AUTOMATED MECHANISM DESIGN

Mechanisms as VCG are very general but instantiate something specific in any specific setting

EXAMPLE: outcomes = {get, museum, other gets, burn}

HIGH TYPE: θ_1 : $u(x_1)=11$; $u(x_2)=6$; $u(x_3)=1$; $u(x_4)=0$

LOW TYPE: θ_2 : $u(x_1)=1,2$; $u(x_2)=1,1$; $u(x_3)=1$; $u(x_4)=0$

Each agent is θ_1 with probability 0.2 and θ_2 with probability 0.8

Apply VCG:

museum — WOMAN — MAN gets

MAN \ WOMAN	H	L
H	Both pay 5	Husband pay 2
L	Wife pays 2	Both pay 0,1

woman gets — museum

Expected utility: 5,136

Pro: A mechanism achieves property P in any setting that belongs to class C

Con: There is no mechanism that achieves a property in a setting

Manual MD issues: INPUT: instance

QUESTION 1: What if no mechanism covers the instance?

QUESTION 2: what if a known mechanism covers the instance? Are there other ways to do it better?

⇒ The design and actual implementation are very slow, so...

AUTOMATED MECHANISM DESIGN (AMD)

Idea: solve mechanism design as optimization problems, automatically, creating a mechanism for the specific setting at hand

Pros: higher values for the designer, allows to circumvent economic impossibility, can be used in new settings, the design burden is moved to a machine

INPUT:

- set of outcomes X
- set of agents N
- objective function $f(\Theta)$
- restrictions on the mechanism

OUTPUT
- Mechanism: it maps combinations of agents' revealed types to outcomes and specifies payments

Informally, we want truthfulness for every outcome we specify

TRUTHFULNESS CONSTRAINTS
Each player must say the truth according to some solution concept

DSIC: telling the truth is the best a player can do, independently by her types or other players' types.

$X = \{x_1, x_2, x_3, x_4\}$

	PLAYER 2: $\theta_{2.1}$	PLAYER 2: $\theta_{2.2}$
PLAYER 1: $\theta_{1.1}$	x_1	x_2
PLAYER 1: $\theta_{1.2}$	x_3	x_4

$x_k \in \{0, 1\}$

$\sum_k x_k = 1$

Constraint for P1 being $\theta_{1.1}$: ($v_i(x, \theta_i)$ is the usual value function)

$$v_1(\underbrace{f(\theta_{1.1}, \theta_{2.1})}_{\text{outcome with true declaration}}, \underbrace{\theta_{1.1}}_{\text{actual type}}) \geq v_1(\underbrace{f(\theta_{1.2}, \theta_{2.1})}_{\text{outcome with a lie}}, \underbrace{\theta_{1.1}}_{\text{actual type}}), \quad \forall \theta_{1.i}, i \neq 1.$$

In general for P1: $v_1(f(\theta_1', \theta_2), \theta_1') \geq v_1(f(\theta_1, \theta_2), \theta_1'), \ \forall \theta_1', \theta_1, \theta_2$

<u>N.B.</u>: changing the pedices, we have the constraints for P2

Maximizing function: $SW = \sum_{\theta_1} \sum_{\theta_2} \sum_i v_i(f(\theta_1, \theta_2), \theta_i)$

BNIC (for P1):
$\forall \theta_1, \theta_1'$, $\sum_{\theta_2} \omega_2(\theta_2) v_1(f(\theta_1', \theta_2), \theta_1') \geq \sum_{\theta_2} \omega_2(\theta_2) v_1(f(\theta_1', \theta_2), \theta_1)$

Since we are considering a bayesian-Nash equilibrium, each type has a probability associated to it.

Example: BNIC mechanism without payments for maximizing the sum of divorcees' utilities (DETERMINISTIC) III

WOMAN

MAN		H	L
	H	Burn	Husband
	L	Wife	Museum

RANDOMIZED MECHANISM

We define the probability that, given the types of the players, the outcomes appear.

DSIC (for P1): $\sum_{x} p(\theta_1', \theta_2, x)\, v_1(x, \theta_1') \geq \sum_{x} p(\theta_1, \theta_2, x)\, v_1(x, \theta_1'), \ \forall \theta_1', \theta_1, \theta_2$

BNIC (for P1):

$$\sum_{x} \sum_{\theta_2} w_2(\theta_2)\, p(\theta_1', \theta_2, x)\, v_1(x, \theta_1') \geq \sum_{x} \sum_{\theta_2} w_2(\theta_2)\, p(\theta_1, \theta_2, x)\, v_1(x, \theta_1'), \ \forall \theta_1', \theta_1$$

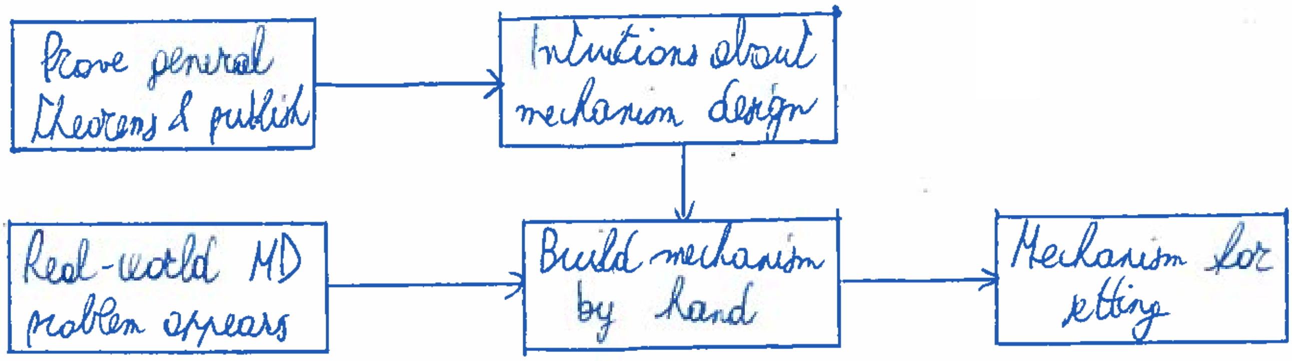
Example: BNIC mechanism without payments for maximizing the sum of divorcees' utilities

WOMAN

MAN		H	L
	H	Wife 0.55, Burn 0.45	Husband
	L	Wife	Husband 0.43, Wife 0.57

Recap: Classical MD vs Automated MD

Classical



Automated