CS599: Algorithm Design in Strategic Settings
Fall 2012
Lecture 4: Prior-Free Single-Parameter Mechanism Design

Instructor: Shaddin Dughmi

Administrivia

- HW out, due Friday 10/5
    - Very hard (I think)
    - Discuss together and with me (but write up independently)

Outline

Outline

Motivated by impossibilities, we agreed to focus on settings where monetary payments can be used to align incentives.

The Quasi-linear Setting

Formally, $\mathcal{X} = \Omega \times \mathbb{R}^n$.

- Ω is the set of allocations
- For $(\omega, p_1, \ldots, p_n) \in \mathcal{X}$, p_i is the payment from (or to) player i.

and player $i's$ utility function $u_i : T_i \times \mathcal{X} \to \mathbb{R}$ takes the following form

$$u_i(t_i, (\omega, p_1, \ldots, p_n)) = v_i(t_i, \omega) - p_i$$

for some valuation function $v_i : T_i \times \Omega \to \mathbb{R}$.

We say players have quasilinear utilities.

Example: Single-item Allocation

- $\Omega = \{e_1, \ldots, e_n\}$
- $u_i(t_i, (\omega, p_1, \ldots, p_n)) = t_i \omega_i - p_i$

The Mechanism Design Problem

Task of Mechanism Design in Quasilinear settings

Find a "good" allocation rule $f : T \to \Omega$ and payment rule $p : T \to \mathbb{R}^n$ such that the following mechanism is incentive-compatible:

- Solicit reports $\widetilde{t_i} \in T_i$ from each player i (simultaneous, sealed bid)
- Choose allocation $f(\widetilde{t})$
- Charge player i payment $p_i(\widetilde{t})$

We think of the mechanism as the pair (f, p).

Incentive Compatibility

Incentive-compatibility (Dominant Strategy)

A mechanism (f, p) is dominant-strategy truthful if, for every player i, true type t_i, possible mis-report $\widetilde{t_i}$, and reported types t_{-i} of the others, we have

$$\mathbf{E}[v_i(t_i, f(t)) - p_i(t)] \geq \mathbf{E}[v_i(t_i, f(\widetilde{t_i}, t_{-i})) - p_i(\widetilde{t_i}, t_{-i})]$$

The expectation is over the randomness in the mechanism.

Incentive Compatibility

Incentive-compatibility (Dominant Strategy)

A mechanism (f, p) is dominant-strategy truthful if, for every player i, true type t_i, possible mis-report $\widetilde{t_i}$, and reported types t_{-i} of the others, we have

$$\mathbf{E}[v_i(t_i, f(t)) - p_i(t)] \geq \mathbf{E}[v_i(t_i, f(\widetilde{t_i}, t_{-i})) - p_i(\widetilde{t_i}, t_{-i})]$$

The expectation is over the randomness in the mechanism.

Incentive-compatibility (Bayesian)

A mechanism (f, p) is Bayesian incentive compatible if, for every player i, true type t_i, possible mis-report $\widetilde{t_i}$, the following holds in expectation over $t_{-i} \sim D|t_i$

$$\mathbf{E}[v_i(t_i, f(t)) - p_i(t)] \geq \mathbf{E}[v_i(t_i, f(\widetilde{t_i}, t_{-i})) - p_i(\widetilde{t_i}, t_{-i})]$$

The expectation is over randomness in both the mechanism and the other players' types.

Outline

Question

What is a “good” mechanism?

Answer

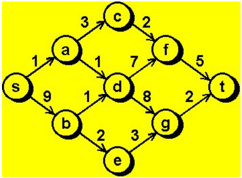
Depends what you are looking for.

- Researchers and practitioners have considered many objectives and hard constraints on desirable mechanisms.
- The task of mechanism design is then to find a mechanism maximizing the objective subject to the constraints.

Example: Single-minded Combinatorial Allocation

- n players, m non-identical items
- For each player, publicly known subset A_i of items the player desires
- Allocations: partitions of items among players
- Each player has type $v_i \in \mathbb{R}_+$, indicating his value for receiving a bundle including A_i (0 otherwise)
- Goal: Social welfare (sum of values of players who receive their desired bundles)

Shortest Path Procurement



- Players are edges in a network, with designated source/sink
- Player i's private data (type): cost $c_i \in \mathbb{R}_+$
- Outcome: choice of s-t shortest path to buy, and payment to each player
- Utility of a player for an outcome is his payment, less his cost if chosen.

Goal: buy path with lowest total cost (welfare), or buy a path subject to a known budget, . . .

Example: Public Project

- Designer considering whether to build a project which costs designer C (public)
- n players, each with private type $v_i \in \mathbb{R}_+$, indicating value for project
- Outcome: Choice of whether or not to build project, and how much to charge each player.
- Possible goal: Build if $\sum_i v_i > C$, charging players enough to cover cost C

Constraints

- Incentive compatibility
- Polynomial-time
- Individual Rationality: never charge a player more than his (reported) value for an allocation.
- Nonnegative [Non-positive] Transfers: never pay [get paid by] a player
  - e.g. Combinatorial allocation, Shortest path procurement
- Budget constraints: sum of total payments to agents must respect budget
  - e.g. reverse (procurement) auctions
- Budget balance: sum of total payments must exceed cost of allocation
  - e.g. public project

Objectives: Prior-free

Given an instance of a mechanism design problem,

- An objective is a map from outcome (allocation and payments) to the real numbers.
- A benchmark is a real number "goalpost"

Single-item auction

- Objective: welfare, i.e. the value of the winning player.
- Benchmark: the maximum welfare over all allocations.

Objectives: Prior-free

Given an instance of a mechanism design problem,

- An objective is a map from outcome (allocation and payments) to the real numbers.
- A benchmark is a real number “goalpost”

Single-item auction

- Objective: welfare, i.e. the value of the winning player.
- Benchmark: the maximum welfare over all allocations.

In prior-free settings, we traditionally judge an algorithm by the worst-case ratio between the performance of the mechanism and the benchmark.

The worst-case approximation ratio of a mechanism is the maximum, over all inputs, of the benchmark divided by the objective of the outcome output by the mechanism.

Objectives: Bayesian

In the presence of a distribution over inputs, no need for a benchmark.

Judge a mechanism by the expected objective over the various inputs.

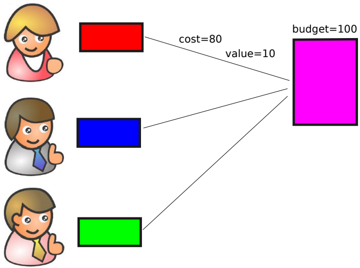
Outline

Next Up

We will begin our exploration of the space of mechanism design problems by restricting attention to

- Prior-free settings, with the goal of designing dominant-strategy truthful mechanisms
- Quasi-linear utilities, so our mechanisms will use payments
- Problems that are single-parameter

Example: Knapsack Allocation



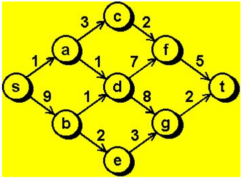
- n players, each player i with a task requiring c_i time
- Machine has total processing time B (public)
- Player i has (private) value v_i for his task

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

Example: Single-minded Combinatorial Allocation

- n players, m non-identical items
- For each player, publicly known subset A_i of items the player desires
- Allocations: partitions of items among players
- Each player has type $v_i \in \mathbb{R}_+$, indicating his value for receiving a bundle including A_i (0 otherwise)
- Goal: Social welfare (sum of values of players who receive their desired bundles)

Shortest Path Procurement



- Players are edges in a network, with designated source/sink
- Player i's private data (type): cost $c_i \in \mathbb{R}_+$
- Outcome: choice of s-t shortest path to buy, and payment to each player
- Utility of a player for an outcome is his payment, less his cost if chosen.

Goal: buy path with lowest total cost (welfare), or buy a path subject to a known budget, . . .

Scheduling

- Designer has m jobs, with publicly known sizes $p_1, \dots, p_m$
- n players, each own a machine
- Player i's type $t_i \in \mathcal{R}$ is time (cost) per unit job
- Outcome: schedule mapping jobs onto machines, and payment to each player
- Utility of a player for a schedule is his payment, less the total time spent processing assigned jobs

Goal: Find schedule minimizing makespan: the time at which all jobs are complete

Single-parameter Problems

Informally

- There is a single homogenous resource (items, bandwidth, clicks, spots in a knapsack, etc).
- There are constraints on how the resource may be divided up.
- Each player's private data is his "value (or cost) per unit resource."

Single-parameter Problems

Formally

- Each player i's type is a single real number t_i. Player i's type-space T_i is an interval in $\mathbb{R}$.
- Each outcome $\omega \in \Omega$ is a vector in $\mathbb{R}^n$.
- Player i's valuation function is $v_i(t_i, x) = t_i x_i$

Single-parameter Problems

Formally

- Each player i's type is a single real number t_i. Player i's type-space T_i is an interval in $\mathbb{R}$.
- Each outcome $\omega \in \Omega$ is a vector in $\mathbb{R}^n$.
- Player i's valuation function is $v_i(t_i, x) = t_i x_i$

Examples

- Single-item allocation: Ω is set of standard basis vectors, t_i is player i's value for an item.
- Knapsack allocation: Ω is the set of indicator vectors of players who fit in the knapsack, t_i is player i's value for being included.
- Scheduling: Ω is the set of possible load vectors, $-t_i$ is player i's time per unit load.

Interpretation and Importance

- Models win/lose situations, and situations where a homogeneous resource is to be divided.
- Simple and pervasive
- Incentive-compatible mechanisms admit a simple and permissive characterization.

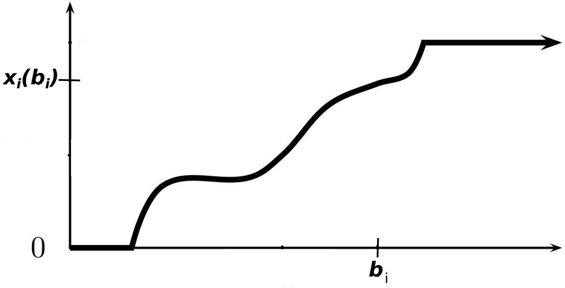
Outline

Myerson's Lemma (Dominant Strategy)

A mechanism (x, p) for a single-parameter problem is dominant-strategy truthful if and only if for every player i and fixed reports b_{-i} of other players,

- $x_i(b_i)$ is a monotone non-decreasing function of b_i
- $p_i(b_i)$ is an integral of $b_i\ dx_i$. Specifically, when $p_i(0) = 0$ then

$$p_i(b_i) = b_i \cdot x_i(b_i) - \int_{b=0}^{b_i} x_i(b)db$$

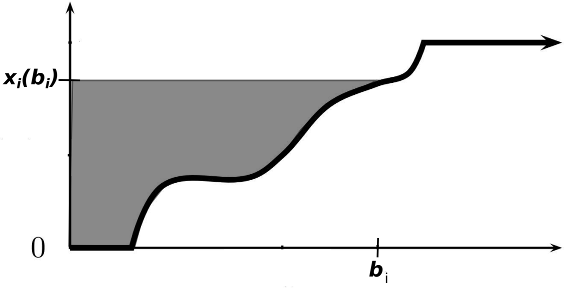


Myerson's Lemma (Dominant Strategy)

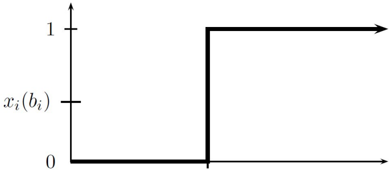
A mechanism (x, p) for a single-parameter problem is dominant-strategy truthful if and only if for every player i and fixed reports b_{-i} of other players,

- $x_i(b_i)$ is a monotone non-decreasing function of b_i
- $p_i(b_i)$ is an integral of $b_i \; dx_i$. Specifically, when $p_i(0) = 0$ then

$$p_i(b_i) = b_i \cdot x_i(b_i) - \int_{b=0}^{b_i} x_i(b)db$$



Interpretation of Myerson's Lemma



Utilitarian Single-item Allocation

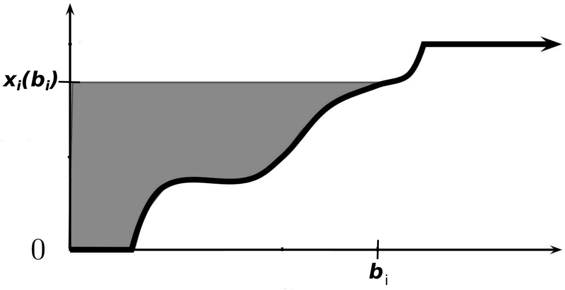
- Once a player wins, he remains a winner by increasing his bid (assuming other bids held fixed)
- The player must pay his critical value if he wins: the minimum bid he needs to win.
- Therefore, Vickrey is the unique welfare-maximizing, individually rational, single-item auction.

Same holds for every problem with a binary (win/lose) outcome per player.

Interpretation of Myerson’s Lemma

More Generally

As player increases his bid, he pays for each additional chunk of resource at a rate equal to the minimum bid needed to win that chunk.



Proof: Necessity

Figure

Monotonicity

- Assume for a contradiction that x_i is non-monotone. Let $b_i' > b_i$ with $x_i(b_i') < x_i(b_i)$.
- Two cases:
  1. $b_i \cdot (x_i(b_i) - x_i(b_i')) < p_i(b_i) - p_i(b_i')$
     Extra "value" gotten by reporting b_i truthfully is dominated by increase in price.
  2. $b_i \cdot (x_i(b_i) - x_i(b_i')) \geq p_i(b_i) - p_i(b_i')$
     Then also $b_i' \cdot (x_i(b_i) - x_i(b_i')) > p_i(b_i) - p_i(b_i')$, and a player with true value b_i' prefers to mis-report b_i.

Proof: Necessity

Payments

- Consider the utility of a player with type b_i reporting b_i'

$$b_i x_i(b_i') - p_i(b_i')$$

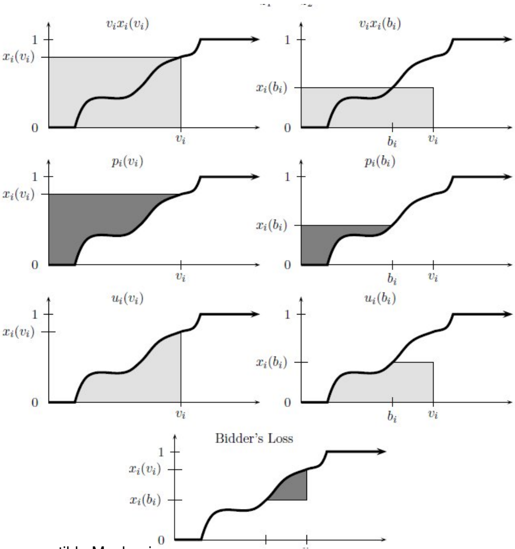
- For truthfulness, this expression must be maximized by setting $b_i' = b_i$
- This implies that the partial derivative w.r.t b_i', evaluated at $b_i' = b_i$, is zero

$$b_i \frac{dx_i}{db_i}(b_i) - \frac{dp_i}{db_i}(b_i) = 0$$

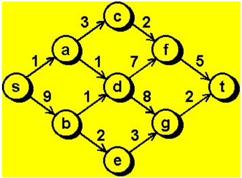
- Multiplying by db_i gives that p_i integrates $b_i dx_i$, as needed.

Proof: Sufficiency

Consider a player with true type v_i, and a possible mis-report $b_i < v_i$. (Exercise: consider $b_i > v_i$)



Example: Dijkstra Shortest Path



- Monotonicity: If an edge in the shortest path decreases its cost, it remains in the shortest path
- Critical Payments: We pay each edge the maximum possible cost it could report and still remain in the shortest path.

Figure

Outline

Bilateral Trade

A seller (player 1) and buyer (player 2) are looking to trade a single item initially held by the seller.

- Type of each player i is his value v_i for the item
- Two outcomes:
  - No trade: $(1, 0)$
  - Trade: $(0, 1)$
- Welfare maximizing allocation rule:

Bilateral Trade

A seller (player 1) and buyer (player 2) are looking to trade a single item initially held by the seller.

- Type of each player i is his value v_i for the item
- Two outcomes:
  - No trade: $(1, 0)$
  - Trade: $(0, 1)$
- Welfare maximizing allocation rule: trade if $v_2 > v_1$

Bilateral Trade

A seller (player 1) and buyer (player 2) are looking to trade a single item initially held by the seller.

- Type of each player i is his value v_i for the item
- Two outcomes:
  - No trade: $(1, 0)$
  - Trade: $(0, 1)$
- Welfare maximizing allocation rule: trade if $v_2 > v_1$

Question

Assuming no payments in the event of no-trade, describe the payment rule of the welfare-maximizing mechanism.

Next Lecture

We finally begin designing “interesting” mechanisms, specifically for problems that are NP-hard. The tricky part will be combining incentive-compatibility and polynomial-time.