Computational Game Theory (CIS 620/OPIM 952)
Large Population Models
M. Kearns Feb 12 2003

Notation:

$\bar{p}$ - population mixed strategy

$M_i(\bar{p})$ - expected payoff to player i under $\bar{p}$

$\bar{p}[i:b]$ - $\bar{p}$ but with p_i replaced by action b

<u>Potential Games</u> (Monderer & Shapley)

Natural idea: a <u>global</u> function whose maxima correspond to N.E.

Assume n players, 2 actions each.
Say that $F:\{0,1\}^n \to \mathbb{R}$ is a <u>potential</u> for the game if:

$\forall$ pure population strategy $\bar{x} \in \{0,1\}^n$
$\forall$ player i, $\forall b \in \{0,1\}$

$M_i(\bar{x}[i:b]) - M_i(\bar{x}[i:\neg b]) > 0$

$\Leftrightarrow$ $F(\bar{x}[i:b]) - F(\bar{x}[i:\neg b]) > 0$
(F <u>independent</u> of i!)

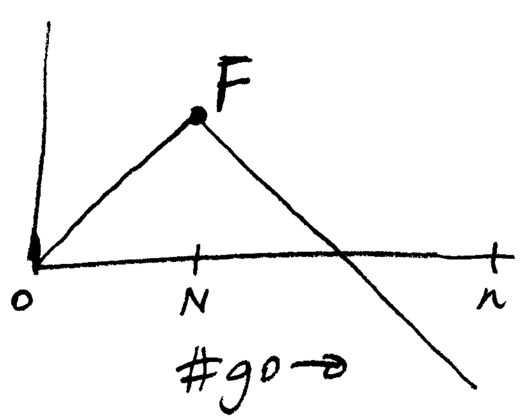
Some remarks:

- "Natural" choice $F(\bar{x}) \triangleq \sum_i M_i(\bar{x})$ may fail – improvement to M_i may be offset by reduction to M_j
- Not necessarily a "team" or cooperative game

Example: Santa Fe Bar Problem

- n potential patrons
- actions: go to bar or stay home
- stay always pays 0
- go pays $\begin{cases} 1 & \text{if} \leq N \text{ people at bar} \\ -1 & \text{else} \end{cases}$
- define $F(\bar{x}) = \begin{cases} \#\text{of "go" } \underline{\text{if}} \leq N \\ N - (\underbrace{\#go - N}_{\text{"excess"}}) & \text{else} \end{cases}$

  $(= 2N - \#go)$



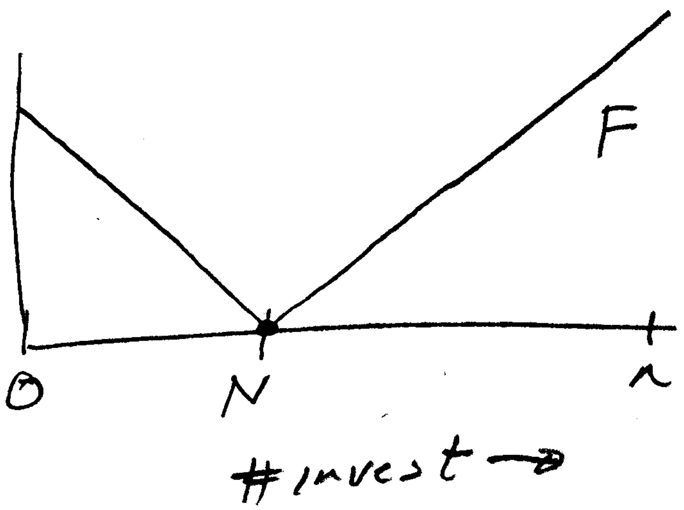
Example of a congestion game

Example: Interdependent Security

- n airlines
- actions: invest in security or don't invest
- recall model (homogeneous case)

  if #invest $\le N$, don't pays better

  if #invest $> N$, invest pays better
- define $F(\bar{x}) = |\#\text{invest} - N|$



Note: F not necc. a measure of "goodness"

Obvious Theorem:
Any maxima of F (over $\{0,1\}^n$) is a pure N.E of the game.

Corollary:
Any potential game has a pure N.E.

May miss mixed N.E.

Computation?

- Can always move closer to N.E. by incrementally modifying $\overline{x}$ to improve F
- For SF Bar and IDS, this will find N.E. quickly
- But in general, "improvement path" may traverse all of $\{0,1\}^n$ (note players may reverse their choices w/o $\overline{x}$ cycling)

Summarization Games

Assume global summarization function:

$$S: \{0,1\}^n \rightarrow [0,1]$$

Intuition: S summarizes the population action sufficient to compute payoffs

Voting example: $S(\overline{x}) = \sum_i x_i$

or weighted: $S(\overline{x}) = \sum_i w_i x_i$

Assume: $M_i(\overline{x}[i:b]) = F_i^b(S(\overline{x}[i:b]))$

(so i has F_i^0 & F_i^b)

Assume the F_i^b are continuous & bounded derivatives (bound $= \rho$)

$\Rightarrow$ Assume S has <u>bounded</u> <u>influence</u>:

$$\tau_i \triangleq \max_{\bar{x} \in \{0,1\}^n} \{ |S(\bar{x}[i:0]) - S(\bar{x}[i:1])| \}$$

$$\tau \triangleq \max_i \{ \tau_i \}$$

<u>Remarks</u>

- Though F_i^0 & F_i^1 are smooth, can <u>differ</u> greatly $\Rightarrow$ i's action has <u>unbounded</u> impact on <u>i's</u> payoff
- But i's action has only influence $\leq \tau$ on S $\Rightarrow$ i has limited impact on payoffs of <u>others</u>
- Smoothness of F_i^b prevents "readout" of bits from S
- <u>Crucial</u>: often expect τ to <u>decay</u> with n! E.g. voting: $\tau = 1/n$

Computational Result:

ρ = bound on derivatives of F_i^b

τ = bound on influence of S

ε = input parameter

assume unit cost of evaluating S & F_i^b

An algorithm whose running time is poly in n, $1/\varepsilon$ and ρ;

that outputs a pure $O(\varepsilon + \tau\rho)$-N.E.

(Algorithm & analysis: see K. & Mansour paper)

Example: Airline Security, Heterogeneous Case

Let x_i be action of player i

$$x_i = \begin{cases} 0 & \text{invest} \\ 1 & \text{don't} \end{cases}$$

Then write

$$M_i(\bar{x}) = (1-x_i)C + x_i p_i(-L) + (1-x_i p_i)\Big[\underbrace{1 - \prod_{j \neq i}\left(1 - \frac{x_j q_j}{n-1}\right)}_{\text{this is the "global" info } i \text{ needs to compute payoffs}}\Big](-L)$$

this is the "global" info i needs to compute payoffs

Define

$$S(\bar{x}) = \prod_{j=1}^{n}\left(1 - \frac{x_j q_j}{n-1}\right)$$

Then

$$\frac{S(\bar{x})}{\left(1 - \frac{x_i q_i}{n-1}\right)} = \prod_{j \neq i}\left(1 - \frac{x_j q_j}{n-1}\right)$$

$$\text{Influence } \tau_i = \prod_{j \neq i}\left(1 - \frac{x_j q_j}{n-1}\right) - \left(1 - \frac{q_i}{n-1}\right)\prod_{j \neq i}\left(1 - \frac{x_j q_j}{n-1}\right)$$

$$\leq 1 - \left(1 - \frac{q_i}{n-1}\right) = \frac{q_i}{n-1}$$