Nash Prop : An Approximate Nash Equilibrium Computation in Large Loopy Non Cooperative games

Shreya Kadambi and Nitisha Pandharpurkar
New York University

Introduction

The Nash Equilibrium represents learning the set of strategies for n players such that deviating from the Nash equilibrium will not increase the expected utility for every player. For computing the Nash equilibrium on loopy graphical games, the computation time is exponential in literature. We are implementing the NashProp algorithm which is optimum for computing approximate Nash equilibrium in loopy graphs. We have seen the case study of the graphical Hotelling's game.

Computing Nash Equilibrium on Loopy n-player Non-Cooperative Games

In graphical games in which the underlying graph is a tree computation of an exact Nash equilibrium is shown to be performed in exponential time, while an ϵ-Nash equilibrium is computed in a polynomial linear to the size of the graph. For general graphs or loopy graphs, if we convert the graph to a tree based network via triangulation, we can run tree based algorithms on the resulting junction tree. This method takes exponential time. The most optimal solution is the NashProp algorithm, and it runs in a polynomial time on a reduced subspace.

Graphical Game Representation of the Hotelling's Problem

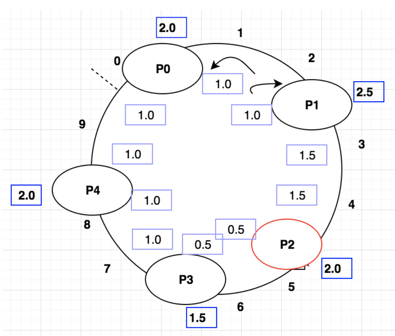


Figure: Case 1

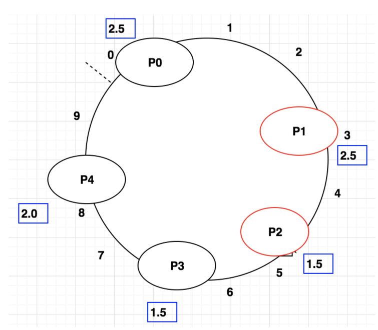


Figure: Case 2

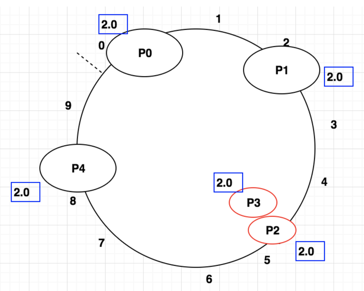


Figure: Case 3

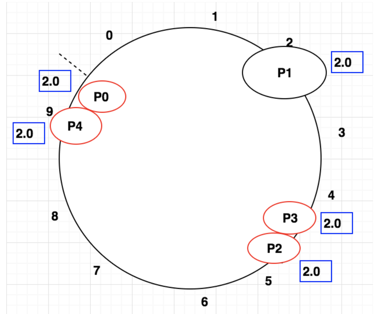


Figure: Case 4

Modification: We are using the Hotelling's game in a circle instead of a street, to make the game loopy.
Let us imagine these shops to be arranged around a circular park. This makes the graph loopy. Each shop has 3 strategies, Cooperating, Defecting to the Left and Defecting to the Right. There is no clear strategy because every strategy depends on whether the neighbor defects, and in what direction. So, in this case there is no dominant strategy. The shop picks it's strategy based on the belief it has for the shops next to it defecting.
Nash Equilibrium: A strategy profile S = $(s_1, ..., s_n)$ is a Nash equilibrium if for every i, s_i is a best response toS_i , i.e., no agent can do better by unilaterally changing his/her strategy
We can see that the Nash equilibrium in cases 3 and 4. **Nash Equilibrium:** $s_i(a_j)$= probability that action a_j will be played under mixed strategy s_i

Graphical Representation

- An n-player, two-action2 game is defined by a set of n matrices $M_i(1 <= i <= n)$, each with n indices. The entry $M_i(x_1, ..., x_n) = M_i(\vec{x})$specifies the payoff to player i when the joint action of the n players is $\vec{x} \in 0, 1^{n3}$.
- For any joint mixed strategy, given by a product distribution $\vec{p}$, we define the expected payoff to player i as $M_i(\vec{p}) = E_{\vec{x}\sim\vec{p}}[M_i(\vec{x})]$ where $\vec{x} \sim \vec{p}$ indicates that each x_j is 0 with probability p_j and 1 with probability $1 - p_j$.

Approximate Nash Equilibrium: An ϵ-Nash equilibrium is a mixed strategy $\vec{p}$ such that for any player i, and for any value $p'_i \in [0, 1]$, $M_i(\vec{p}) + \epsilon >= M_i(\vec{p}[i : p'_i])$

Nash Prop

- Nash Prop is based upon the message passing algorithm on General graphs with loops.
- The Nodes in a Nash Prop unlike in inference models represent Players
- An Edge between any nodes represent the influence of one player over the other.
- If a players action does not directly influence the other players no edge is drawn between them
- For the purpose of project we have implemented the algorithm on bipartite factor graphs.
- Analogous to the beliefs in inference models A player passes in his message his belief of the nash equilibria strategy the other player should follow.
- Advantage of using loopy belief/Nash Prop is to compute local Nash strategy for player U given the evidence V_i for $i \in N(U)$
- Complexity of Algorithm thus is only dependent on the maximum degree of any node in the graph. As local expected payoffs are being computed and passed to the immediate neighbours.

Model architecture

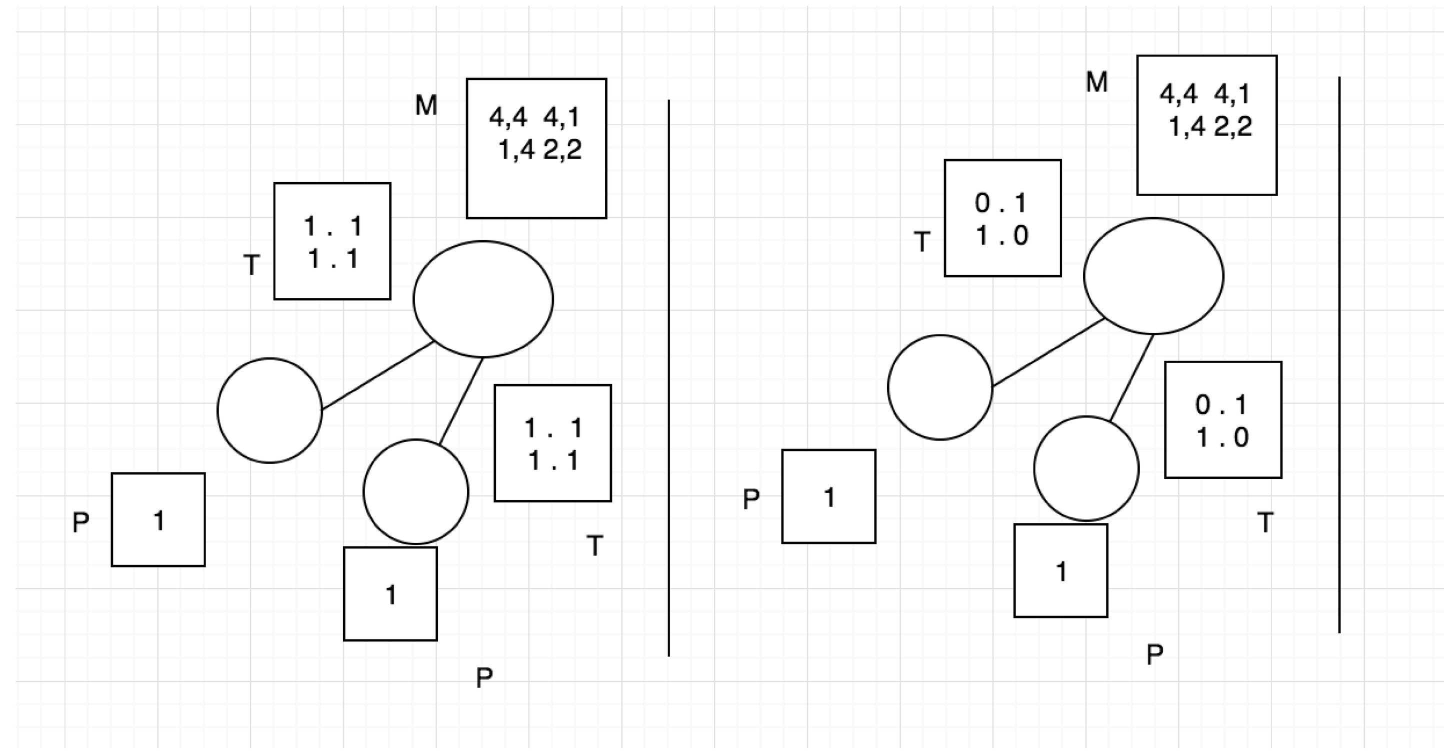


Figure: propagate conditional nash equilibria Where $\mathcal{T}(\sqsupseteq, \sqsubseteq)$ represents the table passed from $\mathcal{W}$ to $\mathcal{V}$

'**Hotellings**' problem has been represented as an 'n' player 3 action game. It has been represented using any generic undirected graphs as $(\mathcal{G},\mathcal{M})$. Where $\mathcal{G}$ represents an undirected graph with 'n' nodes for each player $\mathcal{M}$ represents the payoff matrix consising of 'n' different '$\mathcal{M}_i$' matrices indexed by the player.
For each of the strategy for $x \in [0, 1]^{k+1}$,$\mathcal{M}_V(x)$ denotes the play off when player choses a strategy in x . Given below is the sampling choice of the mixed strategies for epsilon nash equilibria

$$\tau = \epsilon/(2^{k+1} * k \log k) \qquad (1)$$

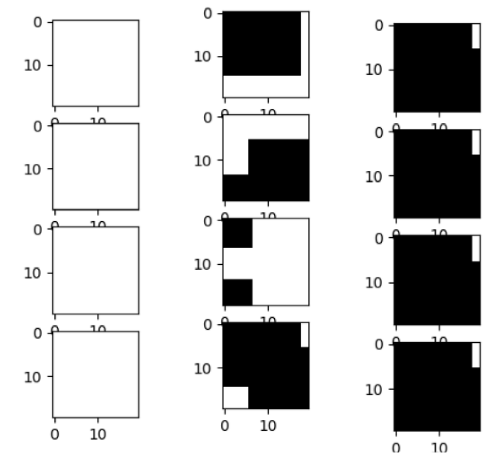
where ϵ is allowed relaxation on the maximum expected payoff and 'k' is the maximum degree of any node in the graph

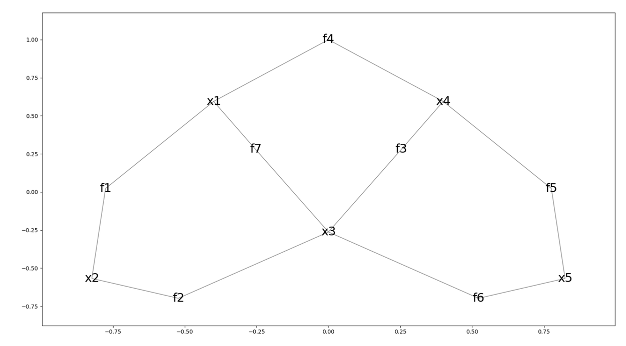
- **Data Structure**
  - ▷ We Initialize each Node with a binary valued table $P(u) \in [0, 1]$. These values correspond to the possible strategies that as seen by the evidence U_k its neighbours in the graph
  - ▷ The factors are initialized with the corresponding pay off matrix of size MxN where M is the size of the Action space for Player 1 and N is the size of action space for player2
  - ▷ The messages transmitted T(w,v) of size $1/\tau x 1/\tau$ where τ is the sampling interval of the strategy. Where each value of the table corresponds to the likelihood of the join strategy (w,u) represented as either 0 or 1.
- **Message/'Table' passing phase**
  - ▷ The Message passing phase passes a binary table to its immediate neighbours based on what it thinks could be the nash equilibria given evidence of its immediate neighbours.
  - ▷ The tables are indexed by the possible strategy w, v
- **Reducing the search space using projection matrix**
  - ▷ With every pass more entries if the table are replaced with 0s. Thus reducing the search space
  - ▷ Assignment of strategies happen at every variable node by computing the projection matrix (P*) which is the cross product of all the incoming tables at the variable Node.
- **Sampling of Strategies**
  - ▷ Table T(w,u) is indexed by continuum of mixed strategies between [0,1].
  - ▷ For implementation purposes the interval [0,1] is sampled by $\tau = \epsilon/2^{k+1} k \log k$
  - ▷ The condition for computing the nash equilibrium has been eased by computing the set of strategies whose expected payoff M_i is ϵ close to the expected payoff if all players use the optimum strategy.
- **Computing the Tables in each pass**
  - ▷ To compute the tables 'T' the factors(in this case the Pay Off Matrix) have been multiplied with each of the 'w' and 'v' strategies for which the Table passed by immediate neighbours have been set to 1.
  - ▷ The Strategy is computed as below

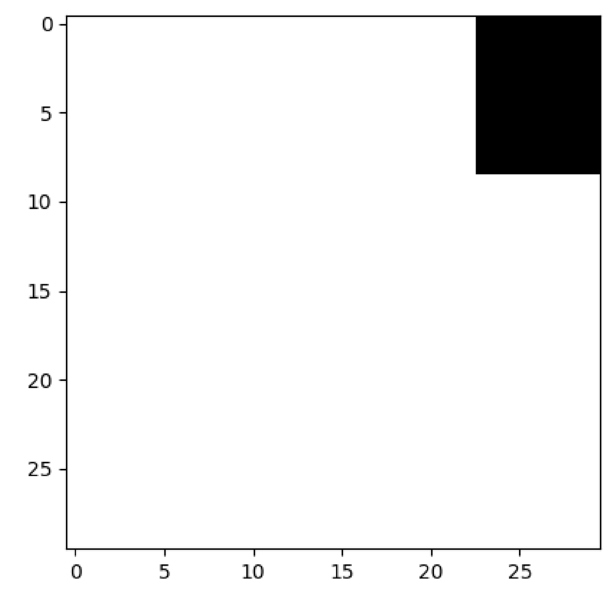
$$T(w, v) = 0 if\ M(w) - M_i(w) > \epsilon \qquad (2)$$

Results

- We cross validated the Loopy belief propagation on a single loop graph with '5' players. The cross validation was done using the Game Theoritic Software 'GAMBIT' which uses standard polynomial solvers to compute Nash strategy.
- In the Figure (4b), We denote a gray scale image of the Table T on the variable nodes with every pass. 0 entries are filled in black and 1 entries are filled as white.
- With every pass the search space of the strategies nash equilibria is reduced and is seen by the filling of black color into the image
- ϵ and τ values have been adjusted for every experiment setting based on the number of nodes 'n' and number of loops 'l'


(a) 4 Node Graph passes.


(b) 5 Node Graphs


(c) 60 Variable Node for the 1st pass.

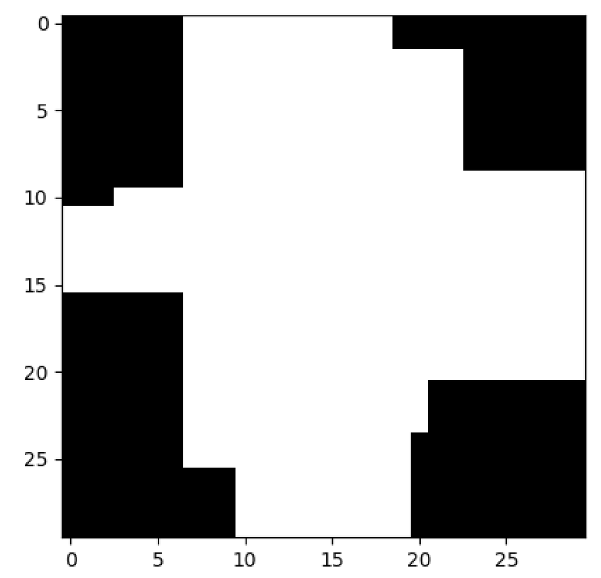

(d) 60 Variable Node for the 7th pass..

Figure: 4 Node Loopy Graph for 4 passes, 60 Node Loopy Graph with 120 links tested for ϵ = 0.4 shown in the table filling done for a single node. (left) 1st pass (right) after convergence .
Graphs for which tests have been performed with multiple loops

Conclusion

A standard problem of Hotellings have been tested and cross validated. In addition we have generated multiple graph with randomized pay off matrices by varying number of loops , maximum Degree of Nodes & number of players

- We find that Nash Prop worked better for Highly loopy graphs over graphs like in hotelling problem that only had one loop
- For many payoff matrices and different graphs Approximate Nash was not found as the graph never converged.
- we have compared the set of Nash equilibria found upon convergence with that by standard solvers to verify the code.

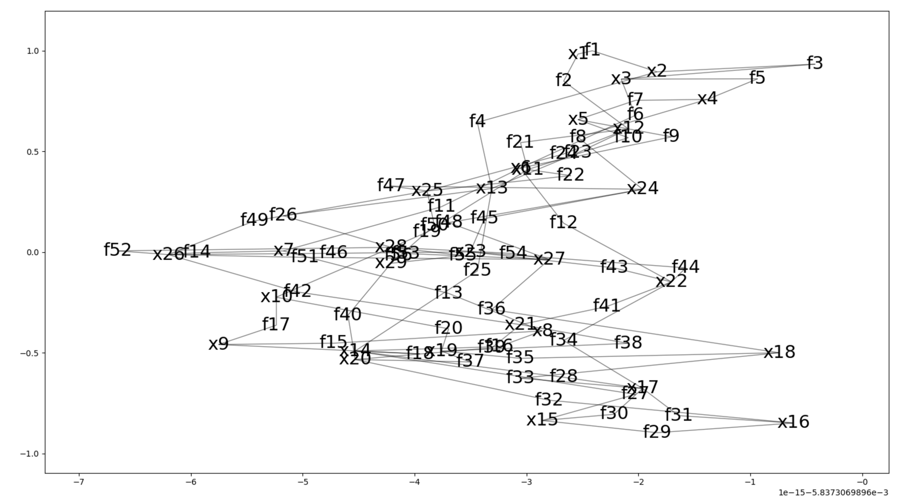


Figure: Graph structure for 30 variable Node.

Mixed strategy range	**Pi** ε [8/30,15/30] **Pj** ε [18/30,21/30]	**Pi** ε [6/30,14/30] **Pj** ε [21/30,29/30]	**Pi** ε 4/10 **Pj** ε [5/10, 6/10, 7/10]
Nodes/links	60/120	30/60	10/20
epsilon/tau	0.4/(1/30)	0.4/ (1/20)	0.26/ 1/10
steps	10	7	4

Figure: Analysis of Mixed strategy found for multiple experiments.