Algorithmic Game Theory - Part 1

Online Mechanism Design

Nikolidaki Aikaterini
aiknikol@yahoo.gr

Corelab,
NTUA

May 2016

Overview

Mechanism Design

Mechanism Design = Algorithm Design + Incentives

- **Direct revelation mechanisms** with dominant truthful strategies
- Mechanism = (Allocation Rule, Payment Rule) = (f, p)
- For which allocation rule (social choice function) are there payment functions so that the resulting mechanism is **truthful**?
  - Example: VCG mechanism ⇒ selecting the outcome with the maximum total value

Truthful Mechanisms

Definition (Truthful Mechanism)

A mechanism is truthful when the outcome and the payment functions are s.t. the players gain nothing by not declaring their true values. This notion of truthfulness is called dominant strategy truthfulness since declaring true values is a dominant strategy for each player.

Theorem (Revelation Principle)

For every mechanism M that has dominant strategies, there is an equivalent truthful mechanism M' that for every bid vector chooses the same outcome and pays the same amounts

Overview

1. Mechanism Design
   - Truthful Mechanisms
2. Scheduling Problems
   - Related Machines
   - Unrelated Machines
3. Online Mechanisms
   - Dynamic Auction with Expiring Items
   - Secretary Problem
   - Adaptive Limited-Supply Auction
4. Procurement Auctions
   - Frugal Path Auctions
   - Budget Feasible Mechanisms
   - Learning on a Budget: Posted Price Mechanisms

Related Machines

- Processing times of tasks: $p_1 \geq ... \geq p_m$
- Speeds: $s_1, ..., s_n$
- Workload assigned to machine i: w_i
- Makespan: $C(w, s) = \max_i \frac{w_i}{s_i}$

⋆ It's a typical **single-parameter** problem

⋆ The optimal allocation is monotone $\Rightarrow$ truthful

⋆ But, it cannot be computed in polynomial time unless $P = NP$

Unrelated Machines

- There are **n** machines and **m** tasks
- Machine **i** can execute task **j** in t_{ij}
- Allocate the tasks to machines to **minimize the makespan**
  - Task **j** is allocated to **exactly one i**: $\forall j, \sum_{i=1}^{n} x_{ij} = 1$

⋆ The problem is NP-hard

⋆ Nisan and Ronen (game theoretic point of view): each machine **i** is a **rational** agent who is the only one knowing the values of t_i

Definition (Monotonicity Property)

An allocation algorithm f is called monotone if it satisfies the following property: for every two sets of tasks t and t' which differ only on machine i (i.e., on the i-the row) the associated allocations x and x' satisfy

$$(x_i - x_i') \cdot (t_i - t_i') \leq 0$$

where $\cdot$ denotes the dot product of the vectors, that is,
$\sum_{j=1}^{m} (x_{ij} - x_{ij}') \cdot (t_{ij} - t_{ij}') \leq 0$

Theorem (Saks & Yu)

A mechanism (f, p) is truthful iff its allocation algorithm f satisfies the Monotonicity Property.

Upper Bounds - Results - Unrelated Machines

- Nisan & Ronen (2001): **n** for any truthful deterministic mechanism
- Nisan & Ronen (2001): 1.75 for randomized universally truthful mechanism for 2 machines
- Mualem & Shapira (2007): 0.875n randomized universally truthful mechanism for n machines
- Lu & Yu (2008): 1.67 and later 1.59 for randomized universally truthful mechanism for n machines
- Christodoulou et al. (2007): $\frac{n+1}{2}$ for fractional mechanisms (optimal for task independent: A task-independent algorithm is any algorithm that, in order to allocate task j, only considers the processing times t_{ij} that concern the particular task.)

Lower Bounds - Results - Unrelated Machines

- Nisan & Ronen (2001): 2 for any truthful deterministic mechanism for 2 machines
- Christodoulou et al. (2007): $1 + \sqrt{2}$ for three or more machines
- Koutsoupias & Vidali (2007): $1 + \phi =$ **2.61** for n machines
- Mualem & Shapira (2007): $2 - \frac{1}{n}$ for randomized truthful in expectation mechanisms
- Christodoulou et al. (2007): $1 + \sqrt{2}$ for fractional domains
- Deterministic & Fractional mechanisms: **tight** bounds for 2 machines
- Randomized mechanisms: GAP with 1.5 lower and 1.59 upper bound

Lower Bounds - Unrelated Machines

Theorem

Let t be a set of tasks and let $x = x(t)$ be the allocation produced by a truthful mechanism. Suppose that we **change** *only the* **processing times of machine i** *in such a way that $t'_{ij} > t_{ij}$ when $x_{ij} = 0$, and $t'_{ij} < t_{ij}$ when $x_{ij} = 1$. A truthful mechanism does not change the* **allocation to machine i**, *i.e.,* $x_i(t') = x_i(t)$.

Theorem

Any truthful mechanism has approximation ratio of at least 2 for **two** *or* **more** *machines.*

Theorem

Any truthful mechanism has approximation ratio of at least $1 + \sqrt{2}$ for **three** *or* **more** *machines.*

Example - Unrelated Machines

Exapmle 1: Let $n = 2$ and $m = 3$ and t_{ij}=1

- Allocate all tasks to a single machine

$$t = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \Rightarrow t' = \begin{pmatrix} 1-\epsilon & 1-\epsilon & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Then, $\frac{2(1-\epsilon)}{1} \approx$ 2-approximation

- Partition them: first two to machine 1 and the rest to machine 2

$$t = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \Rightarrow t' = \begin{pmatrix} 1 & 1 & 1 \\ 1+\epsilon & 1+\epsilon & 0 \end{pmatrix}$$

Then, $\frac{2}{1+\epsilon} \approx$ 2-approximation

General idea of proof $1+\sqrt{2}=2.41$

Let set of tasks for some parameter $a > 1$.

- This set of tasks admits two distinct allocation
- The first three tasks need to be assigned to a single machine

$$t = \begin{pmatrix} 0 & \infty & \infty & a & a \\ \infty & 0 & \infty & a & a \\ \infty & \infty & 0 & a & a \end{pmatrix} \Rightarrow_{allocation} t = \begin{pmatrix} 0 & \infty & \infty & 1 & 1 \\ \infty & 0 & \infty & a & a \\ \infty & \infty & 0 & a & a \end{pmatrix} \Rightarrow$$

$$t' = \begin{pmatrix} a & \infty & \infty & 1-\epsilon & 1-\epsilon \\ \infty & 0 & \infty & a & a \\ \infty & \infty & 0 & a & a \end{pmatrix}$$

Then, $\frac{a+2}{a} \approx 2.41$-approximation, where $a = \sqrt{2}$

Open Questions

- ? Characterize the set of truthful mechanisms for unrelated machines
- ? Close the **gap** between the lower 2.61 and the upper n bound on the approximation ratio for unrelated machines
- ? Randomized & Fractional mechanisms
- ? Deterministic monotone PTAS exists for the related problem

Overview

1. Mechanism Design
   - Truthful Mechanisms
2. Scheduling Problems
   - Related Machines
   - Unrelated Machines
3. Online Mechanisms
   - Dynamic Auction with Expiring Items
   - Secretary Problem
   - Adaptive Limited-Supply Auction
4. Procurement Auctions
   - Frugal Path Auctions
   - Budget Feasible Mechanisms
   - Learning on a Budget: Posted Price Mechanisms

Online Mechanisms

- Extend the methods of mechanism design to dynamic environments with multiple agents and private information
- Direct-revelation online mechanism
- Truthful auctions for domains with expiring items and limited-supply items
- Secretary Problem
- Dynamic VCG mechanism

Dynamic Auction with Expiring Items

- Discrete time periods: $T = 1, 2, ...$
- Type of an agent i: $\theta_i = (a_i, d_i, w_i) \in T \times T\mathbb{R}_{>0}$
- The item is allocated in some period $t \in [a_i, d_i]$
- The value for allocation of the single item in some t: w_i
- Payment p is collected from the agent
- Quasi linear utility function: $w_i - p$

Example

$\Longrightarrow$

per hour

- Let the next buyers with types:
  - Buyer1: $\theta_1 = (9{:}00,\ 11{:}00,\ 100)$
  - Buyer2: $\theta_2 = (9{:}00,\ 11{:}00,\ 80)$
  - Buyer3: $\theta_3 = (10{:}00,\ 11{:}00,\ 60)$
- Results:
  - * Buyer1 take item for 80$ in the 1st hour
  - * Buyer2 take item for 60$ in the 2nd hour

Example Cont

Lie in the value:

$\theta_1 = (9{:}00, 11{:}00, 61)$

Results:

* Buyer2 take item for 61$ in the 1st hour
* Buyer1 take item for 60$ in the 2nd hour

Lie in the arrival time:

$\theta_1 = (10{:}00, 11{:}00, 100)$

Results:

* Buyer2 take item for 0$ in the 1st hour
* Buyer1 take item for 60$ in the 2nd hour

Online Mechanism Model

- Discrete time periods: $T = 1, 2, ...$
- Set of feasible outcomes at time t.
- Sequence of decisions at time t.
- Type of an agent i: $\theta_i = (a_i, d_i, w_i) \in T \times T \in \mathbb{R}_{>0}$
- Valuation function v_i
- Quasi linear utility function: $w_i - p$
- Arrival period is the first time the agent may report its type.
- Valuation component may depend on choices and time

Online Mechanism Model

Definition (Direct-Revelation Online M)

A direct-revelation online mechanism, $M(\pi, x)$ restricts each agent to making a single claim about its type, and defines **decision policy** $\pi = \{\pi^t\}^{t \in T}$ and **payment policy** $x = \{x^t\}^{t \in T}$ where decision $\pi^t(h^t) \in K(h^t)$ is made in state h^t and payment $x_i^t(h^t) \in \mathbb{R}$ is collected from each agent i.

Example:

- h^t: list of reported agent types in period t (agent is allocated or not)
- k: decision to allocate the item in current period to some agent that is present and unallocated

Definition (Limited Misreports)

Let $C(\theta_i) \subseteq \Theta_i$ for $\theta_i) \in \Theta_i$ denote the set of available misreports to an agent with true type θ_i.

Online Mechanism Model

- No early arrival misreports: $a_i' \geq a_i$
- No late departures: $d_i' \leq d_i$
- Agent wasn' t there

Definition (Truthful -DSIC)

Online mechanism $M = (\pi, x)$ is truthful (or dominant strategy incentive compatible - DSIC) given limited misreports C if

$$v_i(\theta_i, \pi(\theta_i, \theta'_{-i})) - p(\theta_i, \theta'_{-i}) \geq v_i(\theta_i, \pi(\theta_i^*, \theta'_{-i})) - p(\theta_i^*, \theta'_{-i})$$

Online Mechanism Model

Definition (critical value)

The **critical value** for agent i given type $\theta_i = (a_i, d_i, (r_i))$ and deterministic policy π in a single-valued domain:

$$v_{(a_i,d_i)} = \begin{cases} \min r'_i & \text{s.t } \pi_i(\theta'_i, \theta_{-i}) = 1, \text{ for } \theta'_i \\ \infty & \text{if no such } r'_i \text{ exists} \end{cases}$$

* Critical value: the bid under which agent i is not allocated any item

Definition (Monotonic Decision Policy)

Agent i gets an item when bidding $r_i \Rightarrow$
still gets an item when bidding $r'_i > r_i$.

Online Mechanism Model

Theorem

A monotonic decision policy can be truthfully implemented using the critival values as payments.

Theorem

A decision policy that is truthfully implementable in and individually rational (IR) mechanism with the extra contrain that only reasonable missreporting is allowed must be monotonic

Competitive Analysis

- Perform worst-case analysis
- A sequence of types are generated by an adversary $\Rightarrow$ the performance becomes as bad as possible
- How effectively is our online algorithm with that of an optimal offline algorithm with full information about agent types

Lower Bounds-Online Mechanism

Theorem

No truthful, IR, and deterministic online auction can obtain a $(2 - \epsilon)$-apx for efficiency in the expiring items environment with no early-arrival and no late-departure misreports, for any constant $\epsilon > 0$.

Theorem

No truthful, IR, and deterministic online auction can obtain a constant-apx for efficiency in the expiring items environment with no early-arrival misreports but arbitrary misreports of departure.

Secretary Problem

- Job applicants: N
- Each applicant has a rank
- While interviewing the rank of the current applicant is learnt relative to the others who were interviewed
- The interviewer must make an irrevocable decision about whether or not to hire.
- Goal: Maximize the probability of selecting the best applicant.
- An adversary can choose an arbitrary set of N qualities but **not** the order (the order of the applicants is sampled uniformly at random).
- Optimal Policy:
  - * interview the first $t-1$ applicants.
  - * hire the first subsequent applicant that is better than all the previous $t-1$ applicants.
- What is the best t?
- Turns out it's an $1/e$ fraction of N

Adaptive Limited-Supply Auction

- An online mechanism is **c-competitive for revenue** if

$$\min \mathbb{E}\left\{\frac{Rev(p(\theta_z))}{R^*(\theta(z))}\right\} \geq \frac{1}{c}$$

- The optimal policy has:
  - Learning phase
  - Accepting phase
- Observe $\lfloor N/e \rfloor$ reports and then price at the maximal value p received $\Rightarrow$ Sell to the first agent to subsequently report a value greater than this price.

Adaptive Limited-Supply Auction

Auction:

- In period τ When the $\lfloor N/e \rfloor$th bid is received, let $p \geq q$ be the bid values
- If p is still present in period τ then sell it to that agent at price q. (break ties randomly)
- Else, sell to the next agent to bid a price at least p at price p

Example:

- Let the next agents with types:
  - $\theta_1 = (1, 7, 6)$
  - $\theta_2 = (3, 7, 2)$
  - $\theta_3 = (4, 8, 4)$
  - $\theta_4 = (6, 7, 8)$
  - θ_5 in later period
  - θ_6 in later period
- Transition to accepting phase occurs when agent $\lfloor 6/e \rfloor = 2$ bids
  - 4: wins in t=6 for p=6
- If $\theta_1 = (5, 7, 6)$
  - 1: wins in t=5 for p=4
- $p = 6,\ q = 2$: 1 wins 2
- If $\theta_1 = (1, 2, 6)$: sold to 4 in t=6

Adaptive Limited-Supply Auction

Theorem

Previous auction is **strongly truthful** *in the single-unit, limited supply environmnent with no early-arrival misreports*

Theorem

Previous auction is **e+o(1)-competitive** *for efficiency in the single-unit, limited supply environmnent in the limit as* $N \to \infty$