Intractability in Algorithmic Game Theory

Tim Roughgarden

Stanford University

How Theory CS Can Contribute

Unsurprising fact: very rich tradition and literature on mechanism design and equilibria in economics.

- largely "Bayesian" (i.e., average-case) settings
- emphasizes exact solutions/characterizations
- usually ignores communication/computation

What we have to offer:

1. worst-case guarantees
2. approximation bounds
3. computational complexity

Overview

Part I: Algorithmic Mechanism Design

- *goal: design polynomial-time protocols so that self-interested behavior leads to socially desirable outcome*
- *intractability from joint computational, incentive constraints*

Part II: Revenue-Maximizing Auctions

- Information-theoretic intractability
- Interpolation of worst-case, average-case analysis

Part III: Complexity of Computing Equilibria

- Computing Nash equilibria is PPAD-complete
- Interpretations and open questions

References

- FOCS 2010 tutorial, “How to Think About Algorithmic Mechanism Design”
  - video available from my home page

- CACM July 2010 survey article, “Algorithmic Game Theory”.

review articles

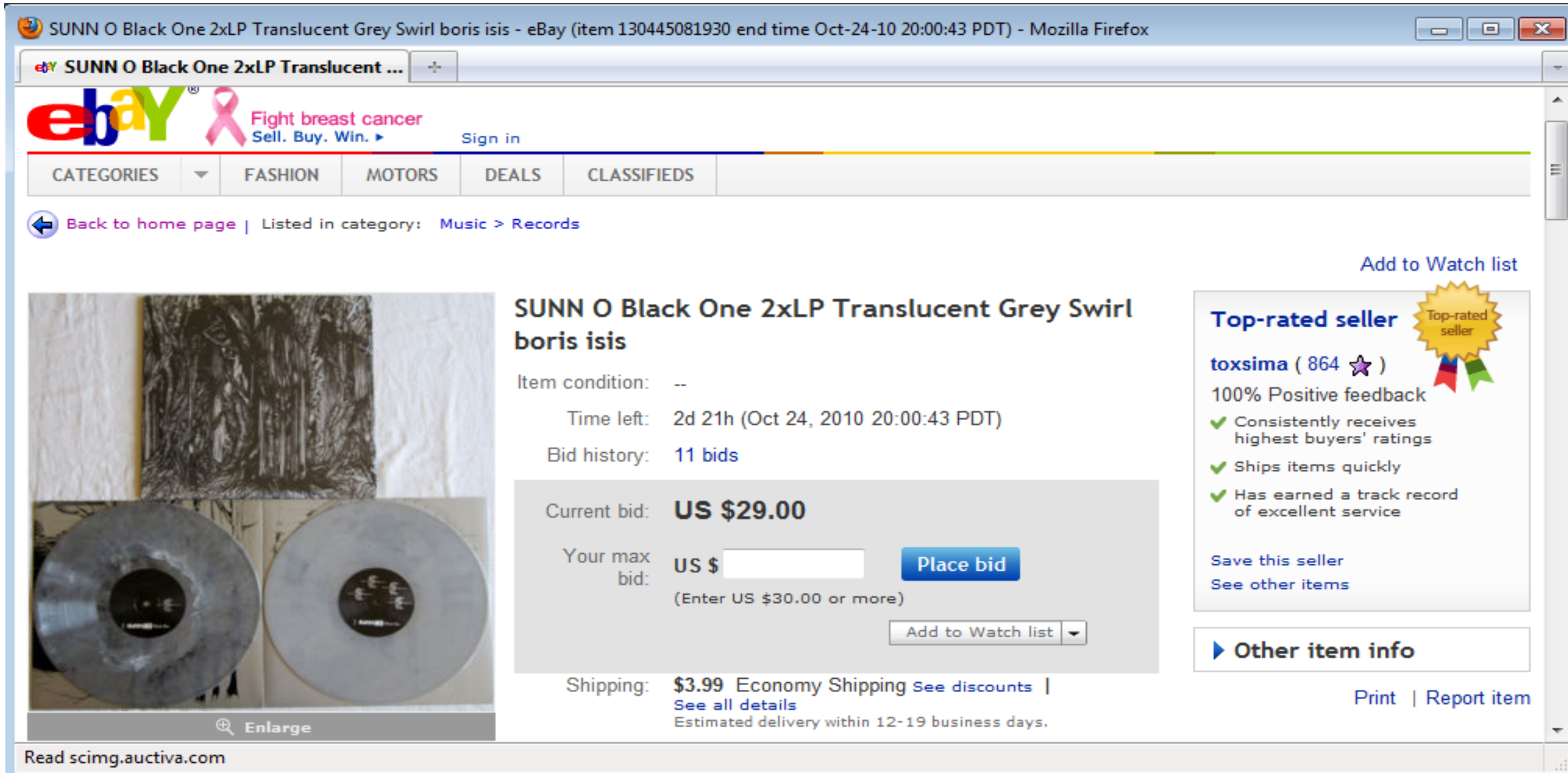
DOI:10.1145/1785414.1785439

A new era of theoretical computer science addresses fundamental problems about auctions, networks, and human behavior.

BY TIM ROUGHGARDEN

Algorithmic Game Theory

An eBay Single-Good Auction



- winner = highest bidder above reserve price
- price = max{second-highest bid, reserve}

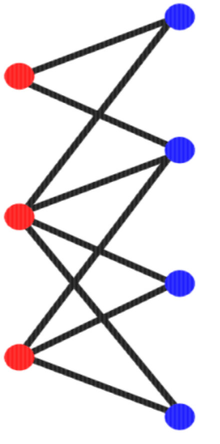
Truthful Auctions

Claim: a second-price auction (like eBay) is *truthful*

- bidding your maximum willingness to pay (your "value") is "foolproof" (a *dominant strategy*)
- i.e., your utility (value – price) is at least as large with a truthful bid as with any other bid

Proof idea: truthful bid equips the auctioneer (who knows all the bids) to bid optimally on your behalf.

A More Complex Example



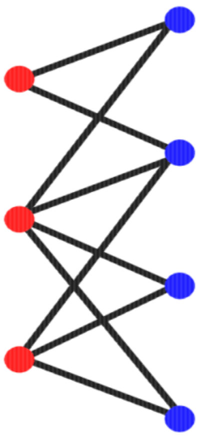
Setup: n bidders, m houses.

- each bidder i has private value v_{ij} for house j, wants ≤ 1 house.

Analog of 2^{nd}-price auction:

- every bidder submits bid for every house

A More Complex Example



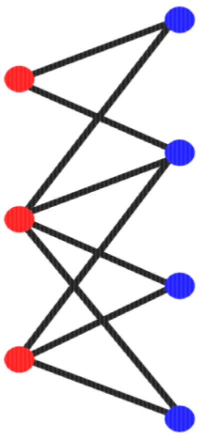
Setup: n bidders, m houses.

- each bidder i has private value v_{ij} for house j, wants ≤ 1 house.

Analog of 2^{nd}-price auction:

- every bidder submits bid for every house
- compute a max-value matching

A More Complex Example



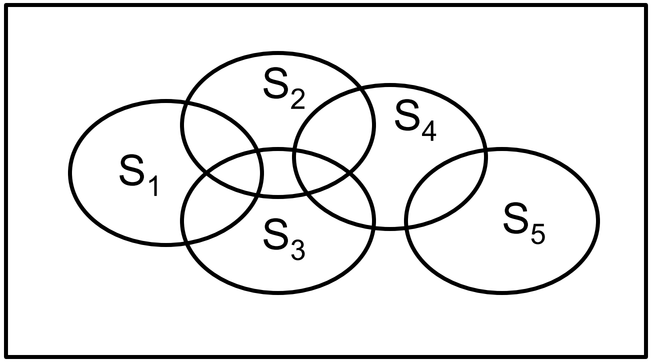
Setup: n bidders, m houses.

- each bidder i has private value v_{ij} for house j, wants ≤ 1 house.

Analog of 2^{nd}-price auction:

- every bidder submits bid for every house
- compute a max-value matching
- charge suitable payments so that bidding true values is dominant strategy for every bidder (relatively simple calculation: this can be done)

Another More Complex Example



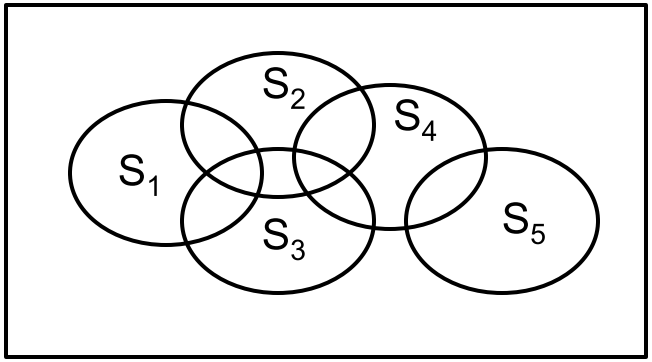
Setup: n bidders, m goods.

- bidder i has private value v_i for known subset S_i of goods

Analog of 2nd-price auction:

- every bidder submits a bid for the bundle it wants
- compute a max-value packing
- charge suitable payments so that bidding true values is dominant strategy for every bidder (relatively simple calculation: this can be done)

Another More Complex Example



Setup: n bidders, m goods.

- bidder i has private value v_i for known subset S_i of goods

Analog of 2nd-price auction:

- every bidder submits a bid for the bundle it wants
- compute a max-value packing ← NP-hard!
- charge suitable payments so that bidding true values is dominant strategy for every bidder (relatively simple calculation: this can be done)

The Research Agenda

[Nisan/Ronen 99] For as many optimization problems as possible, design a mechanism:

- runs in polynomial time
- every player has a dominant strategy
- dominant strategies yield near-optimal outcome

Examples: Can maximize welfare (sum of values) exactly in single-item and matching problems.

- special cases of the "VCG mechanism"

The Research Agenda

goal: runs in poly-time, dominant strategies yield near-optimal outcome

Holy grail: Match approximation factor of best-known poly-time approximation algorithm.

The Research Agenda

goal: runs in poly-time, dominant strategies yield near-optimal outcome

Holy grail: Match approximation factor of best-known poly-time approximation algorithm.

Obvious approach:

1. every bidder submits bids
2. compute an approximately max-value solution using best-known approximation algorithm
3. charge suitable payments so that bidding true values is dominant strategy for every bidder

The Research Agenda

goal: runs in poly-time, dominant strategies yield near-optimal outcome

holy grail: match approximation factor of best approximation algorithm.

Problem [Nisan/Ronen 00] for all but a special type of approximation algorithms, *no payments make truthful bidding a dominant strategy.*

How to think about algorithmic mechanism design: designing truthful mechanisms boils down to designing algorithms in a certain "restricted computational model".

The Punch Line

Theorem(s): Joint intractability of computational and game-theoretic constraints often much more severe than that of either constraint by itself.

- 1st compelling example: [Papadimitriou/Schapira/Singer FOCS 08]
  - O(1)-approximation with only poly-time or GT constraints, Ω(poly(n))-approximation with both
  - blends Robert's theorem, probabilistic method, Sauer-Shelah Lemma, and a clever embedding of 3SAT
- state of the art (2011-12): Dobzinski, Dughmi, Vondrak

Overview

Part I: Algorithmic Mechanism Design

- goal: design polynomial-time protocols so that self-interested behavior leads to socially desirable outcome
- intractability from joint computational, incentive constraints

Part II: Revenue-Maximizing Auctions

- *Information-theoretic intractability*
- *Interpolation of worst-case, average-case analysis*

Part III: Complexity of Computing Equilibria

- Computing Nash equilibria is PPAD-complete
- Interpretations and open questions

Reference

- Jason Hartline, “Approximation in Economic Design”, book in preparation, 2013.

Welfare vs. Revenue

Question: why should revenue maximization be different than welfare-maximization?

Answer: welfare defined *extrinsic* to the auction, payments generated by auction itself.

- in 2nd-price auction, payments only means to an end

- clear what "maximum-possible welfare" means
- not clear what "max-possible revenue" means

Example: Multi-Unit Auctions

Setup: n bidders, k identical goods.

- bidder i has private "valuation" v_i for a good
- v_i = maximum willingness to pay

Design space: decide on:

- (1) at most k winners; and (2) selling prices.

Example: Vickrey auction.

- top k bidders win; all pay (k+1)th highest bid

Variant: Vickrey with a *reserve.* [≈extra bid by seller]

Auction Benchmarks

Goal: design an auction A for which:

"Theorem: for every valuation profile v:
auction A's revenue on v is at least OPT(v)/α."

(for a hopefully small constant α)

Auction Benchmarks

Goal: design an auction A for which:

"Theorem: for every valuation profile v:
auction A's revenue on v is at least OPT(v)/α."

(for a hopefully small constant α)

Idea for OPT(v): sum of k largest v_i's.

Auction Benchmarks

Goal: design an auction A for which:

"Theorem: for every valuation profile v:
auction A's revenue on v is at least OPT(v)/α."
(for a hopefully small constant α)

Idea for OPT(v): sum of k largest v_i's.

Problem: too strong, not useful.

- all auctions A are terrible [no constant α is possible]
- "optimal" auction for this benchmark is uninteresting

Classic Optimal Auctions

Example: 1 bidder, 1 item, v ~ known distribution F

- want to choose optimal reserve price p
- expected revenue of p: $p(1-F(p))$
  - given F, can solve for optimal p^*
  - e.g., $p^* = ½$ for v ~ uniform[0,1]
- but: what about k,n >1 (with i.i.d. v_i's)?

Classic Optimal Auctions

Example: 1 bidder, 1 item, v ~ known distribution F

- want to choose optimal reserve price p
- expected revenue of p: $p(1-F(p))$
  - given F, can solve for optimal p^*
  - e.g., $p^* = ½$ for v ~ uniform[0,1]
- but: what about k,n >1 (with i.i.d. v_i's)?

need minor "regularity" condition on F

Theorem: [Myerson 81] auction with max expected revenue is second-price with above reserve p^*.

- note p^* is *independent of k and n*

Prior-Independent Auctions

New goal: [Dhangwatnotai/Roughgarden/Yan EC 10]
prove results of the form:

"*Theorem:* *for every* distribution *D:*

$E_{v\sim D}[rev(A(v))] \geq (E_{v\sim D}[rev(OPT_D(v))])/\alpha$"

(for a hopefully small constant α)

Prior-Independent Auctions

New goal: [Dhangwatnotai/Roughgarden/Yan EC 10]
prove results of the form:

"Theorem: for every distribution *D (in some set C):*

$E_{v\sim D}[rev(A(v))] \geq (E_{v\sim D}[rev(OPT_D(v))])/\alpha$"

(for a hopefully small constant α)

Interpretation of C: provides knob to tune "optimality vs. robustness" trade-off.

- a single auction that is simultaneously near-optimal across many average-case settings.

Bulow-Klemperer Theorem

Setup: single-item auction. Let D be a valuation distribution. [Needs to be "regular".]

Theorem: [Bulow-Klemperer 96]: for every $n \geq 1$:

E[2^{nd}-price revenue] E[OPT_D's revenue]

Bulow-Klemperer Theorem

Setup: single-item auction. Let D be a valuation distribution. [Needs to be "regular".]

Theorem: [Bulow-Klemperer 96]: for every n≥1:

E[2^{nd}-price revenue] ≥ E[OPT_D's revenue]

[with (n+1) i.i.d. bidders] [with n i.i.d. bidders]

Bulow-Klemperer Theorem

Setup: single-item auction. Let D be a valuation distribution. [Needs to be "regular".]

Theorem: [Bulow-Klemperer 96]: for every $n \geq 1$:

E[2^{nd}-price revenue] $\geq$ E[OPT_D's revenue]

[with (n+1) i.i.d. bidders] [with n i.i.d. bidders]

Usual interpretation: small increase in competition more important than running optimal auction.

Also: 2^{nd}-price $\approx$ a good prior-independent auction!

Overview

Part I: Algorithmic Mechanism Design

- goal: design polynomial-time protocols so that self-interested behavior leads to socially desirable outcome
- intractability from joint computational, incentive constraints

Part II: Revenue-Maximizing Auctions

- Information-theoretic intractability
- Interpolation of worst-case, average-case analysis

Part III: Complexity of Computing Equilibria

- *Computing Nash equilibria is PPAD-complete*
- *Interpretations and open questions*

Reference

- T. Roughgarden, Computing Equilibria: A Computational Complexity Perspective, survey for Economic Theory, 2010.

Example: Prisoner's Dilemma

	cooperate	defect
cooperate	5, 5	0, 10
defect	10, 0	1, 1

Example: Prisoner's Dilemma

	cooperate	defect
cooperate	5, 5	0, 10
defect	10, 0	1, 1

the (unique) Nash equilibrium

Example: Penalty Kick Game

	dive left	dive right
kick left	0, 1	1, 0
kick right	1, 0	0, 1

Example: Penalty Kick Game

	dive left	dive right
kick left	0, 1	1, 0
kick right	1, 0	0, 1

(unique) Nash equilibrium: kicker and goalie each pick a strategy uniformly at random

The 2-Nash Problem

Input: a bimatrix game (one pair of integer payoffs per entry of an m×n matrix).

existence guaranteed by [Nash 51] or by Lemke-Howson algorithm

Output: a Nash equilibrium. (any one will do)

The 2-Nash Problem

Input: a bimatrix game (one pair of integer payoffs per entry of an m×n matrix).

existence guaranteed by [Nash 51] or by Lemke-Howson algorithm

Output: a Nash equilibrium. (any one will do)

Fact: no polynomial-time algorithm is known.

Question: How to amass evidence of intractability?

The 2-Nash Problem

Input: a bimatrix game (one pair of integer payoffs per entry of an m×n matrix).

existence guaranteed by [Nash 51] or by Lemke-Howson algorithm

Output: a Nash equilibrium. (any one will do)

Fact: no polynomial-time algorithm is known.

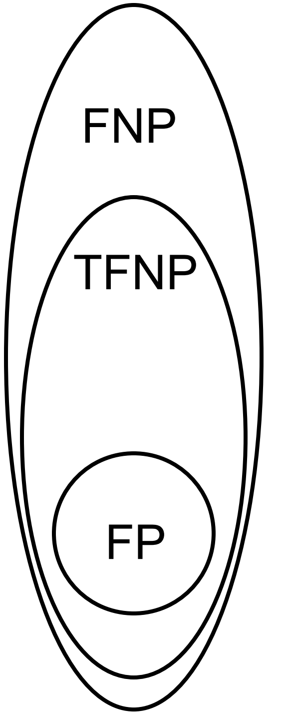
Question: How to amass evidence of intractability?

NP-hard?: Not unless NP=coNP.

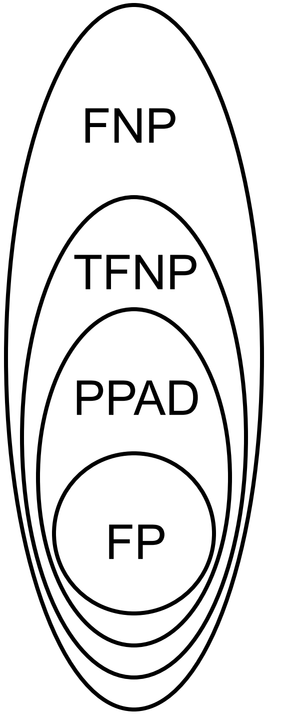
- decision version is trivial, only search version is hard

PPAD

Question: if 2-Nash were complete for some complexity class, what class would it be?



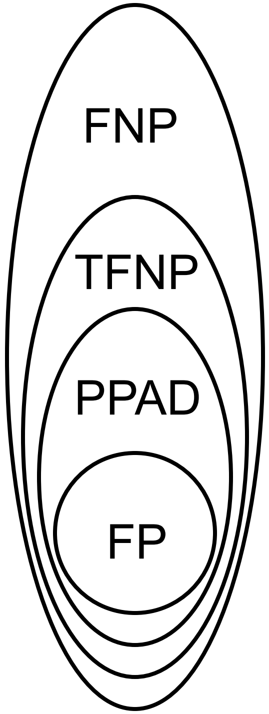
PPAD



Question: if 2-Nash were complete for some complexity class, what class would it be?

Answer: PPAD [Papadimitriou 90]

PPAD

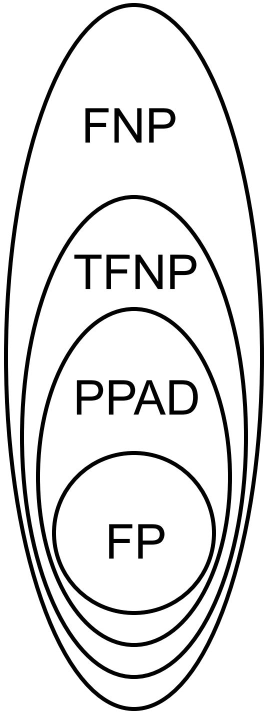


Question: if 2-Nash were complete for some complexity class, what class would it be?

Answer: PPAD [Papadimitriou 90]

Theorem: [Chen/Deng/Teng 06], extending [Daskalakis/Goldberg/Papadimitriou 05] 2-Nash is PPAD-complete.

PPAD



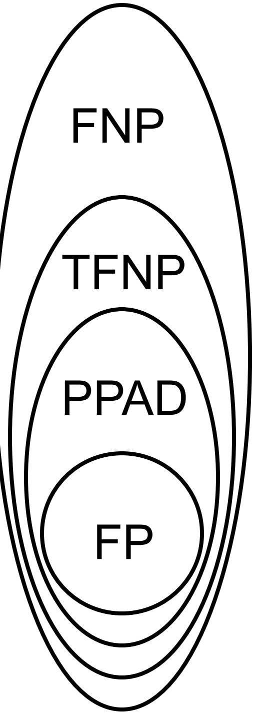
Question: if 2-Nash were complete for some complexity class, what class would it be?

Answer: PPAD [Papadimitriou 90]

Theorem: [Chen/Deng/Teng 06], extending [Daskalakis/Goldberg/Papadimitriou 05] 2-Nash is PPAD-complete.

Question: That's a great result. But how hard is PPAD, really?

Is PPAD Intractable?

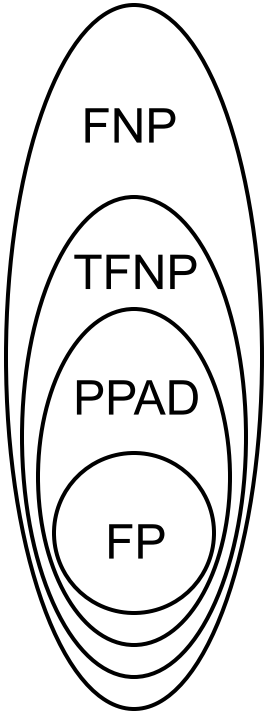


Evidence of intractability (FP≠PPAD):

- several very smart people have worked on a few complete problems
- oracle separation [Beame/Cook/Edmonds/Impagliazzo/Pitassi 98], black-box lower bounds for fixed points [Hirsch/Papadimitriou/Vavasis 89]

But: no known "complexity earthquakes" if FP=PPAD ("just" a bunch of new poly-time algorithms for several tough problems).

Is PPAD Intractable?



Possible research directions:

- find a subexponential-time algorithm for the 2-Nash problem
- relate PPAD-hardness to better-understood hardness notions (existence of one-way functions?)
- relate PPAD (or PLS, etc.) to BQP
- potentially easier for positive results: approximate Nash equilibria (solvable in $n^{O(\log n)}$ time [Lipton/Markakis/Mehta 03])
- convincing implications for economic analysis?

Conclusions

Algorithmic Mechanism Design

- intractability from joint computational, incentive constraints
- new direction: "Bayes-Nash" implementations

Revenue-Maximizing Auctions

- Interpolation of worst-case, average-case analysis
- When are good prior-independent auctions possible?

Complexity of Computing Equilibria

- How hard is PPAD?
- Stronger connections to economic analysis?