenue
- Combinatorial auction: welfare, revenue, fairness

Mechanism design is “reverse game theory.”

Incentive-compatibility

Luckily, our task simplifies further.

Definition

A mechanism is truthful (aka incentive compatible) if truth-telling is an equilibrium.

Revelation Principle

If there is a mechanism that implements an outcome $(\mathcal{A}(v), p(v))$ in equilibrium, then there is also a truthful mechanism that implements the same outcome in truth-telling equilibrium.

Therefore, as a designer it suffices to restrict attention to designing truthful mechanisms.

(Simplified) Task of Mechanism Design

Given resource allocation problem and an objective (welfare, revenue, fairness, . . .), design a truthful mechanism that guarantees a “good” outcome.

(Simplified) Task of Mechanism Design

Given resource allocation problem and an objective (welfare, revenue, fairness, . . .), design a truthful mechanism that guarantees a “good” outcome.

In a truthful mechanism, you may think of the “bids” as the true values. You are working with the right inputs.

(Simplified) Task of Mechanism Design

Given resource allocation problem and an objective (welfare, revenue, fairness, . . .), design a truthful mechanism that guarantees a "good" outcome.

In a truthful mechanism, you may think of the "bids" as the true values. You are working with the right inputs.

- Single-item allocation: Vickrey optimal for welfare. Myerson optimal for revenue (Bayesian settings).
- Knapsack allocation, combinatorial auctions, . . .
  - Vickrey-Clarke-Groves optimal for welfare, but not polytime.
  - Revenue: ???

Achievements of Mechanism Design

- Revelation Principle
- The welfare-optimal Vickrey-Clarke-Groves Mechanism
- Myerson's revenue-optimal single-item auction
- Revenue equivalence theorems
- ...

Algorithm Design

Algorithm Design

The study of computing with limited resources (e.g. polynomial time).

Algorithm Design

Algorithm Design

The study of computing with limited resources (e.g. polynomial time).

Main Challenge: NP-hardness

Unless P=NP, no “optimal” algorithm exists for many resource allocation problems.

Algorithm Design

Algorithm Design

The study of computing with limited resources (e.g. polynomial time).

Main Challenge: NP-hardness

Unless P=NP, no “optimal” algorithm exists for many resource allocation problems.

TCS answer: Approximation Algorithms

Algorithms that compute a “near optimal” solution

Algorithm Design

Algorithm Design

The study of computing with limited resources (e.g. polynomial time).

Main Challenge: NP-hardness

Unless P=NP, no "optimal" algorithm exists for many resource allocation problems.

TCS answer: Approximation Algorithms

Algorithms that compute a "near optimal" solution

- Knapsack Allocation: Fully Polynomial-time Approximation Scheme
- Combinatorial Allocation: Approximation ratio depends on assumptions on valuations.

Algorithmic Mechanism Design

Main Question

For which resource allocation problems can we design (approximately) optimal mechanisms that are truthful and also run in polynomial time?

Challenge

Incentive compatibility and polynomial-time implementation can not be "cut and pasted" together. Requires new algorithmic techniques.

This will send us through a tour of algorithms and optimization, involving approximation algorithms, linear programming, polytope theory, smoothed complexity, and convex analysis

Outline

Game Theory and Mechanism Design Basics

- Complete information Games and Nash equilibrium
- Games of incomplete information, dominant strategy and Bayesian equilibria.
- Mechanisms, revelation principle, incentive compatibility

Approximation Algorithms and Optimization (??)

- Linear Programming
- Approximation Algorithms

Outline

Single-parameter Problems

First half of the class will focus on the prior-free model. We begin with

Single-parameter problems

- There is a homogenous resource to be allocated.
- An allocation defines an amount of the resource for each player
  - $\Omega \subseteq \mathbb{R}^n_+$
- A player's value is linear in the amount of resource received
  - Player i's valuation summarized by $v_i \in \mathbb{R}$
  - Value for $\omega \in \Omega$ is $v_i \cdot \omega_i$

Single-parameter Problems

First half of the class will focus on the prior-free model. We begin with

Single-parameter problems

- There is a homogenous resource to be allocated.
- An allocation defines an amount of the resource for each player
  - $\Omega \subseteq \mathbb{R}^n_+$
- A player's value is linear in the amount of resource received
  - Player i's valuation summarized by $v_i \in \mathbb{R}$
  - Value for $\omega \in \Omega$ is $v_i \cdot \omega_i$

Examples

- Single-item allocation
- Knapsack allocation
- Single-minded combinatorial allocation
- Related machine scheduling
- ...

Monotonicity Characterization

Single-parameter problems receive special attention in part because their space of truthful mechanisms is much more permissive.

Monotonicity Characterization

Single-parameter problems receive special attention in part because their space of truthful mechanisms is much more permissive.

Theorem (Myerson '81, Archer/Tardos '01)

An allocation rule $\mathcal{A}$ for a single-parameter problem can be combined with a payment scheme p to give a DSIC mechanism iff $\mathcal{A}$ is monotone.

An allocation rule $\mathcal{A}$ for a single-parameter problem is monotone if increasing v_i, holding v_{-i} fixed, does not decrease $\mathcal{A}_i(v)$ (in expectation).

Example: Allocation rule that gives single item to highest bidder is monotone, combined with the second-price payment scheme, gives Vickrey Auction.

Algorithmic Results for Single-parameter Problems

Due to the "permissiveness" of monotone algorithms:

Algorithmic Results for Single-parameter Problems

Due to the "permissiveness" of monotone algorithms:

For most natural single-parameter problems, DSIC approximation mechanisms matching guarantee of the best approximation algorithm are known:

- Welfare in Knapsack allocation and generalizations [BKV '05]
- Welfare in Single-minded combinatorial auctions [LOS '02]
- Makespan in Related machine scheduling [DDDR '08]
- …

Algorithmic Results for Single-parameter Problems

Due to the “permissiveness” of monotone algorithms:

For most natural single-parameter problems, DSIC approximation mechanisms matching guarantee of the best approximation algorithm are known:

- Welfare in Knapsack allocation and generalizations [BKV ’05]
- Welfare in Single-minded combinatorial auctions [LOS ’02]
- Makespan in Related machine scheduling [DDDR ’08]
- ...

Approximation-preserving black-box reductions from algorithms to truthful mechanisms for classes of single-parameter problems

- Welfare problems with an FPTAS (e.g. Knapsack) [BKV ’05]
- Welfare problems that are “player-symmetric” [HWZ ’11]

Outline

Multi-parameter Problems

Definition

Mechanism design problems that aren't single-parameter…

Player valuations are described by many private parameters

- Combinatorial allocation: value for each bundle
- Assignment Problems: generalizations of knapsack where there are multiple bins, and value of a player depends on bin to which his task is assigned.

Multi-parameter Problems

Definition

Mechanism design problems that aren't single-parameter...

Player valuations are described by many private parameters

- Combinatorial allocation: value for each bundle
- Assignment Problems: generalizations of knapsack where there are multiple bins, and value of a player depends on bin to which his task is assigned.

Often, multi-parameter problems have single-parameter special cases

- Single-minded combinatorial auctions
- Knapsack auction

VCG Mechanism

Vickrey Clarke Groves (VCG) Mechanism

1. Solicit (purported) valuations $b_1, \ldots, b_n$
2. Find allocation $\omega \in \Omega$ maximizing (purported) welfare: $\sum_i b_i(S_i^*)$
3. Charge each player his externality
   - The increase in (purported) welfare of other players if he drops out

VCG Mechanism

Vickrey Clarke Groves (VCG) Mechanism

1. Solicit (purported) valuations $b_1, \ldots, b_n$
2. Find allocation $\omega \in \Omega$ maximizing (purported) welfare: $\sum_i b_i(S_i^*)$
3. Charge each player his externality
   - The increase in (purported) welfare of other players if he drops out

Theorem (Vickrey, Clarke, Groves)

VCG is truthful, and (therefore also) welfare maximizing.

i.e. Reporting $b_i = v_i$ is a dominant strategy.

VCG Mechanism

Vickrey Clarke Groves (VCG) Mechanism

1. Solicit (purported) valuations $b_1, \ldots, b_n$
2. Find allocation $\omega \in \Omega$ maximizing (purported) welfare: $\sum_i b_i(S_i^*)$
3. Charge each player his externality
   - The increase in (purported) welfare of other players if he drops out

Theorem (Vickrey, Clarke, Groves)

VCG is truthful, and (therefore also) welfare maximizing.

i.e. Reporting $b_i = v_i$ is a dominant strategy.

However, requires exact optimization, which is NP-hard for problems we will look at.

Therefore, we will examine the space of truthful mechanisms beyond VCG. . .

Deterministic Truthfulness with Unrestricted Valuations

First, we examine requirements for truthfulness in a very general setting…

Unrestricted Mechanism Design Problem

Each player's valuation is an arbitrary function $v_i : \Omega \rightarrow \mathbb{R}$.

Deterministic Truthfulness with Unrestricted Valuations

First, we examine requirements for truthfulness in a very general setting...

Unrestricted Mechanism Design Problem

Each player's valuation is an arbitrary function $v_i : \Omega \to \mathbb{R}$.

For a deterministic mechanism to be truthful over all valuations, it must be of a specific form

Theorem (Roberts (Informal))

When player valuations are unrestricted, the allocation rule of every deterministic and dominant-strategy truthful mechanism is (essentially) maximal-in-range. Moreover, its payments are (essentially) the externality.

Maximal-in-Range Algorithm

1. Choose a subset $\mathcal{R}$ of all feasible allocations Ω, independent of valuations.
2. Solicit valuations $v_1, \ldots, v_n$
3. Output $\omega \in \mathcal{R}$ maximizing $\sum_i v_i(\omega)$

Maximal-in-Range Algorithm

1. Choose a subset $\mathcal{R}$ of all feasible allocations Ω, independent of valuations.
2. Solicit valuations $v_1, \ldots, v_n$
3. Output $\omega \in \mathcal{R}$ maximizing $\sum_i v_i(\omega)$

Fact

Any maximal-in-range algorithm can be equipped with VCG payments (externality) to yield a truthful mechanism.

Maximal-in-Range Algorithm

1. Choose a subset $\mathcal{R}$ of all feasible allocations Ω, independent of valuations.
2. Solicit valuations $v_1, \ldots, v_n$
3. Output $\omega \in \mathcal{R}$ maximizing $\sum_i v_i(\omega)$

Fact

Any maximal-in-range algorithm can be equipped with VCG payments (externality) to yield a truthful mechanism.

Essentially a converse of Roberts' Theorem

Maximal-in-Range Algorithm

1. Choose a subset $\mathcal{R}$ of all feasible allocations Ω, independent of valuations.
2. Solicit valuations $v_1, \ldots, v_n$
3. Output $\omega \in \mathcal{R}$ maximizing $\sum_i v_i(\omega)$

Fact

Any maximal-in-range algorithm can be equipped with VCG payments (externality) to yield a truthful mechanism.

Essentially a converse of Roberts' Theorem

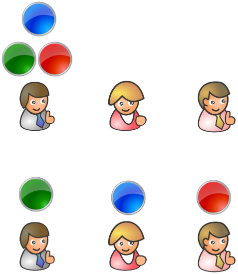
Flexibility in choosing $\mathcal{R}$ allows the design of mechanisms other than VCG that are both polytime and approximately optimal.

Example of MIR

Due to Dobzinski, Nisan, and Schapira ’05.

Range

Allocations that either allocate all items to a single player, or each player at most one item.

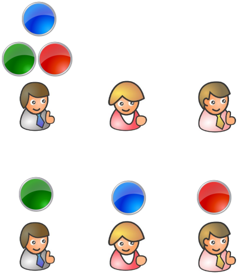


Example of MIR

Due to Dobzinski, Nisan, and Schapira '05.

Range

Allocations that either allocate all items to a single player, or each player at most one item.

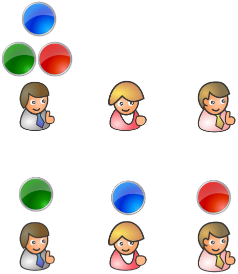


Lemma

When players have complement-free valuations, there is always an allocation in the range guaranteeing a $\sqrt{m}$ *approximation.*

Example of MIR

Due to Dobzinski, Nisan, and Schapira ’05.



Range

Allocations that either allocate all items to a single player, or each player at most one item.

Lemma

When players have complement-free valuations, there is always an allocation in the range guaranteeing a $\sqrt{m}$ approximation.

Lemma

There is a polynomial-time algorithm for optimizing over this range.

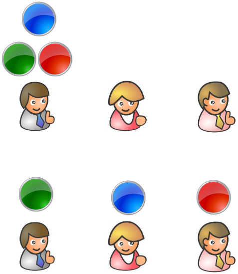
Reduces to maximum matching.

Example of MIR

Due to Dobzinski, Nisan, and Schapira '05.

Range

Allocations that either allocate all items to a single player, or each player at most one item.



Takeaways

- Throwing away "complicated" allocations, ended up with polytime solvable optimization problem without much loss in optimality.
- Plugging in an algorithm for this problem into the allocation step of VCG recovers incentive compatibility.
- Designing truthful mechanisms in this way is akin to working in a restricted computational model

- Roberts' theorem does not formally hold for individual problems, like combinatorial allocation, knapsack allocation, etc.
- Neither is it known to hold if randomization is allowed in the mechanism.
- Nevertheless, a randomized analogue of Roberts' Theorem appears to hold "in spirit".

Trend

Usually, DSIC mechanisms for multi-parameter problems employ maximal-in-distributional-range (MIDR) allocation algorithms and VCG payments.

Techniques for Multi-parameter Problems

Polynomial-time Maximal-in-distributional-range (MIDR) algorithms led to improved mechanisms for many problems

- Combinatorial auctions
- Assignment problems
- Public project problems
- ...

Techniques for Multi-parameter Problems

Polynomial-time Maximal-in-distributional-range (MIDR) algorithms led to improved mechanisms for many problems

- Combinatorial auctions
- Assignment problems
- Public project problems
- ...

These algorithms came as biproducts of new algorithmic techniques, rather than from the direct definition of "range".

1. Lavi/Swamy LP Technique
2. Perturbation-based techniques
3. Rounding-based techniques

Techniques for Multi-parameter Problems

Polynomial-time Maximal-in-distributional-range (MIDR) algorithms led to improved mechanisms for many problems

- Combinatorial auctions
- Assignment problems
- Public project problems
- ...

These algorithms came as biproducts of new algorithmic techniques, rather than from the direct definition of "range".

1. Lavi/Swamy LP Technique
   - The design of linear programs with small integrality gaps
2. Perturbation-based techniques
   - Smoothed Complexity
3. Rounding-based techniques
   - Randomized rounding algorithms for linear programs

These techniques, in turn, built on classical ideas in algorithm design

Techniques for Multi-parameter Problems

Polynomial-time Maximal-in-distributional-range (MIDR) algorithms led to improved mechanisms for many problems

- Combinatorial auctions
- Assignment problems
- Public project problems
- ...

These algorithms came as biproducts of new algorithmic techniques, rather than from the direct definition of “range”.

1. Lavi/Swamy LP Technique
   - The design of linear programs with small integrality gaps
2. Perturbation-based techniques
   - Smoothed Complexity
3. Rounding-based techniques
   - Randomized rounding algorithms for linear programs

These techniques, in turn, built on classical ideas in algorithm design

In much of this portion of the class, we will present these techniques.

Techniques for Multi-parameter Problems

Polynomial-time Maximal-in-distributional-range (MIDR) algorithms led to improved mechanisms for many problems

- Combinatorial auctions
- Assignment problems
- Public project problems
- ...

These algorithms came as biproducts of new algorithmic techniques, rather than from the direct definition of "range".

1. Lavi/Swamy LP Technique
   - The design of linear programs with small integrality gaps
2. Perturbation-based techniques
   - Smoothed Complexity
3. Rounding-based techniques
   - Randomized rounding algorithms for linear programs

These techniques, in turn, built on classical ideas in algorithm design

In much of this portion of the class, we will present these techniques.

Overview

- Considers welfare maximization mechanism design problems in a prior-free setting
- Reduces the design of approximate mechanisms to the design of linear programming relaxations satisfying certain conditions.

Overview

- Considers welfare maximization mechanism design problems in a prior-free setting
- Reduces the design of approximate mechanisms to the design of linear programming relaxations satisfying certain conditions.

Theorem (Lavi and Swamy)

Consider a welfare-maximization problem. If

- *the problem can be written as a packing integer linear program with integrality gap at most α,*

Overview

- Considers welfare maximization mechanism design problems in a prior-free setting
- Reduces the design of approximate mechanisms to the design of linear programming relaxations satisfying certain conditions.

Theorem (Lavi and Swamy)

Consider a welfare-maximization problem. If

- *the problem can be written as a packing integer linear program with integrality gap at most α,*
- *the PILP can be solved in polynomial time,*

Overview

- Considers welfare maximization mechanism design problems in a prior-free setting
- Reduces the design of approximate mechanisms to the design of linear programming relaxations satisfying certain conditions.

Theorem (Lavi and Swamy)

Consider a welfare-maximization problem. If

- *the problem can be written as a packing integer linear program with integrality gap at most α,*
- *the PILP can be solved in polynomial time,*
- *and there is an algorithm that shows integrality gap α,*

Overview

- Considers welfare maximization mechanism design problems in a prior-free setting
- Reduces the design of approximate mechanisms to the design of linear programming relaxations satisfying certain conditions.

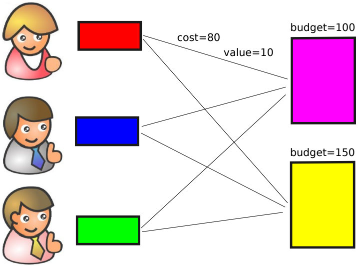
Theorem (Lavi and Swamy)

Consider a welfare-maximization problem. If

- *the problem can be written as a packing integer linear program with integrality gap at most α,*
- *the PILP can be solved in polynomial time,*
- *and there is an algorithm that shows integrality gap α,*

then an α-approximate MIDR algorithm can be generically derived in polynomial time.

Example: Generalized Assignment

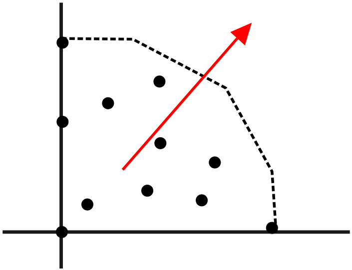


- n self-interested agents, m machines.
- $v_i(j)$ is agent i's value for his task going on machine j. (private)
- $c_i(j)$ is the cost to machine j of agent i's job. (public)
- b_j is machine j's budget. (public)

Goal

Partial assignment of jobs to machines, respecting machine budgets, and maximizing total value of agents.

Packing Linear Integer Programs



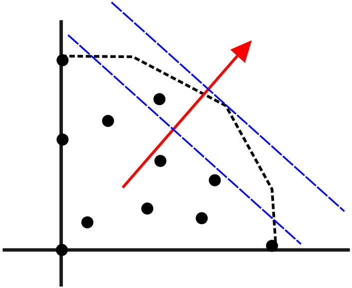
Generic PILP $(A, v \geq 0)$

$$\begin{array}{ll} \text{max} & \sum_i v_i^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \\ & x \in \mathbb{Z}^m \end{array}$$

Example: GAP PILP

$$\begin{array}{lll} \text{max} & \sum_{ij} v_i(j) x_{ij} & \\ \text{s.t.} & \sum_i c_{ij} x_{ij} \leq b_j, & \text{for } j \in [m]. \\ & x_{ij} \geq 0, & \text{for } i \in [n], j \in [m]. \\ & x_{ij} \in \{0, 1\}, & \text{for } i \in [n], j \in [m]. \end{array}$$

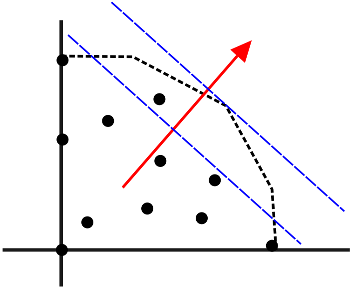
Packing Linear Integer Programs



Definition (Integrality Gap)

A PILP has integrality gap at most α if, for every objective $v \in \mathbb{R}_+^m$, the ratio of the welfare of the best fractional solution and the best integer solution is at most α.

Packing Linear Integer Programs

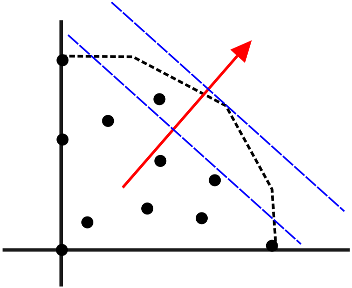


Definition (Integrality Gap)

A PILP has integrality gap at most α if, for every objective $v \in \mathbb{R}_+^m$, the ratio of the welfare of the best fractional solution and the best integer solution is at most α.

Note: must hold for all nonnegative objectives v.

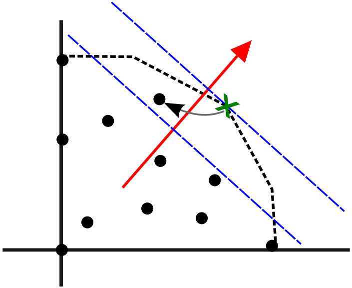
Packing Linear Integer Programs



Definition

An algorithm for a PILP shows an integrality gap of α if, for every objective $v \in \mathbb{R}_+^m$, it always outputs an integer solution with objective value at least $1/\alpha$ of that of the best fractional solution, in expectation.

Packing Linear Integer Programs



Definition

An algorithm for a PILP shows an integrality gap of α if, for every objective $v \in \mathbb{R}_+^m$, it always outputs an integer solution with objective value at least $1/\alpha$ of that of the best fractional solution, in expectation.

Commonly, such an algorithm “rounds” the optimal fractional solution of the LP, but this is not necessary.

Recall: Theorem Statement

Theorem (Lavi and Swamy)

Consider a welfare-maximization problem. If

- *the problem can be written as a packing integer linear program with integrality gap at most α,*
- *the PILP can be solved in polynomial time,*
- *and there is an algorithm that shows integrality gap α,*

then an α-approximate MIDR algorithm can be generically derived in polynomial time.

Recall: Theorem Statement

Theorem (Lavi and Swamy)

Consider a welfare-maximization problem. If

- *the problem can be written as a packing integer linear program with integrality gap at most α,*
- *the PILP can be solved in polynomial time,*
- *and there is an algorithm that shows integrality gap α,*

then an α-approximate MIDR algorithm can be generically derived in polynomial time.

The PILP for gap has integrality gap 2, and there is a rounding algorithm showing it. Therfore, implies a 2-approximate, polynomial-time, DSIC mechanism.

Outline

Second half of the class will focus on Bayesian model

- Assume agent valuations are drawn from a (publicly known) distribution.
- Require incentive-compatibility and good outcomes only in expectation.
- Weaker guarantees depart from the “worst case” paradigm traditional in TCS
- However, can do more...

Black-box Reductions for Welfare

Much of the first half of the class considered the following question in the prior-free setting

Question

When can we convert a “good” polynomial-time algorithm to a truthful polynomial time mechanism without much loss in optimality?

Black-box Reductions for Welfare

Much of the first half of the class considered the following question in the prior-free setting

Question

When can we convert a "good" polynomial-time algorithm to a truthful polynomial time mechanism without much loss in optimality?

The best possible answer ...

Aspirational Answer

For every mechanism design problem and polynomial-time α-approximation algorithm for the problem, a black-box reduction converts the algorithm to a truthful, polynomial-time mechanism with the same approximation ratio.

Black-box Reductions for Welfare

Much of the first half of the class considered the following question in the prior-free setting

Question

When can we convert a “good” polynomial-time algorithm to a truthful polynomial time mechanism without much loss in optimality?

The best possible answer . . .

Aspirational Answer

For every mechanism design problem and polynomial-time α-approximation algorithm for the problem, a black-box reduction converts the algorithm to a truthful, polynomial-time mechanism with the same approximation ratio.

Unfortunately, this is false in prior free settings, even for some concrete welfare-maximization problems.

Theorem (HL '10)

For any single-parameter problem in a Bayesian setting and α-approximation algorithm for that problem, a black box reduction converts the algorithm in polynomial-time to an α-approximate BIC mechanism.

Theorem (HKM / BH '11)

For a multi-parameter problem in a Bayesian setting with small support, and α-approximation algorithm for that problem, a black box reduction converts the algorithm in polynomial-time to an α-approximate BIC mechanism.

Revenue-Optimal Mechanisms

In prior-free settings, we mostly ignored revenue

- No unequivocal benchmark
- Every auction will produce very small revenue on SOME worst case valuation profile
- Even single-item, single-bidder...

Revenue-Optimal Mechanisms

In prior-free settings, we mostly ignored revenue

- No unequivocal benchmark
- Every auction will produce very small revenue on SOME worst case valuation profile
- Even single-item, single-bidder...

In Bayesian settings, we can formulate reasonable benchmarks and get interesting results

Benchmark

The maximum expected revenue of a BIC mechanism, where expectation is over valuations.

Classics: Myerson's Optimal Auction

Theorem (Myerson '81)

Consider a bayesian single-item allocation setting, where player values are drawn i.i.d from some distribution D. The revenue-optimal BIC mechanism is Vickrey with reserve $r = r(D)$.

Vickrey Auction with Reserve

1. Let $r = r(D)$
2. Collect bids
3. If nobody bids above reserve, then cancel the auction, otherwise
4. Give to highest bidder
5. Charge the second highest bid or r, whichever is bigger

Classics: Myerson's Lemma

Recall: Single-parameter problems

- There is a homogenous resource to be allocated.
- An allocation defines an amount of the resource for each player
  - $\Omega \subseteq \mathbb{R}^n_+$
- A player's value is linear in the amount of resource received
  - Player i's valuation summarized by $v_i \in \mathbb{R}$
  - Value for $\omega \in \Omega$ is $v_i \cdot \omega_i$

Classics: Myerson's Lemma

Recall: Single-parameter problems

- There is a homogenous resource to be allocated.
- An allocation defines an amount of the resource for each player
  - $\Omega \subseteq \mathbb{R}_+^n$
- A player's value is linear in the amount of resource received
  - Player i's valuation summarized by $v_i \in \mathbb{R}$
  - Value for $\omega \in \Omega$ is $v_i \cdot \omega_i$

Myerson's Lemma

Consider a bayesian single-parameter setting where v_i are independently drawn from distributions D_i. The revenue-optimal BIC mechanism is the welfare-optimal BIC mechanism for "virtual" valuations $\phi_i(v_i)$.

Upshot: in single-parameter settings, revenue maximization reduces to welfare maximization, which we know how to do using VCG in many contexts.

Recent: Revenue-optimal Mechanisms in Multi-paramter Bayesian Settings

Very recently, there has been work extending Myerson's results to some multi-parameter settings

- Multi-item auctions with additive valuations
- "Single-service" settings

Recent: Revenue-optimal Mechanisms in Multi-paramter Bayesian Settings

Very recently, there has been work extending Myerson's results to some multi-parameter settings

- Multi-item auctions with additive valuations
- "Single-service" settings

We will spend some time trying to understand these very exciting new developments, and examining research directions thereof.

Outline

Student Presentations

- Research papers and ideas for projects will be posted on the course webpage.
- Students will study a research direction (2-4 papers) after discussing with instructor.
- Goal: Presentation to the class, and a summary report.
  - Best case scenario: original research!
- You can pair up, but standards will be raised (prove new stuff!)

Thank You for Listening