Computational Game Theory (CIS 620/OPIM 952) Instructor: M. Kearns
Luis Ortiz January 30, 2003

NashProp

- A heuristic for computing NE in graphical games with arbitrary graphs.

- A generalization of TreeNash.

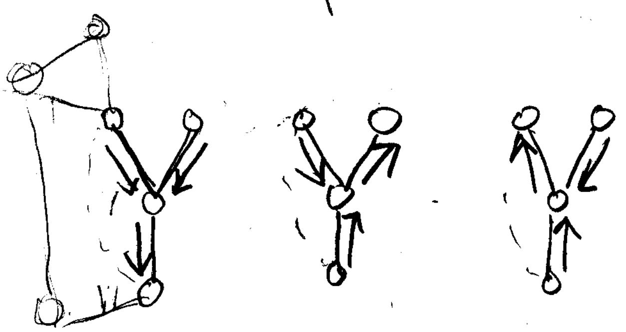
Basic idea

- Each player neighborhood looks locally like a tree.

- So, apply same "table" computation (message-passing operation) as in TreeNash (the algorithm for trees).

- But, there is no global root (in arbitrary graphs): From the player's local perspective, "the player doesn't know where the root is" [actually, there might not be any!]

- So, each players "sends table message" to each neighbor as if "root" could be reached through that neighbor.
  [process can be done distributedly and asynchronously]

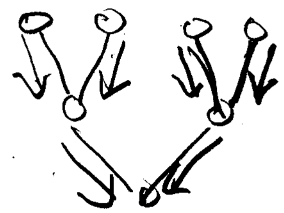


- Two phases: table-passing phase, assignment-passing phase.

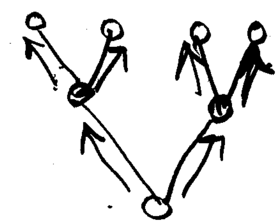
Recall: TreeNash

① Only propagate tables "down" the tree to root; then only propagate solutions "up" the tree to leaves

propagate: tables



assignments



② No "initialization" needed: use "leaves" to initialize.

NashProp

① Propagate tables in rounds; each player sends a table to every neighbor.

TablePropagation "rule" is the same as in TreeNash, except that it is executed for each neighbor independently.

At each round t,

$\forall (W,V) \in E$, $\forall (\omega, v) \in [0,1]^2$,

edge set of graph.

size of V's neighborhood

$T^t(\omega, v) = 1$ iff $\exists$ a witness $\vec{u} \in [0,1]^{k-2}$, s.t.

1. $T^{t-1}(v, u_i) = 1$, $\forall i = 1, \ldots, k-2$
2. $V = v$ is a best response to $W = \omega$, $\vec{U} = \vec{u}$

② How do we start? (i.e. What about $t=1$?)

Initialize to "full" tables $\left[\text{i.e., } \forall (W,V) \in E, \forall (\omega, v) \in [0,1]^2, \; T^0(\omega, v) = 1\right]$

Intuition: Lacking any initial knowledge, a player "believes" any strategy is a best response to any other strategy.

Analysis of Abstract "table-passing phase"

Remarks:

- $\forall (W,V) \in E,\ \forall (\omega, v) \in [0,1]^2,$

$$T^{t-1}(\omega, v) = 0 \Rightarrow T^{t}(\omega, v) = 0$$

$$\{(\omega,v) \in [0,1]^2 : T^{t-1}(\omega, v) = 1\} \supseteq \{(\omega, v) \in [0,1]^2 : T^{t}(\omega, v) = 1\}$$

- So, "tables" converge! $\left[\forall (W,V) \in E,\ \lim_{t\to\infty} \{(\omega,v)\in[0,1]^2 : T^{t}(\omega,v)=1\} = \{(\omega,v) : T^{*}(\omega,v)=1\} \text{ exists!}\right]$

Defn: Given a graphical game, we say a set of "tables"

$\{T_{WV} : [0,1]^2 \to \{0,1\},\ \forall (W,V) \in E\}$ for the game is

balanced if

$\forall (W,V) \in E,\ \forall (\omega, v) \in [0,1]^2,\ T(\omega, v) = 1$ iff $\exists \vec{u} \in [0,1]^{k-2}$, an assignment to neighbors $\vec{U}$ of V other than W, s.t.

1. $T(v, u_i) = 1,\ \forall i = 1, \ldots, k-2$
2. $V = v$ is B.R. to $W = \omega, \vec{U} = \vec{u}$.

Defn: A joint mixed strategy $\vec{p} \in [0,1]^{n}$ (# of players) is **consistent** with balanced tables $\{T_{WV}\}$ if $\forall (W,V) \in E,\ T_{WV}(p_W, p_V) = 1$.

Observation: The limit tables $\{T^{*}_{WV}\}$ are **balanced**.

* A characterization of Nash equilibria in a graphical game:
  A mixed strategy $\vec{p}$ is a NE for the graphical game GG iff $\vec{p}$ is consistent with the limit (balanced) tables for GG.

Approximate NE:

- As in ApproxTreeNash, discretize each player's mixed strategy space.

  $v \in \mathbb{I} = \{0, \tau, 2\tau, \ldots, 1\}$

- Griding size sufficiency results hold for games with arbitrary graph:
  So, set
  $$\tau = \frac{\varepsilon}{2k} \quad \text{as before.}$$

- Table size $\lceil \frac{1}{\tau} \rceil^2 = O\left(\left(\frac{2k}{\varepsilon}\right)^2\right)$ [Poly in game representation size]

  Running time (per round, per player, per neighbor):
  $$O\left(\left(2\lceil \tfrac{1}{k} \rceil\right)^k\right) = O\left(\left(\frac{4k}{\varepsilon}\right)^k\right)$$
  [For k, s.t. $k \log k = O(\log n)$, poly in representation size]

  k: max neighborhood size. n: # of players

- Convergence result ⇒ Table-passing phase converges in a finite # of rounds, So total running time for this phase:
  $$O\Big(\underbrace{|E| \left(\tfrac{2k}{\varepsilon}\right)^2}_{\text{max. \# of rounds}} \underbrace{\left(\tfrac{4k}{\varepsilon}\right)^k}_{\text{each table computation per round}} \underbrace{|E|}_{\text{\# of tables [computed per round]}}\Big) = O\left(n^2 k^{k+4} 2^{3k} \left(\tfrac{1}{\varepsilon}\right)^{k+2}\right)$$
  [for k, s.t. $k \log(k)$, poly in model size]

- at each round, at least one table entry changes [from 1 to 0]
- We have $|E|$ tables, each of size $O(\frac{1}{\tau^2})$.

  Recall, G undirected, so $(W,V) \in E \Rightarrow (V,W) \in E$

  [Easy to modify to directed case].

Observation: Lemma 6 of KLS (union of rectangles representation) is general, So it applies here. Hence, we can do table-message passing exactly for 2-action games, BUT

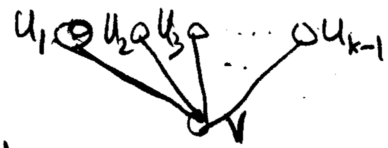
- it might not converge in finite time!
- Table representation can grow exponentially with the # of rounds!

Assignment-passing phase

Basic Idea: Table-passing phase might have left us with a significantly smaller search space!

[See Accompanying PowerPoint presentation for an example of ideal behavior].

- A "generalization" of "upstream pass" in TreeNash BUT with a "wrinkle".



- Defn: The projection set P of player V is

  $\forall v \in [0,1]$, $P_V(v)=1$ iff $\exists \vec{u} \in [0,1]^{k-1}$, s.t.

  - $T(v,u_i)=1$, $\forall i=1,..,k-1$
  - $V=v$ is BR to $\vec{U}=\vec{u}$.

[Alternative definitions exist]

Algorithm Sketch [for Assignment-passing phase]

Initialization: • Pick arbitrary player V

• Select a ~~mixed~~ fixed strategy $v \in P_V(v)=1$ [non-deterministically]

Iterate: At each round,

In some sequential order over each player V

• if V has been set to v:

∘ if V has unset neighbors,

- set unset neighbors s.t. resulting assignment to all neighbors is a <u>witness</u>

[i.e. if $\vec{u}$ is assignment to all neighbors,

• $T(v, u_i) = 1$, $\forall i = 1, \ldots, k-1$

• $V = v$ is B.R. to $\vec{U} = \vec{u}$]

- if not possible, <u>backtrack</u>!

∘ else: check for "consistency"

· <u>backtrack</u> if inconsistent neighborhood assignment

<u>Remarks</u>: • For tree graphical games, no backtracking necessary!

• In general, need to backtrack to take care of inconsistencies

• Assignment-passing phase is worst case exponential in # of players.

[See Accompanying PowerPoint presentation for experimental results]