CSCE475/875 Multiagent Systems

Handout 17: Why Mechanism Design?

October 10, 2017

(Based on Shoham and Leyton-Brown 2011)

Background

Social choice theory is nonstrategic; it takes the preferences of the agents as given, and investigates ways in which they can be aggregated.

But of course those preferences are usually not known. What you have, instead, is that the various agents *declare* their preferences, which they may do truthfully or not. Assuming the agents are self-interested, in general they will not reveal their true preferences. Since as a designer **you wish to find an optimal outcome with respect to the agents' true preferences** (e.g., electing a leader that truly reflects the agents' preferences), optimizing with respect to the declared preferences will not in general achieve the objective. (***Note***: Design the system such that each agent is motivated to tell the truth!)

Introduction

Mechanism design is a strategic version of social choice theory, which adds the assumption that **agents will behave so as to maximize their individual payoffs**. (***Note***: Rationality!)

Example: strategic voting

Consider again our babysitting example. This time, in addition to Will, Liam, and Vic you must also babysit their devious new friend, Ray. Again, you invite each child to select their favorite among the three activities—going to the video arcade (a), playing basketball (b), and going for a leisurely car ride (c). As before, you announce that you will select the activity with the highest number of votes, breaking ties alphabetically. Consider the case in which the true preferences of the kids are as follows:

$$Will: b \succ a \succ c$$

$$Liam: b \succ a \succ c$$

$$Vic: a \succ c \succ b$$

$$Ray: c \succ a \succ b$$

Will, Liam, and Vic are sweet souls who always tell you their true preferences. But little Ray, he is always figuring things out and so he goes through the following reasoning process. He prefers the most sedentary activity possible (hence his preference ordering). But he knows his friends well, and in particular he knows which activity each of them will vote for. He thus knows that if he votes for his true passion—slumping in the car for a few hours (c)—he will end up playing basketball (b). So he votes for going to the arcade (a), ensuring that this indeed is the outcome. Is there anything you can do to prevent such manipulation by little Ray?

This is where *mechanism design*, or *implementation theory*, comes in. **Mechanism design is sometimes colloquially called "inverse game theory."**

Game Theory: Given an interaction among a set of agents, how do we predict or prescribe the course of action of the various agents participating in the interaction?

Mechanism Design: Given certain **desired behaviors on the part of agents** and ask what strategic interaction among these agents might **give rise** to these behaviors.

Roughly speaking, from the technical point of view this will translate to the following.

- We will assume unknown individual preferences, and
- ask whether we can design a game such that,
- no matter what the secret preferences of the agents actually are,
- the equilibrium of the game is guaranteed to have a certain desired property or set of properties.

(***Note***: Mechanism Design == Engineering emergent behavior!)

Mechanism design is perhaps the most "computer scientific" part of game theory, since it concerns itself with **designing effective protocols for distributed systems.** The key difference from the traditional work in distributed systems is that **in the current setting the distributed elements are not necessarily cooperative**, and **must be motivated to play their part**.

For this reason one can think of mechanism design as an exercise in "incentive engineering."

*** The most famous application of mechanism design is *auction theory*, to which we devote Chapter 11.**

(***Note***: Also think: The Matrix movie …)