Internal Implementation

Ashton Anderson, Yoav Shoham, Alon Altman
Stanford University

May 2010

Outline

Introduction

We introduce a *constrained mechanism design* setting

Informal Description

- Start with a base game.
- One of the players is the "implementor".
- The implementor can make any non-negative, outcome-specific promises she desires, as long as the resulting game has a dominant strategy for all players besides herself.

Motivation

- Model mechanism designer as a player in the game
- Main question: How does the power to make binding promises (reliable contracts) affect games?

Previous Work

Monderer and Tennenholtz introduced k-implementation

A **trusted external party** interested in the outcome of a game can give outcome-specific transfers to the players

Example

G:

	L	R
U	$3, 3$	$6, 4$
D	$4, 6$	$2, 2$

G':

	L	R
U	$3 + 10 = 13, 3$	$6, 4$
D	$4, 6$	$2, 2 + 10 = 12$

Our Work

Model the external party as a player in the game (the *implementor*)

Example (Battle of the Sexes)

Consider the following game:

G:

	L	R
U	2,1	0,0
D	0,0	1,2

If the row player offers a transfer of 3 if the outcome is (D, L), then the game is transformed to:

G':

	L	R
U	2,1	0,0
D	0−3,0+3	1,2

In the transformed game, L is dominant for Player 2.

Game Theory Notation

- Games are triples (N, X, U) where N are players, X is the outcome space, and U are the payoffs. ($N = \{1, 2\}$ for today).
- $\bar{X}_i$ is the set of non-dominated strategies for player i, and $\bar{G}$ is the restriction of G to the smaller strategy space $\bar{X}$.
- i's pure safety value is $\alpha_i(G(U)) = \max_{x_i} \min_{x_{-i}} U_i(x_i, x_{-i})$.
- i's non-dominated pure safety value is $\bar{\alpha}_i(G(U)) = \alpha_i(\bar{G}(U))$

Model

Definition (Internal implementation)

Given a game G with player 1 as implementor, an *internal implementation* I_1 is a matrix Z of non-negative offers from player 1 to player 2.

Definition (Induced game)

The game G' *induced* by implementation I_1 from game $G = (X, U)$ is written $G' = I_1(G)$, where $G' = (X, U')$, and U' is specified by $U'_1 = U_1 - Z$ and $U'_2 = U_2 + Z$.

Example

G:

	C	D
C	$5, 5$	$-2, 6$
D	$6, -2$	$1, 1$

$+ Z$:

2	0
4	0

$I_1(G)$:

	C	D
C	$3, 7$	$-2, 6$
D	$2, 2$	$1, 1$

Model continued

Definition (Implemented outcome)

Let I_1 be an implementation for player 1 in game G, and let $x = (x_1, x_2) \in X$ be a pure outcome. x is the outcome *implemented* by I_1 if x_2 is a dominant strategy for player 2 in $I_1(G)$, and x_1 is player 1's best response to x_2.

In games with an implemented outcome x, the non-dominated pure safety value of every player i is simply their payoff in the implemented outcome $[\bar{\alpha}_i = U_i(x)]$.

Example

Example

G:

	C	D
C	$5, 5$	$-2, 6$
D	$6, -2$	$1, 1$

$I_1(G)$:

	C	D
C	$3-\epsilon, 6+\epsilon$	$-2, 6$
D	$2, 2$	$1, 1$

$\epsilon > 0$ small

In this example, (C, C) is the implemented outcome.

Calculation of k

To implement outcome x, the implementor has to compensate the other player for his best deviation from x.

Example

	C	D
C	$5, 5$	$-2, 6$
D	$6, -2$	$1, 1$

	C	D
C	$3 - \epsilon, 6 + \epsilon$	$-2, 6$
D	$3 - \epsilon, 1 + \epsilon$	$1, 1$

Model Details

- Only allow pure strategies
- Assume transferable utility
- For this talk, 2-player games
- Offers need to exceed best deviation by at least ϵ, but we'll simplify and assume $\epsilon \to 0$

Internal Implementation Value

The *internal implementation value (IIV)* for j is the ratio of the best value j can get from implementation to what she gets without implementation:

Definition (Internal Implementation Value)

For a game G and player j,

$$IIV_j(G) = \max_{I_j} \frac{\bar{\alpha}_j(I_j(G))}{\bar{\alpha}_j(G)}$$

For a class of games $\mathbb{G}$:

$$IIV(\mathbb{G}) = \sup_{G \in \mathbb{G},\, j \in N} IIV_j(G)$$

Internal Implementation Value

Theorem

1. *Let $\mathcal{C}$ be the class of such that the highest payoffs for all players coincide in the same outcome. Then*

$$IIV(\mathcal{C}) = \infty$$

2. *Let $\mathcal{T}$ be the class of 2×2 games. Then*

$$IIV(\mathcal{T}) = \infty$$

Internal implementation is very powerful in general.

Internal Implementation Value

Theorem

Let $\mathcal{Z}$ be the class of two-player zero-sum games. Then

$$IIV(\mathcal{Z}) = 1$$

In zero-sum games it is *no help at all*.

Sometimes your opponent can help you more

Example

G:

	L	R
U	50,100	0,0
D	101,-50	1,51

$\bar{\alpha}_1(G) = 1$ and $\bar{\alpha}_2(G) = 51$. An optimal implementation is $I_1^* = \{Z\}$ where $Z_{D,L} = 102$ and $Z = 0$ elsewhere, and the resulting payoff in the induced game $I_1^*(G)$ is $(50, 100)$. The best implementation for player 2 is the trivial implementation $I_2^* = \{\mathbf{0}\}$ where $\mathbf{0}$ is the zero matrix, and it results in the same payoff as in G. Since $100 > 51$, player 2 would benefit more from player 1's optimal implementation more than her own.

Change in Social Welfare

The social welfare after an internal implementation can be arbitrarily worse than it was before.

Example

G:

	L	R
U	$3,x-1$	$0,x$
D	$6,-1$	$1,4$

$+ Z$:

	L	R
U	$1+\epsilon$	0
D	6	0

$\rightarrow G'$:

	L	R
U	$2-\epsilon,x+\epsilon$	$0,x$
D	$0,5$	$1,4$

Summary

- We introduced a constrained mechanism design setting where the designer is a player in the game
- The implementor has the power to make outcome-specific transfers
- In general, internal implementation is powerful, but in certain games it can be useless
- The social welfare can increase and decrease arbitrarily
- Sometimes you'd rather give the opponent implementation power than have it yourself

Thanks!
Questions?