Multiplayer Pushdown Games

Anil Seth
IIT Kanpur
Multiplayer Games we Consider

- These games are played on graphs (finite or infinite)
- Generalize two player infinite games.
- Any number of players are allowed.
- These are sequential (or turn based).
  Each position in the graph belongs to a unique player.
• Non-zero sum games – each player has an independent winning condition.
• Parity winning conditions for each player.
• Solution - strategies of players in some equilibrium.
• What is a strategy:
• Choice of a move for a player at each node in the game tree. (Game tree is unfolding of the game graph from starting vertex)
Equilibrium Notions

• Nash Equilibrium (NE): any player deviating unilaterally does not improve his gain.

• Subgame Perfect Equilibrium (SPE): Restrictions of the chosen strategies from each position in the game tree form a NE.
Previous Work

• (Chatterjee, Majumdar and Jurdziński  CSL 2004) showed that NE always exists in such games.

Their proof Idea:

• Consider player-\textit{i} against the coalition of the other players. Let $W_i$ be the winning region for player-\textit{i} in this game.

• Player-\textit{i} plays winning strategy in $W_i$ and a fixed strategy outside $W_i$. Any deviation of a player from his strategy is punished jointly by the rest of the players. These strategies are easily seen to form a NE.
• The strategies of Chatterjee et al. do not form SPE.
• Reason: Joint punishing strategy of a player-j, may sacrifice his own winning.
• Grädel and Ummels improved this result by showing existence of a SPE.
• They show some results on the existence of SPE in finite or infinite game graphs and algorithms for finding a SPE on finite game graphs.
Our Results

• We consider configuration graphs of pushdown systems (PDS) as our game graphs.
• We give effective procedure for finding SPE in these games.
• The SPE strategies are presented as pushdown automata executing them.
Our Techniques

• We slightly modify arguments of Grädel and Ummels to establish the existence of SPE and appropriate strategies in game G over a PDS.

• Use the theory of two way automata to find strategies effectively.
Definitions and Notations

- PDS $M$ is $(Q, \Gamma, q_0, \delta)$

- Game $G=(\pi, (Q_i)_{i \in \pi}, E, (\Omega_i)_{i \in \pi})$, where
  - $\pi=\{1, \ldots, n\}$ is a set of players
  - $(Q_i)_{i \in \pi}$ is a partition of $Q$. Configurations with state in $Q_i$ belong to player $i$.
  - $V = Q \times \Gamma^*$, the set of vertices of $G$, is the configurations of $M$.
  - $E$, the edge relation, is given by transition function of $M$
- $\Omega_i : Q \rightarrow \{1, \ldots, P_i \}$ is a priority function for player $i$.

- We assume the game to start from the initial configuration of $M$. 
Existence Lemma

For all $\sigma_i, i \in \pi$, there exist

- $W_i \subseteq V$

- Uniform and history free strategies $\sigma_i$ for player $i$ over $W_i$

- Uniform and history free joint strategies $\sigma_{-i}$ of players $\pi - \{i\}$ over region $V - W_i$
Such that

- The strategies $\sigma_{-i}$ of players $j, j \in \pi - \{i\}$, over region $W_j$ agree with $\sigma_j$.
- In any play starting in $W_i$, where all players-$j$ play according to $\sigma_j$ over $W_j$, player $i$ wins.
- In any play starting in $V-W_i$, where all players-$j$, $j \neq i$, play according to $\sigma_{-i}$, player $i$ loses.
Proof Idea of Existence Lemma

• Starting from the configuration graph of M, we define a transfinite sequence of graphs \( (G_\alpha)_{\alpha \in \text{Ord}} \) with the set of edges monotonically decreasing.

• Given \( G_\alpha \), for each i, we fix a uniform and history free winning strategy in the winning region \( W_i^\alpha \) of player-i in \( G_\alpha \) against coalition of the other players.

• \( G_\alpha + 1 \) is obtained by deleting from positions of player-i in \( W_i^\alpha \), all except those edges which are dictated by the winning strategy.
• The sequence finally stabilizes at an ordinal $\lambda$, s.t. $G_\lambda = G_\lambda \text{ plus } 1$

• Let $W_i$ be the winning region of player-$i$ in $G_\lambda$. Let $\sigma_i$ be the uniform and history free winning strategy of player-$i$ over $W_i$.

• Let $\sigma_{-i}$ be the uniform and history free joint winning strategy of rest of the players against player $i$ in $G_\lambda$ over the region $V - W_i$.

• These strategies can be shown to satisfy the required conditions.
Defining SPE strategies from σ’s

• Extend strategies $\sigma_i$ by fixing any move from vertices outside $W_i$.

• All players play these extended strategies unless some player $j$ deviates (in a way that the vertex played is outside $W_j$). In the latter case, if the last player so deviating is $k$ then players in $\pi - \{k\}$ play $\sigma_{-i}$.

• It can be shown that these strategies form a SPE.
To compute $\sigma$’s effectively

- We use the theory of two way tree automata to do this.

- Consider a $|\Gamma|$ ary tree. It naturally encodes stack configurations of $M$.

- We consider tree alphabet which can encode strategies like $\sigma$’s locally on tree nodes.
• The tree alphabet $\Sigma$ is
  \[ \Gamma \times [Q \rightarrow 2^\Pi] \times [Q \rightarrow \text{moves}]^\Pi \]
  If node $u$ is labeled with $(\gamma, f, (h_1, ..., h_\pi))$ then
• $\gamma$ stores the direction of the node $u$ w.r.t. its parent
• $f$ is a function $Q \rightarrow 2^\Pi$ s.t. $j \in f(q)$ iff $(q, u) \in W_j$.
• $\sigma_i(q, u) = h_i(q)$ and
  for $(q, u) \notin W_j$, $(\sigma_j)_i(q, u) = h_i(q)$. 
• We design a two way tree automaton $B$ which checks if the tree labels indeed code strategies satisfying conditions for $\sigma$’s as in existence lemma.

• Our tree automata also store the state of $M$. So a automaton at a node of the tree represents a configuration.

• A two way automaton from a node $v$ can move to its children nodes, parent nodes or remain stationary.
• This allows a path of the two way tree automaton to represent a PDS play.
• We design automaton $A$ which accepts a tree if it does not code $\sigma$’s with the desired properties. Automaton $B$ is obtained as a complementation of automaton $A$.
• The automaton $A$ is designed as follows. $A$ starts from the root of the tree and nondeterministically chooses any node $v$ of the tree and guesses a state $q$ of $M$. 
• Automaton A nondeterministically guesses if there is a **bad play starting** \((q,v)\).

• There are two kinds of bad plays. One where \((q,v) \in W_j\) and there is a **losing play for player\(-\)j** from \((q,v)\) (in which any player \(k\) play \(\sigma_k\) over \(W_k\)).

• Second where \((q,v) \notin W_j\) and there is a **winning play for player\(-\)j** (in which any player \(k\) plays with \((\sigma_{-j})_k\))

• A accepts iff it finds a bad play.
• The no. of states of $A$ is $O(|Q|)$.
• We convert $A$ to an equivalent one way automaton $C$ and then complement $C$ to obtain $B$.
• As both the above conversions take exponential time, size of $B$ is $O(|Q|^{2^{|Q|^k}})$.
• We check for emptiness of the tree automaton $B$ and extract a regular tree from it.
• This regular tree allows us to define a pushdown strategy for each player which collectively form a SPE.
Conclusions and Future Directions

• We showed existence of SPE in PDS games and how to compute one effectively.
We believe that some improvements are possible.

• It may be possible to get finite state strategies for SPE instead of pushdown strategies.
• We have not examined complexity of computing a SPE critically, it may not be optimal
• Algorithm to find best SPE should also be possible.