Introduction to algorithmic mechanism design

Elias Koutsoupias
Department of Computer Science
University of Oxford

EWSCS 2014

March 5-7, 2014

Part I

Truthfulness

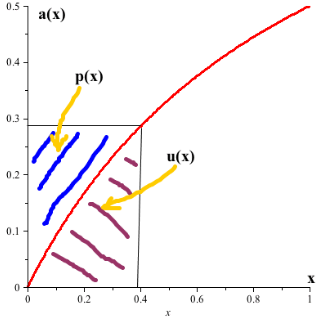
Main points on truthfulness

- In many computational problems, the input is controlled by selfish entities
- The algorithm must provide incentives to these entities to be truthful
- A necessary and sufficient condition is that the algorithm must be monotone
- This entails two conditions:
  - The solution must be the gradient of the utilities of the entities
  - The utilities must be convex
- The payments can be computed directly from the monotone solution

Truthfulness for single-parameter domains

The following picture summarizes truthfulness for single-parameter domains:

- x: private value
- $a(x)$: probability of getting the item
- $p(x)$: payment
- $u(x)$: utility of the bidder



The allocation function $a(x)$ must be monotone.

Convexity = Monotonicity = Truthfulness

For convex domains, truthfulness and convexity are identical notions.

	One parameter	Many parameters
valuation (or type)	$v \in R$	$v \in R^n$
utility	$u(v) = a(v) \cdot v - p(v)$	$u(v) = a(v) \cdot v - p(v)$
allocation	$a(v) = u'(v)$	$a(v) = \nabla u(v)$
payment	$p(v) = u^*(a(v))$	$p(v) = u^*(a(v))$

The condition of the allocation probabilities

$$a(v) = \left(\frac{\partial u(v)}{\partial v_1}, \ldots, \frac{\partial u(v)}{\partial v_n} \right)$$

connects the output of different inputs. This asks for a "holistic" algorithm.

Many players?

- A mechanism is truthful if every player is truthful: for every player i, $u_i(v)$ is convex in the type v_i of player i
- Major open problem: Find a good characterization for mechanisms in multi-player multi-parameter domains.

Example: single item

- The only deterministic truthful mechanisms are those that offer the item at a fixed price (independent of the value of the bidder)
- If there are many bidders, the price may depend on the values of the other bidders
- Second price auction (Vickrey auction) is truthful because

$$a_i(v) = \begin{cases} 0 & \text{when } v_i \leq \max_{k \neq i} v_k \\ 1 & \text{otherwise} \end{cases}$$

  and the payment is computed correctly
- First price auction is not truthful (it has the same allocation probabilities with the Vickrey auction, but the payment is not appropriate)

Part II

Domains

A more general domain

Two voters, three candidates:

	Alice	**Bob**	**Carol**
Voter 1	10	18	20
Voter 2	21	18	12

Or more generally: 2 players, 3 outcomes

	Outcome 1	**Outcome 2**	**Outcome 3**
Player 1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$
Player 2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$

- This is called *unrestricted* domain.
- Players declare their values and the mechanism selects an outcome and payments for each player.

The combinatorial auction domain

Problem (Combinatorial auction)

- *There are n players (bidders) and m objects (items)*
- *Each player i has a value $v_{i,S}$ for each subset (bundle) S of the objects. These are private values.*
- *Objective: Allocate the objects to the players to maximize the sum of the values of their bundles.*

Example (3 players, 2 items)

	O1	**O2**	**O3**	**O4**	**O5**	**O5**	**O7**	**O8**	**O9**
Player 1	$v_{1,12}$	$v_{1,1}$	$v_{1,1}$	$v_{1,2}$	$v_{1,2}$	0	0	0	0
Player 2	0	$v_{2,2}$	0	$v_{2,1}$	0	$v_{2,12}$	$v_{2,1}$	$v_{2,2}$	0
Player 3	0	0	$v_{3,2}$	0	$v_{3,1}$	0	$v_{3,2}$	$v_{3,1}$	$v_{3,12}$

The combinatorial auction domain - additive case

Example (3 players, 2 items)

	O1	O2	O3	O4	O5	O5	O7	O8	O9
Player 1	$v_{1,1} + v_{1,2}$	$v_{1,1}$	$v_{1,1}$	$v_{1,2}$	$v_{1,2}$	0	0	0	0
Player 2	0	$v_{2,2}$	0	$v_{2,1}$	0	$v_{2,1} + v_{2,2}$	$v_{2,1}$	$v_{2,2}$	0
Player 3	0	0	$v_{3,2}$	0	$v_{3,1}$	0	$v_{3,2}$	$v_{3,1}$	$v_{3,1} + v_{3,2}$

Example (Single-item auction)

	O1	O2	O3
Player 1	v_1	0	0
Player 2	0	v_2	0
Player 3	0	0	v_3

The shortest-path domain

- Given a graph, we want to find the shortest path from node s to point t
- Every edge e belongs to some agent who is willing to sell it at price v_e, which is a private value
- The players are the edges
- The outcomes are all possible paths between s and t
- The value for every edge e and path P is

$$v_{e,P} = \begin{cases} -v_e & e \in P \\ 0 & \text{otherwise} \end{cases}$$

Part III

The VCG mechanism

The VCG mechanism

- The Vickrey-Clarke-Groves (VCG) mechanism selects the outcome which **maximizes the social welfare**, i.e., the sum of the values of the players.
- The utility of player i is

$$u_i = \max_j \left\{ \sum_k v_{k,j} - h_i(v_{-i}) \right\},$$

where $h_i(v_{-i}) = \max_l \sum_{k \neq i} v_{k,l}$ is a quantity that depends only on the other players and is independent of the outcome.
- The payment of player i when the outcome is j is

$$p_i = -\sum_{k \neq i} v_{i,j} + \max_l \sum_{k \neq i} v_{k,l},$$

which can be described as: the player pays her value but she gets a discount equal to the increase of the global objective because of her participation.

	Outcome 1	**Outcome 2**	**Outcome 3**
Player 1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$
Player 2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$

Examples of VCG

	Outcome 1	**Outcome 2**	**Outcome 3**
Player 1	3	5	9
Player 2	8	7	1

- The VCG selects the second outcome with the maximum sum (5+7).
- Player 1 pays 8-7=1
- Player 2 payes 9-5=4

Truthfulness

Theorem

The VCG mechanism is truthful.

Proof: the utilities of VCG are convex functions.

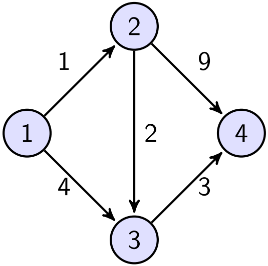
Single item

- VCG for a single item auction is the Vickrey (second-price) auction

	O1	O2	O3
Player 1	9	0	0
Player 2	0	12	0
Player 3	0	0	5

Player 2 gets the item and pays $9 - 0 = 9$

VCG for the shortest-path problem



- VCG selects a shortest path P: $P = (1, 2, 3, 4)$
- Edges not in P are paid nothing
- To compute the payment of an edge e on the path P:
  - We remove e and compute a shortest path P_e
  - The payment for edge e is

$$p_e = v_e + \text{length of } P_e - \text{length of } P$$

For example,

  - for edge $[1, 2]$, $P_e = (1, 3, 4)$. The payment is $1 + 7 - 6 = 2$
  - for edge $[2, 3]$, $P_e = (1, 3, 4)$. The payment is $2 + 7 - 6 = 1$

Efficient algorithm?

- To compute the paths of the VCG for the shortest path problem, we need to compute a shortest path for each edge on the selected path
- Can we do it faster than computing them independently?
- Yes! Almost as fast as computing only one shortest path

The VCG and the affine maximizer

Definition (Affine maximizer)

In an affine maximizer (or generalized VCG) there are constants $\lambda_i \geq 0$ (one for each player) and γ_j (one for each outcome) and the mechanism selects the outcome j which maximizes $\sum_i \lambda_i v_{ij} + \gamma_j$.

Example (Affine maximizer for 2 players, 3 outcomes)

$$\begin{array}{ccccc} v_{11} & v_{12} & v_{13} & \leftarrow & \lambda_1 \\ v_{21} & v_{22} & v_{23} & \leftarrow & \lambda_2 \\ \uparrow & \uparrow & \uparrow & & \\ \gamma_1 & \gamma_2 & \gamma_3 & & \end{array}$$

Theorem

The affine maximizers (generalized VCG) are truthful.

Is VCG good?

Pros

- VCG works in every domain!
- It is intuitive and simple

Cons

- For some problems, and in particular in combinatorial auctions, it is NP-hard to compute the outcome and payments of VCG
  - If the input is the whole $n \times k^n$ array, then the problem is computationally trivial.
  - If the input is given implicitly, then the problem can be NP-hard.
- VCG maximizes the social welfare, but it may not be good for other objectives (for example, minimizing the makespan in scheduling)

Part IV

The revelation principle

Revelation principle

Theorem (The revelation principle)

For every mechanism with dominant strategies, there is an equivalent truthful mechanism (with the same payments and outcome).

Why?

Given a non-truthful mechanism, we can design a new truthful mechanism which first simulates the lying strategies of the players and then applies the original mechanism. The players would tell the truth to this mechanism.

The revelation principle