Graphical Games

Michael Kearns

May 23, 2011

M.Kearns, M. Littman, and S.Singh

Graphical Models for Game Theory

Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence

Multiplayer Games

Players $i = 1...n$

each with a finite set of **pure** strategies A_i

for simplicity we will assume that the action chosen by player i
$a_i \in \{0, 1\}$, (binary actions)

Normal Form

The payoffs to player i are given by a matrix M_i

This matrix is indexed by a joint action $\vec{a} \in \{0, 1\}^n$

The value $M_i(\vec{a})$ ($wlog \in [0, 1]$) is the payoff for player i if players play the joint action $\vec{a}$

Normal Form (mixed strategies)

In our binary setting a mixed strategy for player i is given by the probability $p_i \in [0, 1]$ that the player will play 0

The expected payoff to player i from the joint mixed strategy $\vec{p}$ is then defined as

$$M_i(\vec{p}) = E_{\vec{a} \sim \vec{p}}[M_i(\vec{a})]$$

Nash Equilibrium

$\vec{p}[i : \bar{p}_i]$ denotes the joint mixed strategy which is the same as $\vec{p}$ except that player i deviates to $\bar{p}_i$

Then $\vec{p}$ is a NE for the game iff $\forall i, \bar{p}_i \in [0,1]$

$$M_i(\vec{p}) \qquad \geq M_i(\vec{p}[i : \bar{p}_i])$$

ϵ-Nash Equilibrium

$\vec{p}[i : \bar{p}_i]$ denotes the joint mixed strategy which is the same as $\vec{p}$ except that player i deviates to $\bar{p}_i$

Then $\vec{p}$ is a ϵ-NE for the game iff $\forall i, \bar{p}_i \in [0, 1]$

$$M_i(\vec{p}) + \epsilon \geq M_i(\vec{p}[i : \bar{p}_i])$$

Issues with Normal Form

Assuming n players and 2 actions, as we have here, leads to the need for : n matrices M_i (one for each player) each of size 2^n

Issues with Normal Form

Assuming n players and 2 actions, as we have here, leads to the need for : n matrices M_i (one for each player) each of size 2^n

Furthermore tabular form fails to capture structure inherently present in the game

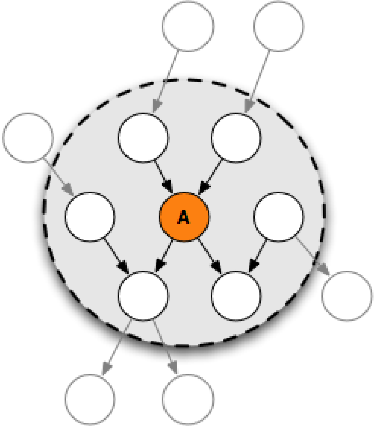
Structure in Games

It is assumed that the payoff M_i for player i is a function of all the components $a_j, (j = 1....n)$ in the joint action vector $\vec{a}$

However the payoff for player i may be dependent only on the actions of a subset of players $N(i)$

$\longrightarrow$ conditional independence payoff assumption

Cond. Ind. Payoff



Graphical Games

An undirected graph G

- n vertices one for each player i
- $N(i)$ is the neighborhood of player i
  $\longrightarrow$ there is an edge $(i, j), \forall j \in N(i)$

Graphical Games Payoff Representation

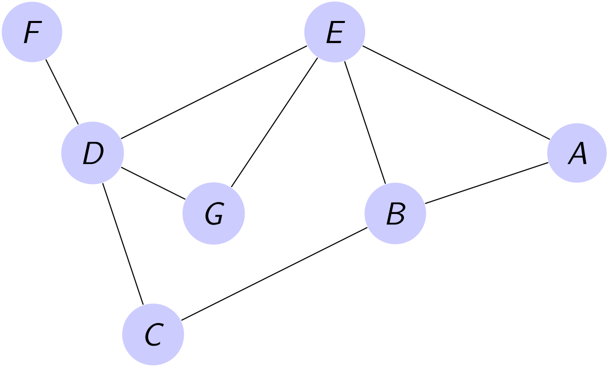
Local payoff matrix for $\bar{M}_i$ depends only on the actions taken by players in $N(i)$

$$M_i(\vec{a}) = \bar{M}_i(\vec{a}[N(i)])$$

Representation Complexity

$|N(i)|$ is the degree of local interaction for node i

The maximum k over the graph $k = \max_i |N(i)|$ defines the complexity of the representation $O(n2^k)$



Why this is cool

- **Computational** Specific topological properties can be used to yield effficient algorithms for finding Nash equilibria
- **Structural**
- **Interdisciplinary**

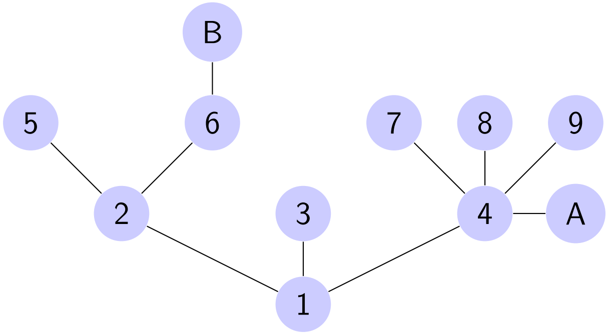
Why this is cool

- **Computational**
- **Structural** Provide a tool for examining whether the topology of G implies structural properties of the equilibria
- **Interdisciplinary**

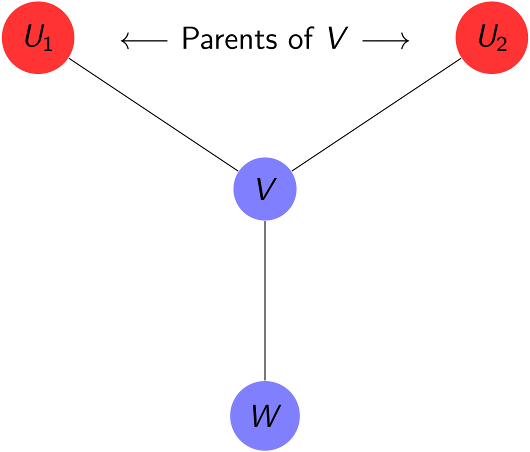
Why this is cool

- **Computational**
- **Structural**
- **Interdisciplinary** Allow the use of powerful methods from different fields (e.g. machine learning , statistics)

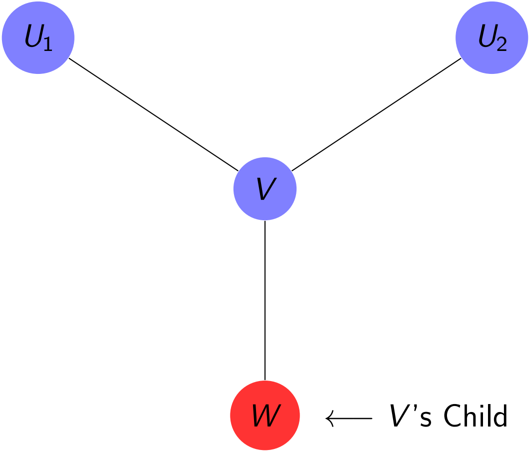
Tree Graphical Games



Tree Games

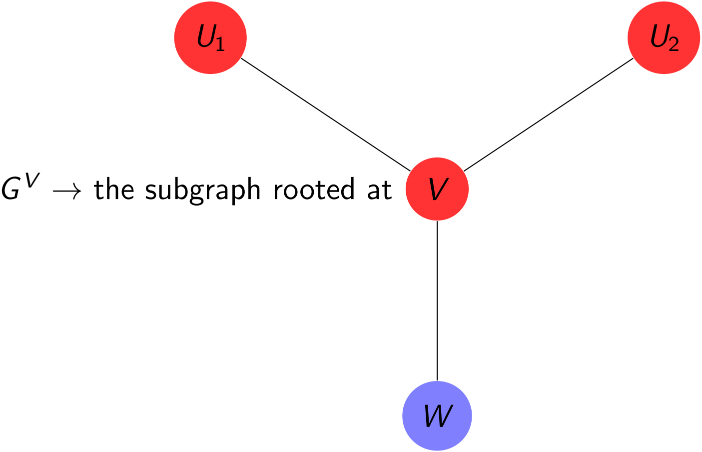


Tree Games

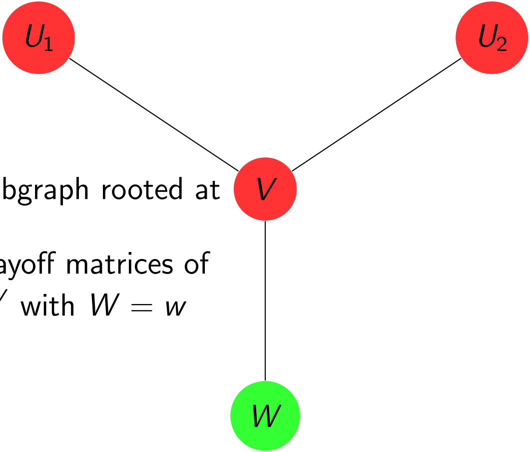


$\longleftarrow$ V's Child

Tree Games



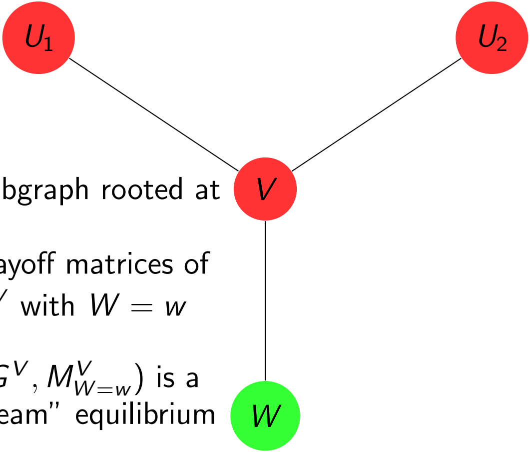
Tree Games



$G^V \rightarrow$ the subgraph rooted at V

$M^V_{W=w} \rightarrow$ payoff matrices of G^V with $W = w$

Tree Games



$G^V \rightarrow$ the subgraph rooted at V

$M^V_{W=w} \rightarrow$ payoff matrices of G^V with $W = w$

A NE for $(G^V, M^V_{W=w})$ is a cond. "upstream" equilibrium

TreeNash Algorithm

Two pass algorithm.

- **Downstream**
- **Upstream**

TreeNash Algorithm

Two pass algorithm.

- **Downstream** Calculates cond. equilibria and passes "witness" lists down the tree
- **Upstream**

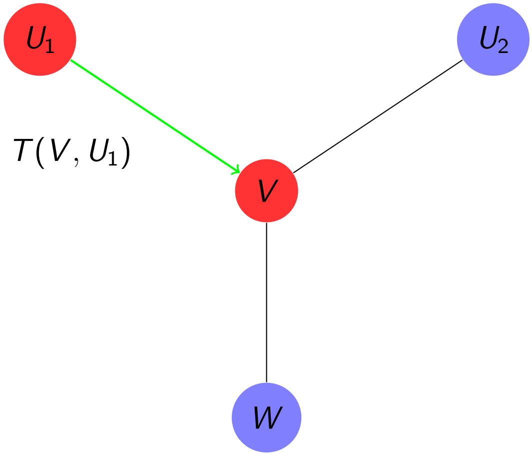
TreeNash Algorithm

Two pass algorithm.

- **Downstream**
- **Upstream** Selects "witness" lists going from the root to the leaves and calculates a NE

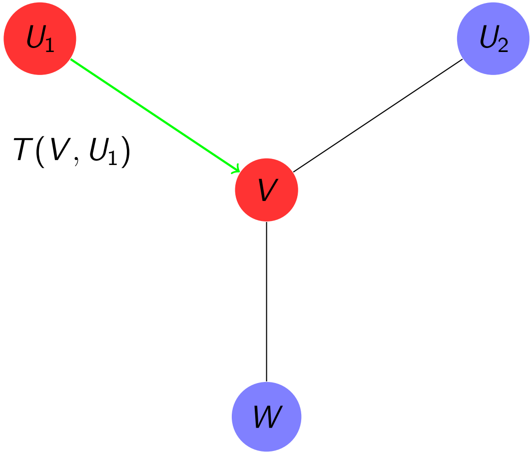
TreeNash Downstream

Each parent sends a table $T(V, U_i)$ such that $T(v, u_i) = 1$ iff there exists a NE in $(G^U, M^U_{V=v})$ for which $U_i = u_i$



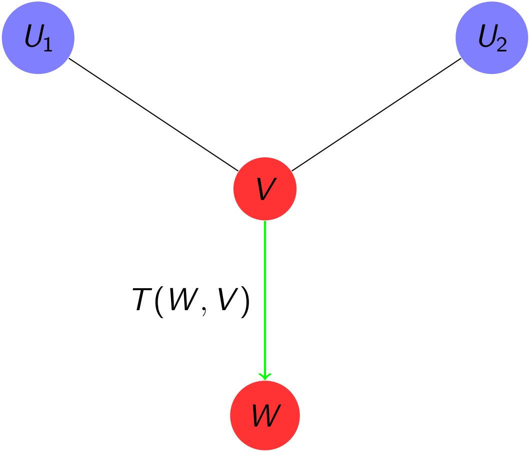
TreeNash Downstream

If U_i is a leaf then $T(v, u_i) = 1$ iff u_i is a best response to v



TreeNash Downstream

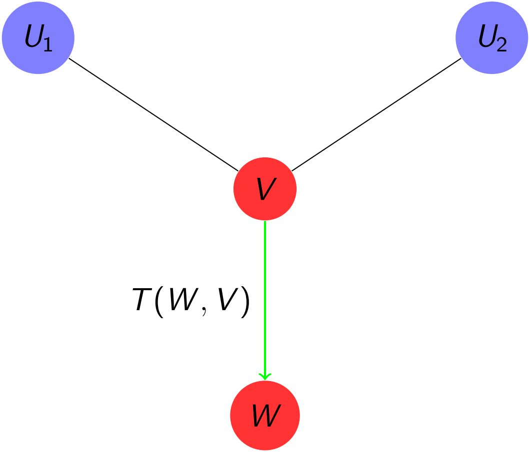
$T(w, v) = 1$ iff v is a best response to w and $T(v, u_i) = 1, \forall i$



TreeNash Downstream

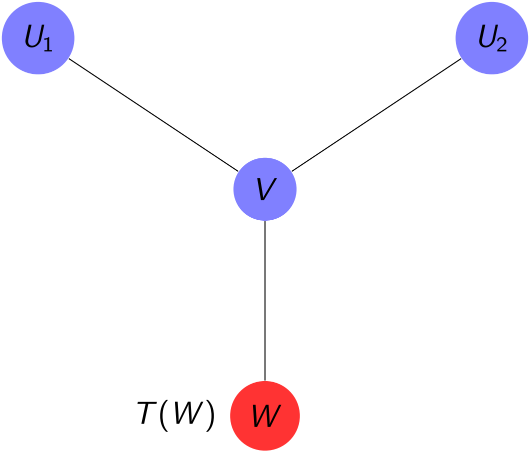
$T(w, v) = 1$ iff v is a best response to w and $T(v, u_i) = 1, \forall i$

$\vec{u}$ is then added to the "witness" list of $T(w, v)$



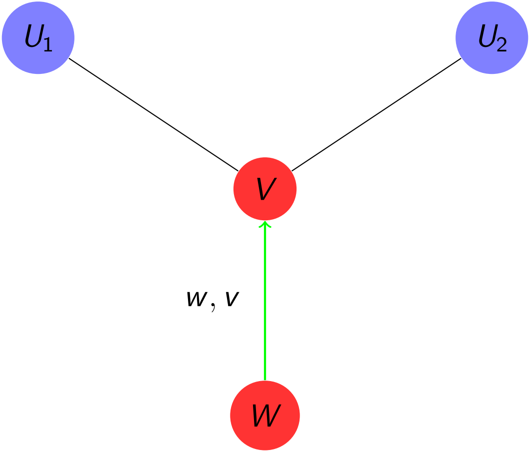
TreeNash Downstream

At the root, the algorithm computes the table $T(W)$ where $T(w) = 1$ iff w is a best response to $\vec{v}$ and $T(w, v_i) = 1, \forall i$



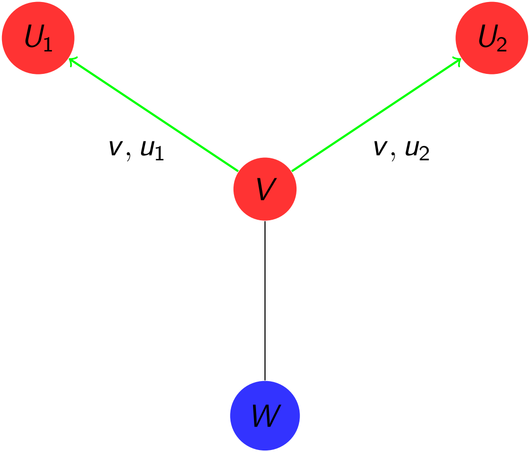
TreeNash Upstream

The algorithm chooses a value w for which $T(w) = 1$, then passes this value plus the witness v to its parent (instructing it to “play“ v)



TreeNash Upstream

V receives w, v and sends the witness $\vec{u}$ of $T(w, v) = 1$ to its parents



TreeNash (A slight issue)

The actions $(u, v, w ...)$ are continuous variables $\rightarrow$ Can $T(w, v)$ be represented compactly?

Approximate TreeNash

Discretization of the action space

Player i can now only play action $q_i \in \{0, \tau, 2\tau, ..., 1\}$

Approximate TreeNash

Discretization of the action space

Player i can now only play action $q_i \in \{0, \tau, 2\tau, ..., 1\}$

Algorithm takes an extra input parameter ϵ

At each node the ϵ-best response is computed ($\tau = O(\epsilon/d)$)

Approximate TreeNash

Discretization of the action space

Player i can now only play action $q_i \in \{0, \tau, 2\tau, ..., 1\}$

Algorithm takes an extra input parameter ϵ

At each node the ϵ-best response is computed ($\tau = O(\epsilon/d)$)

Theorem Approximate TreeNash computes a ϵ-NE for the game (G, M) in time polynomial in the representation of (G, M)

Exact TreeNash ?

Exact TreeNash

Yes! However its complexity is exponential in the number of vertices of G

Exact TreeNash

Yes! However its complexity is exponential in the number of vertices of G

Computing an exact equilibrium in time polynomial in the size of the tree remains an open issue

The End