Online Mechanism Design

M.Sc. Thesis

Yiannis Giannakopoulos
users.uoa.gr/~ygiannak

University of Athens
Graduate Program in Logic, Algorithms and Computation (MPLA)

Wednesday 17th September, 2008

μΠλ∀

Motivation

- Online Auctions:
  - Bidders arrive *dynamically* over time
  - Auctioneer has no knowledge of future events
  - His decisions can only be based on past events

μΠλ∀

Motivation

- Online Auctions:
  - Bidders arrive *dynamically* over time
  - Auctioneer has no knowledge of future events
  - His decisions can only be based on past events
- Examples:
  - eBay auctions
  - selling airplane seats or theater tickets
  - scheduling computer jobs on a shared server
  - Allocating and pricing network resources (e.g. access to a WiFi network)

μΠλ∀

Motivation

- Online Auctions:
  - Bidders arrive *dynamically* over time
  - Auctioneer has no knowledge of future events
  - His decisions can only be based on past events
- Examples:
  - eBay auctions
  - selling airplane seats or theater tickets
  - scheduling computer jobs on a shared server
  - Allocating and pricing network resources (e.g. access to a WiFi network)
- "Decision Making under Uncertainty" problems
- We choose the framework of Game Theory to study it because we assume our agents (bidders) are
  - Selfish → they only care about their own "happiness"
  - Rational → they always do the right choices to maximize this happiness

μΠλ∀

Outline

1. Online Problems

μΠλΑ

Outline

μΠλ∀

Outline

Outline

μΠλ∀

Presentation Progress

Optimization Problems

- Computer Science → optimization problems
  - goal: maximize or minimize an objective function
  - Examples: bin packing, shortest paths, ...
  - Running time (minimization)

μΠλ∀

Optimization Problems

- Computer Science → optimization problems
  - goal: maximize or minimize an objective function
  - Examples: bin packing, shortest paths, ...
  - Running time (minimization)
- Approximation Algorithms:
  - Optimal solutions can be determined mathematically but many times are "hard" to compute (NP-hard)
  - What if we could easily compute an *almost* optimal solution?
  - Algorithm A is c-approximate if it performs within a factor of c with respect to the optimal ($c > 1$)
  - Performance benchmark: optimal algorithm

Online (Optimization) Problems

- “Traditional” (offline) Algorithmic Design: the input is known *in advance* and *in full*

μΠλ∀

Online (Optimization) Problems

- "Traditional" (offline) Algorithmic Design: the input is known *in advance* and *in full*
- Online Algorithms: the input is revealed gradually over time. Thus, decisions must be based *only* on information already received in the past.
  - many interesting problems are online by their nature: paging, load balancing, k-server, etc.
  - online versions of offline problems, e.g. bin packing.

Online (Optimization) Problems

- "Traditional" (offline) Algorithmic Design: the input is known *in advance* and *in full*
- Online Algorithms: the input is revealed gradually over time. Thus, decisions must be based *only* on information already received in the past.
  - many interesting problems are online by their nature: paging, load balancing, k-server, etc.
  - online versions of offline problems, e.g. bin packing.
- *Uncertainty* about future → things can go really bad...
- What benchmark shall we use to measure performance?

Competitive Analysis

- Notation:
  - online optimization (maximization, w.l.o.g.) problem $\mathcal{P}$
  - set of possible instances (inputs) X
  - for each $x \in X$ a set $F(x)$ of *feasible* solutions (outcomes)
  - Objective function $\sigma(x, y) \in \mathbb{R}_{\geq 0}$, $x \in X, y \in F(x)$.
- Every algorithm A for $\mathcal{P}$ computes a solution $A(x) \in F(x)$ given input x and achieves $\sigma(x, A(x))$.

μΠλ∀

Competitive Analysis

- Notation:
  - online optimization (maximization, w.l.o.g.) problem $\mathcal{P}$
  - set of possible instances (inputs) X
  - for each $x \in X$ a set $F(x)$ of *feasible* solutions (outcomes)
  - Objective function $\sigma(x, y) \in \mathbb{R}_{\geq 0}$, $x \in X, y \in F(x)$.
- Every algorithm A for $\mathcal{P}$ computes a solution $A(x) \in F(x)$ given input x and achieves $\sigma(x, A(x))$.
- Benchmark: an (unrealizable) optimal *offline* algorithm that has full knowledge of the future (and x).

Competitive Analysis (cont.)

- We will say that online algorithm A is c-competitive, if

$$\frac{\sigma(x, \mathrm{OPT}(x))}{\sigma(x, A(x))} \leq c \quad \text{for all} \;\; x \in X.$$

- $c > 1$ (for minimization problems, invert fraction)

μΠλ∀

Competitive Analysis (cont.)

- We will say that online algorithm A is c-competitive, if

$$\frac{\sigma(x, \mathrm{OPT}(x))}{\sigma(x, A(x))} \leq c \quad \text{for all } \ x \in X.$$

- $c > 1$ (for minimization problems, invert fraction)
- What is the best (minimum) c we can achieve? (best performance guarantee)

Definition

Let A be an online algorithm for some (online) maximization problem P with objective function σ. The *competitive ratio* of A is

$$\mathrm{CR}_P(A) = \sup_{x \in X} \frac{\sigma(x, \mathrm{OPT}(x))}{\sigma(x, A(x))}.$$

Competitive Analysis (cont.)

- $\sup_{x \in X} \frac{\mathrm{OPT}(x)}{A(x)} \rightarrow$ "competing" against a malicious, almighty *adversary* who tries to destroy us...

μΠλ∀

Competitive Analysis (cont.)

- $\sup_{x \in X} \frac{\mathrm{OPT}(x)}{A(x)} \rightarrow$ "competing" against a malicious, almighty *adversary* who tries to destroy us...
- Competitive ratio of a *problem*:

$$\mathrm{CR}_P = \inf_A \sup_{x \in X} \frac{\sigma(x, \mathrm{OPT}(x))}{\sigma(x, A(x))}$$

  We first design the best algorithm A we can (without knowing x) and then the adversary chooses the worst input x.
- Worst-case analysis framework (CS) vs. average-case (Bayesian, distribution assumptions) analysis (Economics)

Presentation Progress

μΠλ∀

Games

- "game": players, strategies, payoffs (rewards)

Games

- "game": players, strategies, payoffs (rewards)
- The Prisoner's Dilemma

		Prisoner 2	
		confess	*silent*
Prisoner 1	*confess*	−5, −5	−1, −10
	silent	−10, −1	−2, −2

μΠλ∀

Games

- "game": players, strategies, payoffs (rewards)
- The Prisoner's Dilemma

		Prisoner 2	
		confess	*silent*
Prisoner 1	*confess*	$-5, -5$	$-1, -10$
	silent	$-10, -1$	$-2, -2$

- "rational" game solution (outcome)?

µΠλ∀

Games

- "game": players, strategies, payoffs (rewards)
- The Prisoner's Dilemma

		Prisoner 2	
		confess	*silent*
Prisoner 1	*confess*	$-5, -5$	$-1, -10$
	silent	$-10, -1$	$-2, -2$

- "rational" game solution (outcome)?
  - Stable equilibrium

μΠλ∀

Dominant Strategy Equilibria

- Game (strategic, in normal form):
  - a finite set of *players* $\mathcal{N} = \{1, 2, \ldots, N\}$
  - for every player $i \in \mathcal{N}$, a set of strategies S_i and
  - for every player i, a utility function $u_i : S \longrightarrow \mathbb{R}$ where, $S = \prod_{i=1}^{N} S_i$ is the set of all possible *strategy profiles*.

μΠλ∀

Dominant Strategy Equilibria

- Game (strategic, in normal form):
  - a finite set of *players* $\mathcal{N} = \{1, 2, \ldots, N\}$
  - for every player $i \in \mathcal{N}$, a set of strategies S_i and
  - for every player i, a utility function $u_i : S \longrightarrow \mathbb{R}$ where, $S = \prod_{i=1}^{N} S_i$ is the set of all possible *strategy profiles*.

Definition

A strategy profile $(s_1^*, s_2^*, \ldots, s_N^*)$ is a dominant strategy equilibrium for our game $\mathcal{G}$ if, for every $i \in \mathcal{N}$, s_i^* is a dominant strategy for player i, i.e.

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \qquad \text{for all } \; s_i \in S_i, \; s_{-i} \in S_{-i}.$$

Whatever s_{-i} the other players choose to play, s_i^* is the best choice for player i: very strong solution concept (compare Nash or pure Nash equilibria).

μΠλ∀

Mechanism Design

- Games are extremely useful "gadgets" to model/predict *strategic* behaviour, but "passive". We cannot interfere.
- Players have full information of the games elements (other players' utilities)

Mechanism Design

- Games are extremely useful "gadgets" to model/predict *strategic* behaviour, but "passive". We cannot interfere.
- Players have full information of the games elements (other players' utilities)
- We need to express much more complex environments, set rules, make decisions.
- Model uncertainty

Direct-revelation Mechanisms

- We have a set of possible decisions (outcomes) $\mathcal{O}$
- Each player i has a piece of *private* information θ_i, her *type*.

μΠλ∀

Direct-revelation Mechanisms

- We have a set of possible decisions (outcomes) $\mathcal{O}$
- Each player i has a piece of *private* information θ_i, her *type*.
- A *direct-revelation mechanism* asks all players to report their types and, based on the type profile $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$
  - makes a *decision* $x(\theta) \in \mathcal{O}$
  - collects *payment* $p_i(\theta) \in \mathbb{R}$ from every player i

μΠλ∀

Direct-revelation Mechanisms

- We have a set of possible decisions (outcomes) $\mathcal{O}$
- Each player i has a piece of *private* information θ_i, her *type*.
- A *direct-revelation mechanism* asks all players to report their types and, based on the type profile $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$
    - makes a *decision* $x(\theta) \in \mathcal{O}$
    - collects *payment* $p_i(\theta) \in \mathbb{R}$ from every player i
- Every player i has a value $v_i(\theta_i, o)$ for every outcome $o \in \mathcal{O}$ and receives utility $u_i = v_i - p_i$,

$$u_i(\theta_i, x(\theta)) = v_i(\theta_i, x(\theta)) - p_i(\theta)$$

μΠλ∀

Truthfulness

- Why payments? Give (negative) incentives to players to report θ_i truthfully. We need it to make the "right" decisions.
- Players are strategic and will lie (without regret) by reporting some other type $\widehat{\theta}_i$ if this gets her a better utility

$$u_i(\theta_i, x(\widehat{\theta}_i, \theta_{-i})),$$

where $(\widehat{\theta}_i, \theta_{-i}) = (\theta_1, \ldots, \theta_{i-1}, \widehat{\theta}_i, \theta_{i+1}, \ldots, \theta_N)$.

μΠλ∀

Truthfulness

- Why payments? Give (negative) incentives to players to report θ_i truthfully. We need it to make the "right" decisions.
- Players are strategic and will lie (without regret) by reporting some other type $\widehat{\theta}_i$ if this gets her a better utility

$$u_i(\theta_i, x(\widehat{\theta}_i, \theta_{-i})),$$

where $(\widehat{\theta}_i, \theta_{-i}) = (\theta_1, \ldots, \theta_{i-1}, \widehat{\theta}_i, \theta_{i+1}, \ldots, \theta_N)$.

Definition (Truthfulness)

A (direct-revelation) mechanism $\mathcal{M} = (x, p)$ is called *truthful* (or DSIC) if, for all possible type profiles $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$, every player's i best strategy is to report her type truthfully, i.e.

$$u_i(\theta_i, x(\theta_i, \theta'_{-i})) \geq u_i(\theta_i, x(\widehat{\theta}_i, \theta'_{-i})),$$

for all θ'_{-i}, $\widehat{\theta}_i$.

Vickrey Auction

- Consider a single-item, sealed-bid auction. Let $p > q$ be the two highest submitted bids.
- Obviously, we want to give the item to the bidder that desires it the most (i.e. bids p).
- But how much is she going to pay us?

μΠλ∀

Vickrey Auction

- Consider a single-item, sealed-bid auction. Let $p > q$ be the two highest submitted bids.
- Obviously, we want to give the item to the bidder that desires it the most (i.e. bids p).
- But how much is she going to pay us?
- First-price auctions: The highest bidding player gets allocated and pays p. Utility= p-p=0.
  - NOT truthful! She could have (mis)reported some p' with $p > p' > q$ for a higher utility of $p - p' > 0$.

Vickrey Auction

- Consider a single-item, sealed-bid auction. Let $p > q$ be the two highest submitted bids.
- Obviously, we want to give the item to the bidder that desires it the most (i.e. bids p).
- But how much is she going to pay us?
- First-price auctions: The highest bidding player gets allocated and pays p. Utility= p-p=0.
  - NOT truthful! She could have (mis)reported some p' with $p > p' > q$ for a higher utility of $p - p' > 0$.
- Vickrey (second-price) auction: Again, allocate the highest bidder, but for a payment of q. Utility $p - q >> 0$
  - Truthful! If she reports $p' > q$ she still gets the item for the same utility $p - q$ and if she reports $p' < q$ she loses the item.

W. Vickrey [1961]

μΠλ∀

Efficiency

- *Efficiency* → combined "social welfare" (satisfaction).

$$E(\theta) = \sum_{i=1}^{N} v_i(\theta_i, x(\theta))$$

- Revenue: $R(\theta) = \sum_{i=1}^{N} p_i(x(\theta))$.

μΠλ∀

Efficiency

- *Efficiency* → combined "social welfare" (satisfaction).

$$E(\theta) = \sum_{i=1}^{N} v_i(\theta_i, x(\theta))$$

- Revenue: $R(\theta) = \sum_{i=1}^{N} p_i(x(\theta))$.
- The "big" ideas behind the Vickrey auction:
  - every player's payment is *independent* of her report θ_i.
  - decision $x(\theta)$ is the best possible outcome for every player.
- We want to design truthful mechanisms for general settings that also maximize efficiency. Based on the Vickrey ideas:

μΠλ∀

VCG mechanisms

- VCG mechanism (x, p):
  - x maximizes efficiency

μΠλ∀

VCG mechanisms

- VCG mechanism (x, p):
  - x maximizes efficiency
  - payments:

$$p_i(\theta) = \left[\max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o)\right] - \sum_{j \neq i} v_j(\theta_j, x(\theta)).$$

Each player submits a payment equal to the damage that her presence causes to the other players. "Internalize the externalities". W. Vickrey[1961], E. H. Clark [1971], T. Groves[1973]

μΠλ∀

VCG mechanisms

- VCG mechanism (x, p):
  - x maximizes efficiency
  - payments:

$$p_i(\theta) = \left[\max_{o \in \mathcal{O}} \sum_{j \neq i} v_j(\theta_j, o)\right] - \sum_{j \neq i} v_j(\theta_j, x(\theta)).$$

  Each player submits a payment equal to the damage that her presence causes to the other players. "Internalize the externalities". W. Vickrey[1961], E. H. Clark [1971], T. Groves[1973]

- Generally, every truthful mechanism is a simple variation of the VCG mechanism: weighted VCG (affine maximizers)

μΠλ∀

Presentation Progress

μΠλ∀

Online Mechanism Design

Model:

- Discrete time periods $\mathcal{T} = \{1 < 2 < 3 < \dots\}$, indexed by t
- Agents arrive dynamically, having types

$$\theta_i = (a_i, d_i, w_i)$$

  - a_i: arrival time
  - d_i: departure time
  - w_i: valuation component

Online Mechanism Design

Model:

- Discrete time periods $\mathcal{T} = \{1 < 2 < 3 < \dots\}$, indexed by t
- Agents arrive dynamically, having types

$$\theta_i = (a_i, d_i, w_i)$$

  - a_i: arrival time
  - d_i: departure time
  - w_i: valuation component
- An *online* mechanism makes enforces a sequence of decisions $x(\theta) = (x^1(\theta), x^2(\theta), x^3(\theta), \dots)$
  - $x^t(\theta)$ has to be decided knowing only types θ_i with $a_i \leq t$
  - payments $p_i(\theta)$ have to be collected before d_i.

μΠλ∀

Limited Misreports

- We apply "natural" limitations to the available misreports $\widehat{\theta}_i = (\widehat{a}_i, \widehat{d}_i, \widehat{w}_i)$
  - No early-arrivals: $a_i \leq \widehat{a}_i$
  - No late-departures: $\widehat{d}_i \leq d_i$

μΠλ∀

Limited Misreports

- We apply "natural" limitations to the available misreports $\widehat{\theta}_i = (\widehat{a}_i, \widehat{d}_i, \widehat{w}_i)$
  - No early-arrivals: $a_i \leq \widehat{a}_i$
  - No late-departures: $\widehat{d}_i \leq d_i$
- No early-arrivals: very natural assumption.
- No late-departures: not so natural. We will comment on relaxing it later...

μΠλ∀

Single-valued Online Domains

- Every agent i has a set of *interesting decisions* $L_i \subseteq \mathcal{O}$ and is satisfied (at a same degree) whenever a decision from L_i is made:

$$v_i(\theta_i, x(\theta)) = \begin{cases} r_i, & \text{if } x^t(\theta) \in L_i \text{ for some } t \in [a_i, d_i], \\ 0, & \text{otherwise.} \end{cases}$$

r_i: reward (bid)
$\theta_i = (a_i, d_i, r_i)$

μΠλ∀

Single-valued Online Domains

- Every agent i has a set of *interesting decisions* $L_i \subseteq \mathcal{O}$ and is satisfied (at a same degree) whenever a decision from L_i is made:

$$v_i(\theta_i, x(\theta)) = \begin{cases} r_i, & \text{if } \ x^t(\theta) \in L_i \ \text{ for some } \ t \in [a_i, d_i], \\ 0, & \text{otherwise.} \end{cases}$$

  r_i: reward (bid)
  $\theta_i = (a_i, d_i, r_i)$
- Notation: $x_i(\theta_i) = 1$ i is "satisfied" and $x_i(\theta_i) = 0$ otherwise.
- Efficiency:

$$E(\theta) = \sum_{i=1}^{N} x_i(\theta) \cdot r_i$$

μΠλ∀

Critical Values

Definition (Critical Value)

$$v^c_{(a_i,d_i)}(\theta_{-i}) = \begin{cases} \inf \{ r'_i \mid x_i(\theta'_i, \theta_{-i}) = 1 \}, & \text{if this exists,} \\ \infty, & \text{otherwise.} \end{cases}$$

- The smallest reward she can report and still receive an interesting decision, keeping everything else unchanged.
- The critical value is *independent* of the agent's reported reward r_i

μΠλ∀

Monotonicity

- Notation: $\theta_i \prec_\theta \theta'_i \iff (a'_i \leq a_i) \wedge (d_i \leq d'_i) \wedge (r_i < r'_i)$: "better" type

μΠλ∀

Monotonicity

- Notation: $\theta_i \prec_\theta \theta'_i \iff (a'_i \leq a_i) \wedge (d_i \leq d'_i) \wedge (r_i < r'_i)$: "better" type

Definition (Monotonicity)

A mechanism, in a single-valued online domain, is called monotonic if, for every agent i and types θ_i, θ'_i with $\theta_i \prec_\theta \theta'_i$,

$$x_i(\theta_i, \theta_{-i}) = 1 \quad \Longrightarrow \quad x_i(\theta'_i, \theta_{-i}) = 1,$$

for every θ_{-i}.

If agent i gets allocated by reporting a type θ_i then she will also be allocated if she reports a "better" type θ'_i.

μΠλ∀

Truthfulness

Theorem

Every monotonic decision policy x can be truthfully implemented, i.e. there is a payment rule p such that mechanism (x, p) is truthful.

Proof.

Constructive:

$$p_i^t(\theta) = \begin{cases} v^c_{(\widehat{a}_i, \widehat{d}_i)}(\widehat{\theta}_{-i}), & \text{if } x_i(\widehat{\theta}_i, \widehat{\theta}_{-i}) = 1 \wedge t = \widehat{d}_i, \\ 0, & \text{otherwise}, \end{cases}$$

□

μΠλ∀

Truthfulness (cont.)

Definition

In any single-valued online domain, every truthful mechanism must be monotonic.

Proof.

- Every truthful mechanism must ask for *critical value* payments, i.e. $p_i(\theta) = v^c_{(a_i,d_i)}(\theta_{-i})$

$\square$

μΠλ∀

Presentation Progress

μΠλ∀

Expiring Items

- Canonical Expiring Items (CIE) problem: we have a *single*, indivisible, re-usable item to allocate at each period $t \in \mathcal{T}$. Bidders care to get only *one* instance of the item. No extra value if get more.

Expiring Items

- Canonical Expiring Items (CIE) problem: we have a *single*, indivisible, re-usable item to allocate at each period $t \in \mathcal{T}$. Bidders care to get only *one* instance of the item. No extra value if get more.
- The GREEDY auction:
  - Decision policy: at every time point t, allocate an item to the unassigned (unallocated so far) agent with the highest bid. Break ties randomly.

Expiring Items

- Canonical Expiring Items (CIE) problem: we have a *single*, indivisible, re-usable item to allocate at each period $t \in \mathcal{T}$. Bidders care to get only *one* instance of the item. No extra value if get more.
- The GREEDY auction:
  - Decision policy: at every time point t, allocate an item to the unassigned (unallocated so far) agent with the highest bid. Break ties randomly.
  - Payment policy: Every agent i that gets allocated, pays her critical bid $v^c_{(a_i,d_i)}(\theta_{-i})$ upon her departure d_i. Non-allocated agents make no payments.

μΠλ∀

Expiring Items

- Canonical Expiring Items (CIE) problem: we have a *single*, indivisible, re-usable item to allocate at each period $t \in \mathcal{T}$. Bidders care to get only *one* instance of the item. No extra value if get more.
- The GREEDY auction:
  - Decision policy: at every time point t, allocate an item to the unassigned (unallocated so far) agent with the highest bid. Break ties randomly.
  - Payment policy: Every agent i that gets allocated, pays her critical bid $v^c_{(a_i,d_i)}(\theta_{-i})$ upon her departure d_i. Non-allocated agents make no payments.
  - Greedy: myopically maximizes efficiency
  - If all agents are impatient, i.e. $a_i = d_i$, then Greedy Auction is a sequence of Vickrey (second-price) auctions.

Expiring Items

- Canonical Expiring Items (CIE) problem: we have a *single*, indivisible, re-usable item to allocate at each period $t \in \mathcal{T}$. Bidders care to get only *one* instance of the item. No extra value if get more.
- The GREEDY auction:
  - Decision policy: at every time point t, allocate an item to the unassigned (unallocated so far) agent with the highest bid. Break ties randomly.
  - Payment policy: Every agent i that gets allocated, pays her critical bid $v^c_{(a_i,d_i)}(\theta_{-i})$ upon her departure d_i. Non-allocated agents make no payments.
  - Greedy: myopically maximizes efficiency
  - If all agents are impatient, i.e. $a_i = d_i$, then Greedy Auction is a sequence of Vickrey (second-price) auctions.

Theorem

The GREEDY auction is truthful.

Proof.

GREEDY is monotonic. □

CEI: Upper bound

Theorem

The GREEDY Auction is 2-competitive.

μΠλ∀

CEI: Upper bound

Theorem

The GREEDY Auction is 2-competitive.

Proof.

Let x be the Greedy Auction decision rule, x^* the optimal offline decision rule and advesary chooses types θ. Also, let A be the set of all agents allocated by x, B by both x and x^* ,and C by x^* only.

- $E(x(\theta)) = \sum_{i \in A} r_i$, $E(x^*(\theta)) = \sum_{i \in B} r_i + \sum_{i \in C} r_i$

CEI: Upper bound

Theorem

The GREEDY Auction is 2-competitive.

Proof.

Let x be the Greedy Auction decision rule, x^* the optimal offline decision rule and advesary chooses types θ. Also, let A be the set of all agents allocated by x, B by both x and x^* ,and C by x^* only.

- $E(x(\theta)) = \sum_{i \in A} r_i$, $E(x^*(\theta)) = \sum_{i \in B} r_i + \sum_{i \in C} r_i$
- $B \subseteq A$, $r_i \geq 0 \quad \Longrightarrow \quad \sum_{i \in B} r_i \leq \sum_{i \in A} r_i$

CEI: Upper bound

Theorem

The GREEDY Auction is 2-competitive.

Proof.

Let x be the Greedy Auction decision rule, x^* the optimal offline decision rule and advesary chooses types θ. Also, let A be the set of all agents allocated by x, B by both x and x^* ,and C by x^* only.

- $E(x(\theta)) = \sum_{i \in A} r_i$, $E(x^*(\theta)) = \sum_{i \in B} r_i + \sum_{i \in C} r_i$
- $B \subseteq A$, $r_i \geq 0 \implies \sum_{i \in B} r_i \leq \sum_{i \in A} r_i$
  For every $i \in C$ allocated only by x^* at some period t_i, x allocates some other agent j_i at the same period t_i for which $r_i \leq r_{j_i}$. So
- $\sum_{i \in C} r_i \leq \sum_{i \in C} r_{j_i} \leq \sum_{j \in A} r_j$

CEI: Upper bound

Theorem

The GREEDY Auction is 2-competitive.

Proof.

Let x be the Greedy Auction decision rule, x^* the optimal offline decision rule and advesary chooses types θ. Also, let A be the set of all agents allocated by x, B by both x and x^* ,and C by x^* only.

- $E(x(\theta)) = \sum_{i \in A} r_i$, $E(x^*(\theta)) = \sum_{i \in B} r_i + \sum_{i \in C} r_i$
- $B \subseteq A\ ,\ r_i \geq 0 \quad \Longrightarrow \quad \sum_{i \in B} r_i \leq \sum_{i \in A} r_i$
  For every $i \in C$ allocated only by x^* at some period t_i, x allocates some other agent j_i at the same period t_i for which $r_i \leq r_{j_i}$. So
- $\sum_{i \in C} r_i \leq \sum_{i \in C} r_{j_i} \leq \sum_{j \in A} r_j$
- $E(x^*(\theta)) \leq 2E(x(\theta)) = \sum_{i \in A} r_i \quad \Longrightarrow \quad \max_\theta \frac{E(x^*(\theta))}{E(x(\theta))} \leq 2$

□

CEI: Lower bound

Theorem

No truthful online auction for the CEI problem can be $(2-\epsilon)$*-competitive, for every* $0 < \epsilon < 1$.

CEI: Lower bound

Theorem

No truthful online auction for the CEI problem can be $(2-\epsilon)$*-competitive, for every* $0 < \epsilon < 1$.

Corollary

$$\mathrm{CR}_{\mathrm{CEI}} = 2.$$

μΠλ∀

CEI: An Impossibility Result

Theorem

No truthful auction for the CEI problem can have a constant competitive ratio, if we relax the no late-departures assumption and allow arbitrary misreports of departure.

CEI: Extensions

- k re-usable goods instead of 1: The Greedy Auction (k unallocated agents with largest bids) still is 2-competitive.

CEI: Extensions

- k re-usable goods instead of 1: The Greedy Auction (k unallocated agents with largest bids) still is 2-competitive.
- The asynchronous model: time is continuous and not discrete. $\mathrm{CR} = 5$

CEI: Extensions

- k re-usable goods instead of 1: The Greedy Auction (k unallocated agents with largest bids) still is 2-competitive.
- The asynchronous model: time is continuous and not discrete. $\mathrm{CR} = 5$
- Relax truthfulness: $\mathrm{CR} \in [\phi, 2]$, $\phi \approx 1.618034$.

CEI: Extensions

- k re-usable goods instead of 1: The Greedy Auction (k unallocated agents with largest bids) still is 2-competitive.
- The asynchronous model: time is continuous and not discrete. $\mathrm{CR} = 5$
- Relax truthfulness: $\mathrm{CR} \in [\phi, 2]$, $\phi \approx 1.618034$.
- Taking the auction's revenue as a performance criterion: no constant CR.

CEI: Open problems

- Randomized auctions
- Revenue maximizing auctions. Revenue competitve ratio with respect to $h = \frac{b}{a}$, $[a, b]$ bid interval.

μΠλ∀

Limited Supply

- Canonical Limited Supply (CLS) problem: we have a *single*, indivisible item to allocate at only *one* agent.
- Assumption: All bids are different (easy to relax)

μΠλ∀

Limited Supply

- Canonical Limited Supply (CLS) problem: we have a *single*, indivisible item to allocate at only *one* agent.
- Assumption: All bids are different (easy to relax)
- We need a weaker adversarial model: otherwise trivially infinite CRs
- Random Ordering Hypothesis: the adversary selects bids r_i, arrival-departure intervals $[a_i, d_i]$ but the matching is done randomly (and uniformly).
- Maximize *expected* efficiency or revenue.

CLS: Adaptive, threshold auctions

Definition

For every $k = 1, 2, \ldots, N$ define $\mathcal{A}(k)$ to be the following auction for the CLS problem:

(i) *Learning Phase*: Make no allocation until you receive the k'th bid at time period τ. Let $p \geq q$ be the two top bids received so far.

(ii) *Transition Phase*: If some agent i with bid $w_i = p$ is still active at time period τ, then allocate i for a payment of q (breaking ties randomly).

(iii) *Accepting Phase*: If no agent got allocated during the transition phase (i.e. at τ), allocate the first agent to arrive after τ bidding at least p (no ties possible), for a payment of p. If no such agent arrives, allocate the last bidder to arrive.

μΠλ∀

The Classic Secretary Problem

- N applicants are interviewed for a job opening, one after the other.
- They arrive in a *random* order (random-ordering) hypothesis

μΠλ∀

The Classic Secretary Problem

- N applicants are interviewed for a job opening, one after the other.
- They arrive in a *random* order (random-ordering) hypothesis
- After reviewing each applicant, we must irrevocably decide whether to hire her or not.

μΠλ∀

The Classic Secretary Problem

- N applicants are interviewed for a job opening, one after the other.
- They arrive in a *random* order (random-ordering) hypothesis
- After reviewing each applicant, we must irrevocably decide whether to hire her or not.
- We do not know the qualification (quality) of an applicant, until we actually interview her.
- How can we maximize the probability of hiring the best applicant?

The Classic Secretary Problem

- N applicants are interviewed for a job opening, one after the other.
- They arrive in a *random* order (random-ordering) hypothesis
- After reviewing each applicant, we must irrevocably decide whether to hire her or not.
- We do not know the qualification (quality) of an applicant, until we actually interview her.
- How can we maximize the probability of hiring the best applicant?
- Interview the first $\left\lfloor \frac{N}{e} \right\rfloor$ applicants without hiring any of them and then hire the first applicant ranking higher than all these first $\left\lfloor \frac{N}{e} \right\rfloor$ candidates.
- Probability of success at least $\frac{1}{e} \approx 36.8\%$, as $N \rightarrow \infty$.

Optimal Stopping Theory

CLS: Upper bounds

Theorem

As $N \to \infty$, the adaptive auction $\mathcal{A}(\lfloor \frac{N}{e} \rfloor)$ is e-competitive for efficiency and e^2-competitive for revenue.

$e \approx 2.718$, $e^2 \approx 7.389$

μΠλΑ

CLS: Upper bounds

Theorem

As $N \to \infty$, the adaptive auction $\mathcal{A}(\lfloor \frac{N}{e} \rfloor)$ is e-competitive for efficiency and e^2-competitive for revenue.

$e \approx 2.718$, $e^2 \approx 7.389$

Theorem

As $N \to \infty$, the adaptive auction $\mathcal{A}(\lfloor \frac{N}{2} \rfloor)$ is $\frac{2}{\ln 2}$-competitive for efficiency and 4-competitive for revenue

$\frac{2}{\ln 2} \approx 2.885$

μΠλ∀

CLS: Upper bounds

Theorem

As $N \to \infty$, the adaptive auction $\mathcal{A}(\lfloor \frac{N}{e} \rfloor)$ is e-competitive for efficiency and e^2-competitive for revenue.

$e \approx 2.718$, $e^2 \approx 7.389$

Theorem

As $N \to \infty$, the adaptive auction $\mathcal{A}(\lfloor \frac{N}{2} \rfloor)$ is $\frac{2}{\ln 2}$-competitive for efficiency and 4-competitive for revenue

$\frac{2}{\ln 2} \approx 2.885$

Corollary

As $N \to \infty$,

$$\mathrm{CR}^{E}_{\mathrm{CLS}} \leq e \approx 2.718 \qquad \text{and} \qquad \mathrm{CR}^{R}_{\mathrm{CLS}} \leq 4.$$

CLS: Extensions and Open Problems

- Lower bound $\rightarrow$ efficiency: 2, revenue: $\frac{3}{2}$
- Multi-item case (identical items): constant (very large) CRs and $CR \rightarrow 1$ as $k \rightarrow \infty$
- Matroid Secretary problems

CLS: Extensions and Open Problems

- Lower bound $\rightarrow$ efficiency: 2, revenue: $\frac{3}{2}$
- Multi-item case (identical items): constant (very large) CRs and $CR \rightarrow 1$ as $k \rightarrow \infty$
- Matroid Secretary problems

Open problems:

- Bids drawn independently from an unknown distribution (our upper bounds still hold)
- Close CR "gaps"

μΠλ∀

References

D. C. Parkes
Online Mechanism Design.
In N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani, eds., *Algorithmic Game Theory*, Cambridge University Press, 2007. Chapter 16.

M. T. Hajiaghayi, R. Kleiberg, M. Mahdian and D. C. Parkes
Online auctions with re-usable goods.
In *Proceedings of the 6th Conference on Electronic Commerce (EC' 05)*, pp.165-174,2005.

M. T. Hajiaghayi, R. Kleinberg. and D. C. Parkes
Adaptive limited supply auctions.
In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC '04)*, 2004.

μΠλ∀

THE END

Thank you!

μΠλ∀

μΠλΑ