CSCE 875 Seminar:

Multiagent Learning using a Variable Learning Rate

Team: Wolfpack

Mamur Hossain
Istiaque Ali
Jarod Petersen

12/08/2011

Citation of the Article

Bowling, M. and M. Veloso (2002). Multiagent Learning Using a Variable Learning Rate, Artificial Intelligence, 136:215-250.

Outline

- Motivation
- Previous work
- WoLF principle
- Result analysis with self-play games
- Result analysis with variable strategies
- Conclusion
- Praises
- Critiques
- Applications

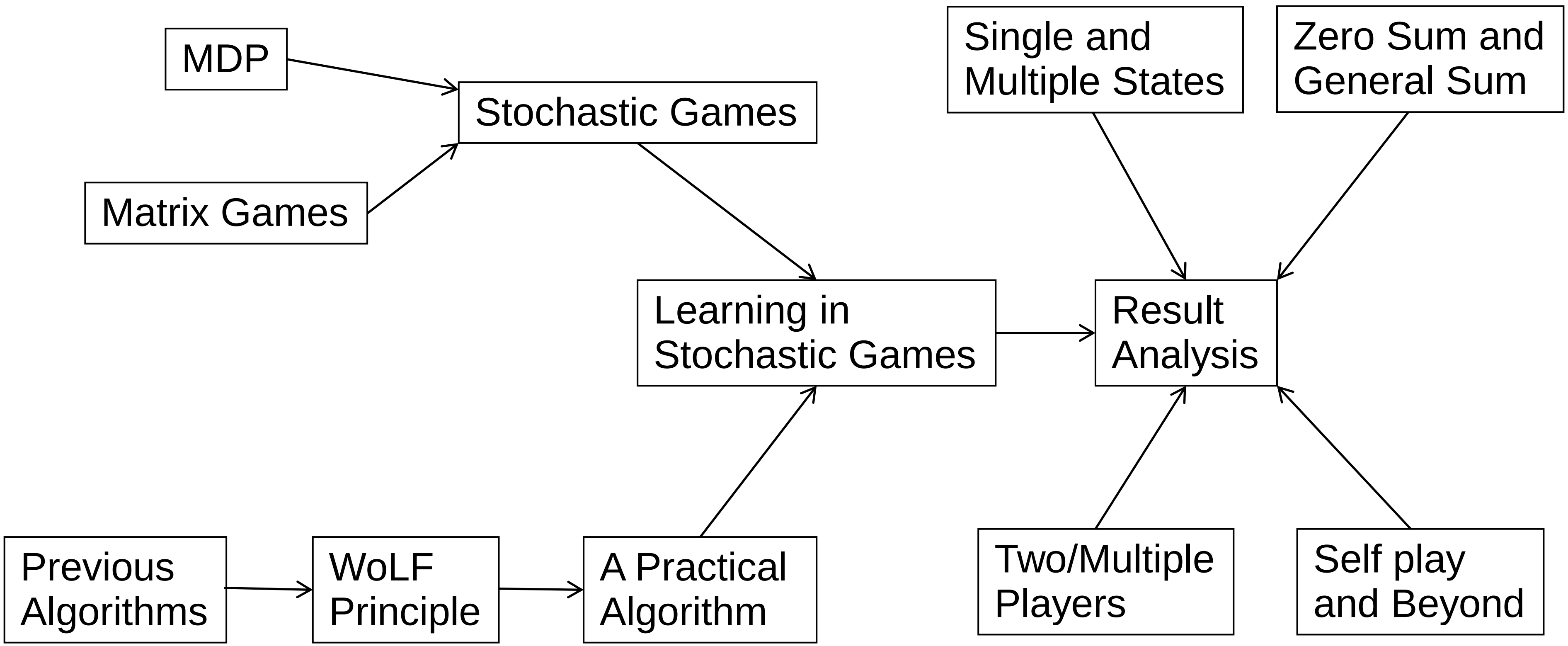
Introduction

- Multiagent systems are being applied in various fields such as robotics, disaster management, e-commerce
- Need robust algorithms for coordinating multiple agents
- Agent Learning is required to discover and exploit the dynamics of the environment
- Learning is difficult in case of an environment with “moving target”
- Multiagent learning has strong connection with game theory
- Introduction of a learning algorithm based-on game theory

Motivation

- Previous contributions on multiagent learning introduce two important desirable properties –
  - Rationality
  - Convergence
- Previous algorithms offer either one of these properties, not both
- This paper introduces an algorithm that addresses both
- The developed algorithm uses the WoLF principle

Overview of the Paper



Stochastic Game Framework

- Markov Decision Process (MDP) – a single agent, multiple state framework
- Matrix games – a multiple agent, single state framework
- Stochastic games – merging of MDP and Matrix games
- Learning in stochastic games is difficult because of moving targets
- Some previous work have been done using “On-Policy Q-learning”

Markov Decision Process

- Also known as MDP – *single agent*, *multiple state* framework
- A model for decision making in an uncertain, dynamic world
- Formally, MDP is a tuple, (S,A,T,R), where S is the set of states, A is the set of actions, T is a transition function $S \times A \times S \rightarrow [0, 1]$, and R is a reward function $S \times A \rightarrow R$.

Matrix Games

- A matrix game or strategic game is a tuple $(n, A_{1\ldots n}, R_{1\ldots n})$
  - n is the number of players
  - A is the joint action space and
  - R is the payoff function of player i
- In a matrix game, players find strategies to maximize their payoffs
  - Pure strategy – selection of action deterministically
  - Mixed strategy – selection of action probabilistically from available actions
- Types of matrix games – zero sum games, general sum games

$$R_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad R_2 = -R_1$$

(a) Matching Pennies

$$R_1 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \qquad R_2 = -R_1$$

(b) Rock-Paper-Scissors

$$R_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

(c) Coordination Game

(a) and (b) are *zero sum* games, (c) is a *general sum* game

Stochastic Games

- A Stochastic game is a combination of Matrix games and MDP
- Multiple agents, multiple states
- A stochastic game is a tuple $(n, S, A_{1\ldots n}, T, R_{1\ldots n})$
  - n is the number of players
  - S is the set of states
  - A is the joint action space and
  - T is a transition function $S \times A \times S \rightarrow [0, 1]$
  - R is the payoff function of player i
- Types of stochastic games – strictly collaborative games, strictly competitive games

Learning in Stochastic Games

- Simultaneous learning of agents
- Two desirable properties of multiagent learning algorithms –
  - *Rationality* – The learner plays its best response policy in reply to other agents' stationary policies.
  - *Convergence* – The learner's policy will converge to a stationary policy in reply to other players' learning algorithms (stationary or rational)
- In case of using rational learning algorithm by the players, if their policies converge, they will converge to an equilibrium
- In this article, the discussion is mostly in case of *self play*

Previous Algorithms

- A number of algorithms for “solving” stochastic games
- Algorithms using reinforcement learning -
  - Q-learning –
    - Single agent learning that finds optimal policies in MDPs
    - Does not play stochastic policy
    - Rational but not convergent
  - Minimax Q –
    - Extension of Q-learning to zero-sum stochastic games
    - Q-function is extended to maintain the value of *joint* actions
    - Not rational but convergent in self play
  - Opponent modeling
    - Learn explicit models of other players assuming their stationary policy
    - Rational but not convergent

Gradient Ascent Algorithms

- Gradient Ascent as a technique of learning
- Simple two player, two action, general sum repeated games
- Players choose new strategy according to these equations –

$$\alpha_{k+1} = \alpha_k + \eta \frac{\partial V_r(\alpha_k, \beta_k)}{\partial \alpha_k}$$
$$\beta_{k+1} = \beta_k + \eta \frac{\partial V_r(\alpha_k, \beta_k)}{\partial \beta_k}.$$

  - α is the strategy of the row player, β is the strategy of the column player
  - η is a fixed step size
  - $\partial V_r(\alpha,\beta)/ \partial\alpha$ and $\partial V_r(\alpha,\beta)/ \partial\beta$ are expected payoffs w.r.t. strategies
  - k is the number of iterations
- Rational but not convergent

Infinitesimal Gradient Ascent

- IGA – cases with infinitesimal step size ($\lim_{\eta} \rightarrow 0$)
- *Theorem:* If both players follow Infinitesimal Gradient Ascent (IGA), where ($\eta \rightarrow 0$), then their strategies will converge to a Nash equilibrium OR the average payoffs over time will converge in the limit to the expected payoffs of a Nash equilibrium.
- This is one of the first convergence results of a rational multiagent learning algorithm
- The notion of convergence is rather weak because –
  - players' policies may not converge
  - expected payoffs may not converge

WoLF IGA

- Introduction of variable learning rate in Gradient Ascent
- Steps taken in the direction of the gradient varies –

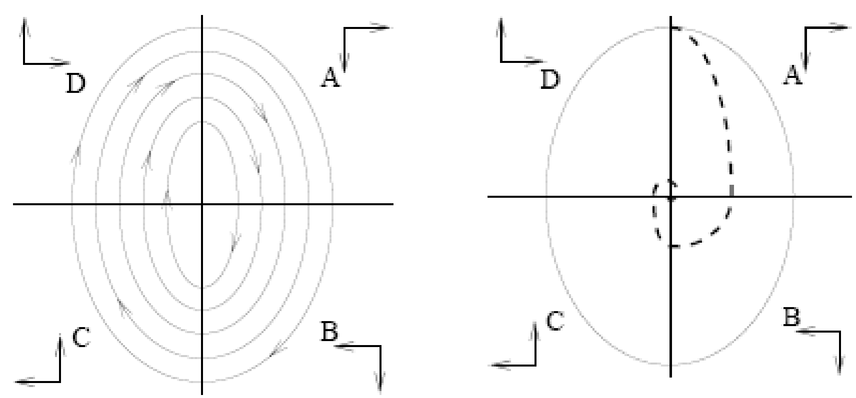
$$\alpha_{k+1} = \alpha_k + \eta \ell_k^r \frac{\partial V_r(\alpha_k, \beta_k)}{\partial \alpha}$$
$$\beta_{k+1} = \beta_k + \eta \ell_k^c \frac{\partial V_r(\alpha_k, \beta_k)}{\partial \beta}$$

where,

$$\ell_k^{r,c} \in [\ell_{\min}, \ell_{\max}] > 0.$$

- WoLF principle – learn quickly when losing, cautiously when winning
- If α^e and β^e are equilibrium strategies, then –

$$\ell_k^r = \begin{cases} \ell_{\min} \text{ if } V_r(\alpha_k, \beta_k) > V_r(\alpha^e, \beta_k) & \text{WINNING} \\ \ell_{\max} \text{ otherwise} & \text{LOSING} \end{cases}$$
$$\ell_k^c = \begin{cases} \ell_{\min} \text{ if } V_c(\alpha_k, \beta_k) > V_c(\alpha_k, \beta^e) & \text{WINNING} \\ \ell_{\max} \text{ otherwise} & \text{LOSING} \end{cases}$$



IGA: does not converge vs. *WoLF IGA:* converges

Requirements of Gradient Ascent Algorithms

- Gradient Ascent requires –
  - Player’s own payoff matrix
  - Actual distribution of actions the other player is playing
- Limitations of Gradient Ascent are -
  - Payoffs are often not known and needed to be learned from experience
  - Often the action of other player is known, not the distribution of actions
- WoLF Gradient Ascent requires –
  - Known Nash equilibrium (unknown for more general algorithm)
  - Difficulty of determining win / loss in case of unknown equilibrium

A Practical Algorithm

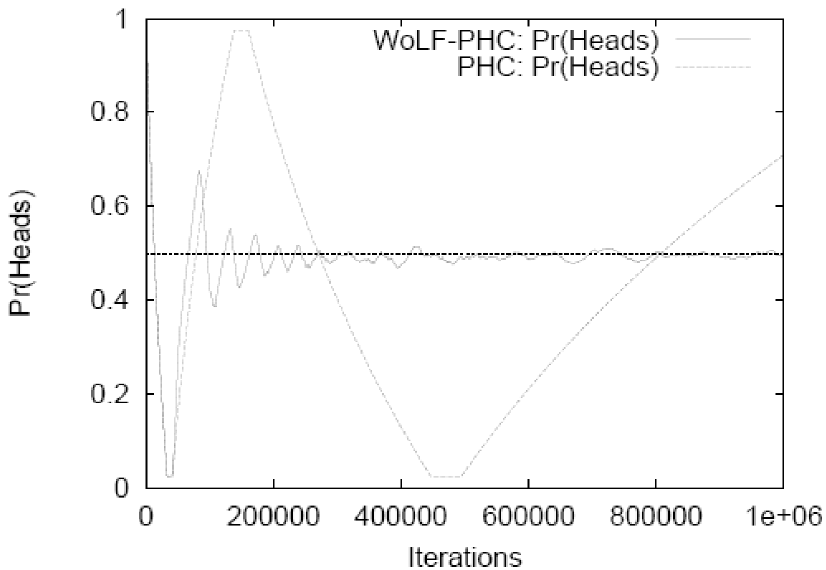
- Policy Hill Climbing (PHC) –
  - A simple rational learning algorithm
  - Capable of playing mixed strategies
  - Q-values are maintained as in normal Q-learning
  - In addition, a current mixed policy is maintained
  - The policy is improved by increasing the probability of highest valued action according to a learning rate δ(0,1]
  - Rational but not convergent
- WoLF PHC
  - Variable learning rate, δ
  - Win / loss is determined using average policy
  - No need to use equilibrium policy
  - Rational *and* convergent

Result Analysis

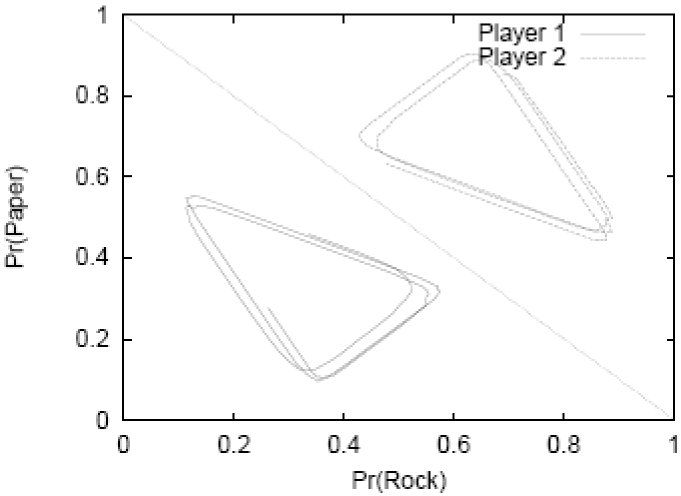
- Examples of applying PHC and WoLF PHC for the following games –
  - Matching pennies and rock-paper-scissor
  - Grid World
  - Soccer
  - Three player matching pennies

Matching Pennies and Rock-Paper-Scissor

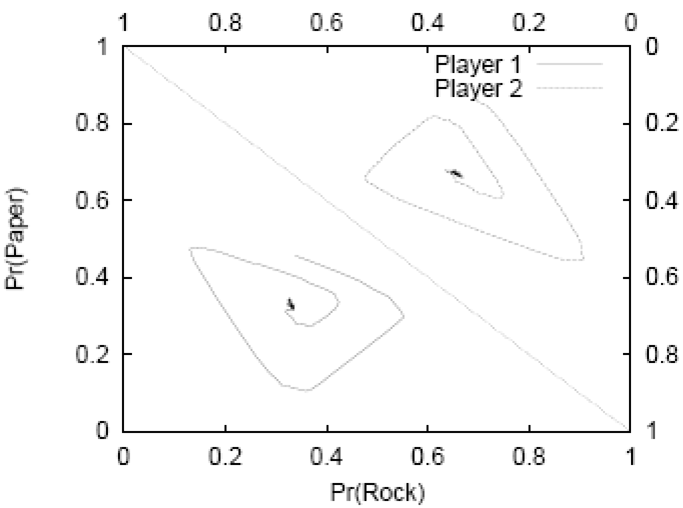
- PHC oscillates around equilibrium, without appearance of converging
- WoLF PHC oscillates around equilibrium with ever decreasing amplitude



Matching pennies game



(a) Policy Hill-Climbing

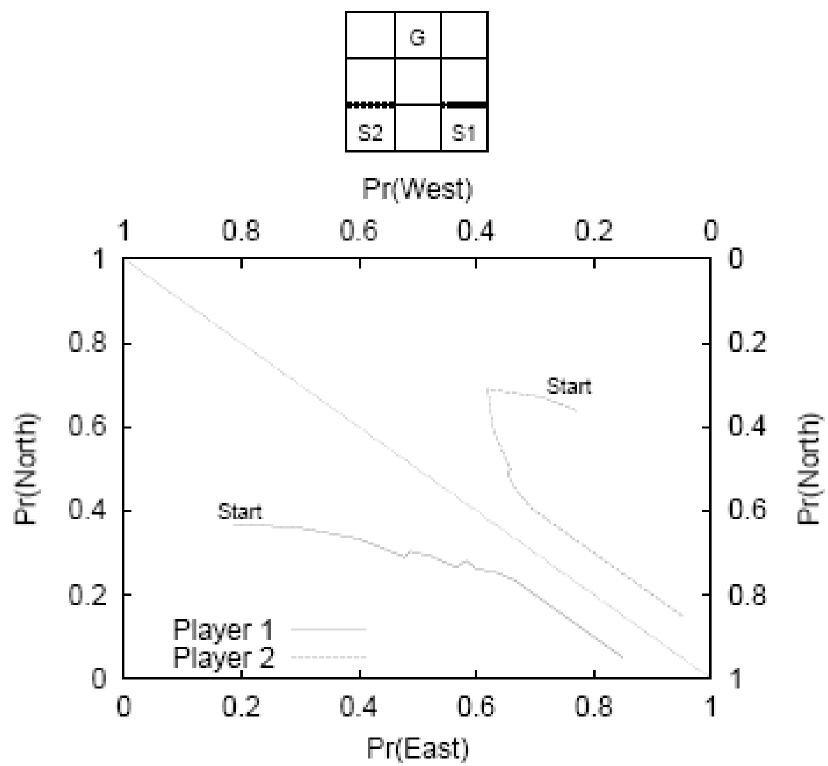


(b) WoLF Policy Hill-Climbing

Rock-paper-scissors game (one million iterations)

Grid World Game

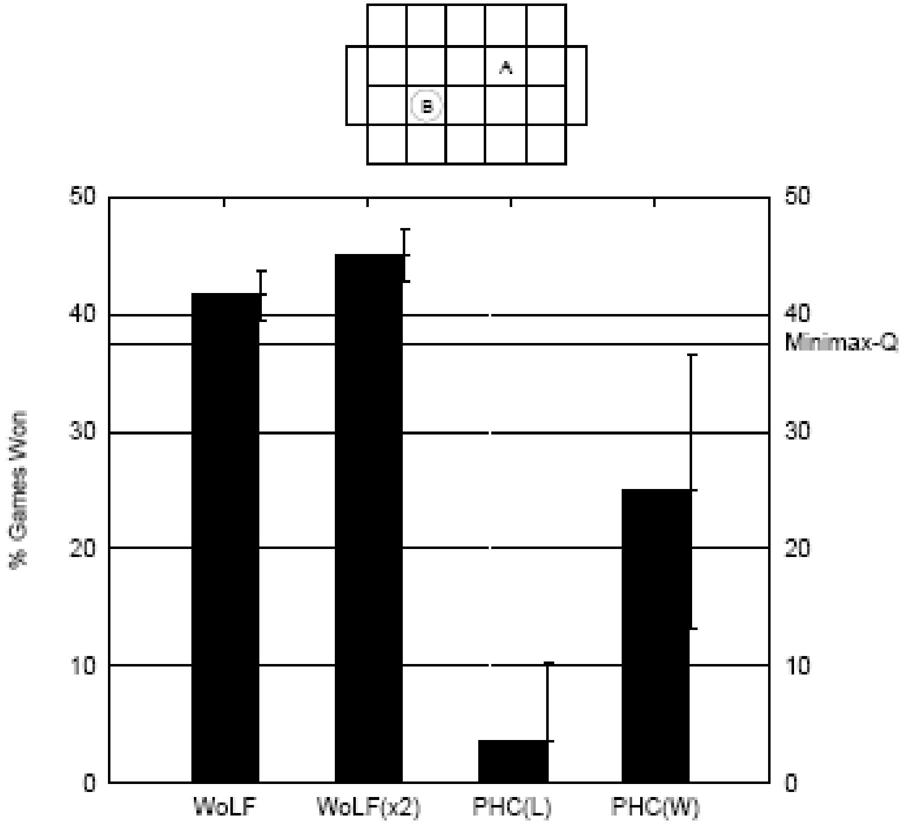
- Agents start in two corners, try to reach the goal in the opposite wall
- Players have four compass directions (N,S,E,W)
- In attempt to move to same squares, both moves fail
- For WoLF PHC, players converges to equilibrium (PHC is not tested)



For WoLF PHC: Initial states of learning (100,000 steps)

Soccer Game

- Goal of the players is to carry the ball to the goal in the opposite wall
- Available actions are four compass directions and not moving
- Attempt to move to an occupied square results in ball possession of stationary agent
- Closer to 50% win against opponent means closer to the equilibrium

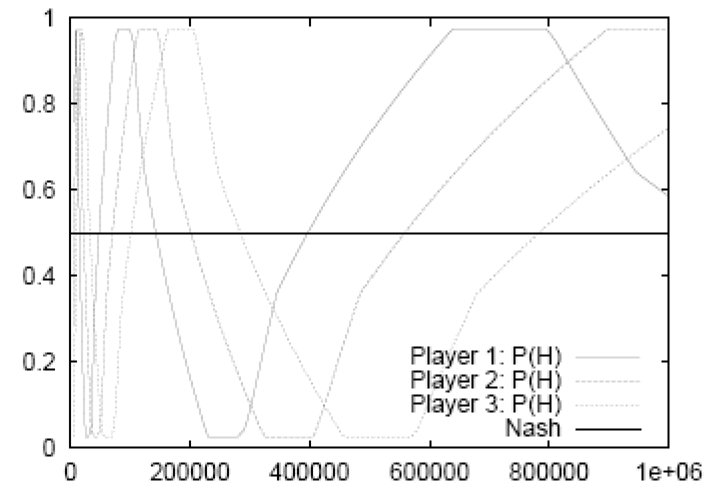


Three Players Matching Pennies Games

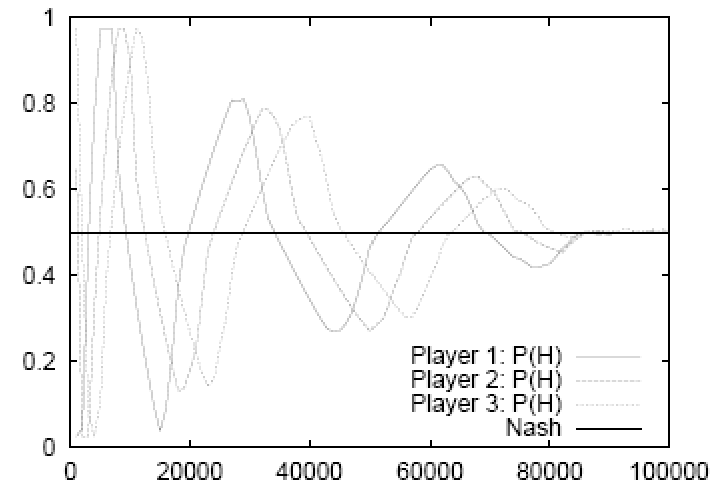
- Involving more than two players in a game
- Player 1: row, player 2: column, player 3: right or left table
- WoLF PHC is compared against Nash equilibrium
- Convergent in case of high ratio of learning rate

	H	T
H	+1, +1, −1	−1, −1, −1
T	−1, +1, +1	+1, −1, +1

	H	T
H	+1, −1, +1	−1, +1, +1
T	−1, −1, −1	+1, +1, −1



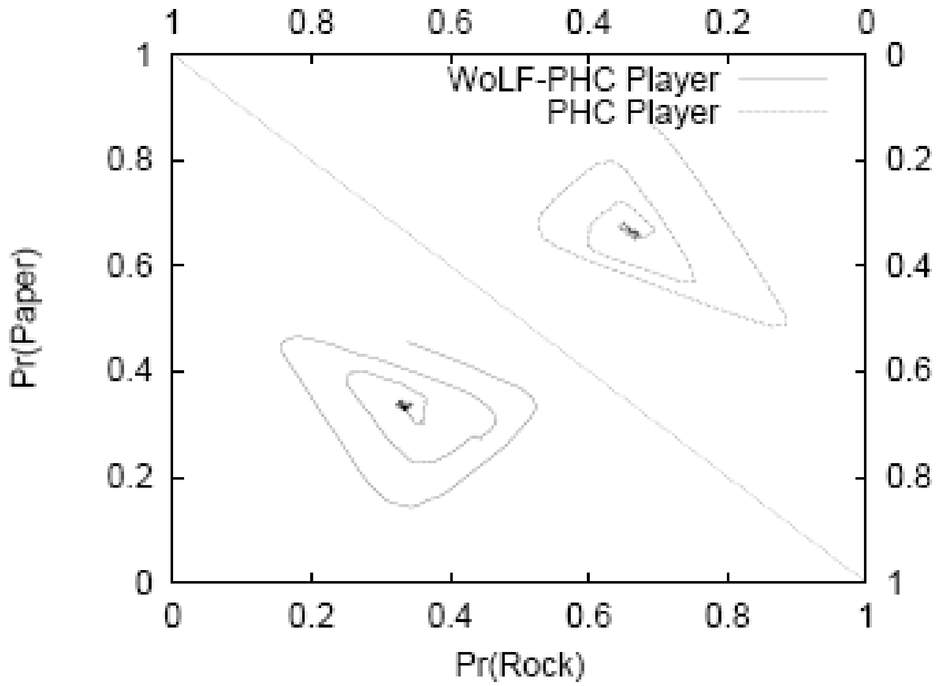
(a) $\delta_l/\delta_w = 2$



(b) $\delta_l/\delta_w = 3$

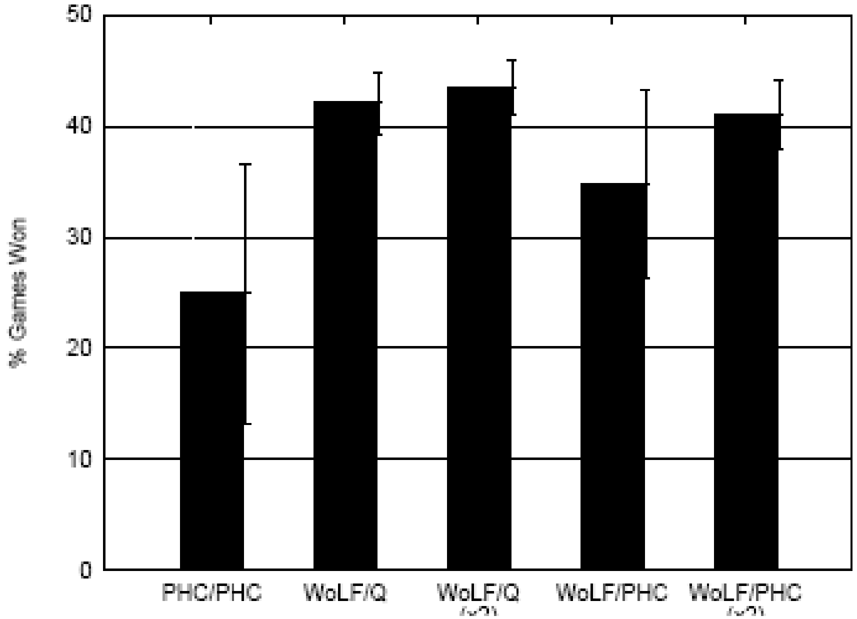
Matrix Game beyond Self-Play

- Rock-paper-scissors was tested for PHC vs. WoLF PHC
- Convergence was attained to Nash equilibrium
- Convergence is slower than with two WoLF learners (i.e, self play)



Soccer Game beyond Self-Play

- WoLF tested against opponents having PHC and Q-learning
- Closer to 50% win against opponent means closer to the equilibrium
- Learned policy is comparatively closer to equilibrium
- More training moves the policy closer to equilibrium



Conclusion

- Learning in stochastic game framework elucidates learning moving targets
- In this paper, WoLF principle is introduced to define how to vary the learning rate
- Using WoLF principle, a rational algorithm can be made convergent
- Proof has been provided for several different cases –
  - Single vs. multiple state
  - Zero sum vs. general sum games
  - Two player vs. multiple player stochastic games
- Two important future directions –
  - Explore learning outside self play
  - Making the algorithm scale to large problems

Discussion

- Our discussion is presented in terms of -
  - Praises in favor of the WoLF PHC algorithm
  - Critiques against the algorithm
  - Applications of the developed algorithm

Praises

- The paper introduces a strong algorithm for obtaining two important desirable learning properties – rationality and convergence
- The developed algorithm is robust – it can be used for two / multiple players, self play and beyond, zero sum and general sum games
- The algorithm was successful to handle mixed strategy profiles
- It demonstrates the effects of training rates on convergence
- The paper also demonstrates effects of high / low learning ratio on convergence

Critiques

- The algorithm uses MDP which is a discretized approximation of a continuous system
- In case of a large system, the algorithm may be computationally challenging because of maintaining the Q-values and variable learning rates
- The algorithm required very high number of training / iterations to converge to equilibrium
- The paper did not discuss consequences of communication among the learning agents

Applications

- The algorithm is suitable for stochastic games
- It can be applied both in the cases of self play and beyond
- Another possible application is for multiple players (as well as two players) games
- Practical applications are –
  - Robocup robots' learning that includes multiple players
  - Disaster management robotic systems where they use different learning strategies
  - Share market where multiple agents learn in different strategies
- In our final project of "Shark-Sardine Model," such learning could be applied –
  - For learning among the shark agents
  - For learning among the sardine agents

Summary

- The paper introduces a new learning algorithm utilizing variable learning rate
- The developed algorithm addresses two desirable properties: rationality and convergence
- Explanation of a stochastic game framework is provided
- Previous algorithms are explained with examples
- Results using the new algorithm for different games is presented
- The praises, critiques and applications of the WoLF algorithm are presented

Learning Algorithms	**Rationality**	**Convergence**	**Mixed Policy**
Q-Learning			
Minimax Q			
Opponent Modeling			
Gradient Ascent			
IGA			
WoLF IGA			
PHC			
WoLF PHC			

Q & A