Leading Best-Response Strategies in Repeated Games

Investigation of weaknesses and improvements

Giliam de Carpentier
Richard Noorlandt

June 8, 2006

Overview

- Introduction
- Bi-Matrix Games
- Q-Learning
- Leader Strategies
- Extensions
- Results
- Conclusions

Introduction

- Littman and Stone propose leader-follower strategies: Bully and Godfather
- Leaders manipulate followers in an attempt to maximize performance
- This only works under certain conditions
- Can we improve performance?

Bi-Matrix Games

- The two-player game is represented by a payoff matrix for each player
- Both players known each other's matrix
- Player 1 chooses a column of the matrices. Player 2 chooses a row

Our focus:

- 2x2 bi-matrix games where $M_1 = M_2^T$

Zero-Sum Games

- Characteristic: $M_1 + M_2 = 0$
- Well understood in literature
- Proof of convergence

Example: Matching Pennies

$$M_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

General-Sum Games

Best choice:

Mutual cooperation, else defect

- Prisoner's Dilemma

$$M_1 = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix}$$

Best choice:

Opposite action

- Chicken

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix}$$

Q-Learning

- Reinforcement learning
- Learns from previous interactions
- Assumes stationary, single-agent environments but also performs well in other environments
- Can be made optimal for zero-sum games
- Non-optimal for larger general-sum games

Q-Learning

- Updates a $Q_{<state, action>}$ value after each iteration
- General update rule:

$$Q(x,i) = \alpha(r + \gamma \max_{i'} Q(y,i')) + (1-\alpha)Q(x,i)$$

- Q_0: state-less: 2 Q-values

$$Q_0(i) = \alpha(r + \gamma \max_{i'} Q_0(i')) + (1-\alpha)Q_0(i)$$

- Q_1: x $\in$ {own actions}: 4 Q-values

$$Q_1(x,i) = \alpha(r + \gamma \max_{i'} Q_1(i,i')) + (1-\alpha)Q_1(x,i)$$

α = learning rate
r = payoff
γ = discount rate (importance future)
i = action to choose
i' = next action
x = current state
y = next state

Q-Learning

- Convergence of Q values are independent of exploration strategy
- Paper suggest ε-greedy policy
  - Fixed learning rate => Exploration-exploitation ratio constant throughout experiment.
  - (Very) slow convergence, remains flexible.
- Alternative: Boltzmann policy
  - Dynamic learning rate => Exploring (learning) Q-values at startup, exploiting (using) Q-values later on.
  - Fast convergence, fast fixation.

Leader Strategies

- Q-learning is only guaranteed to perform well in stationary, single-agent environments
- This doesn’t include multi-agent systems
- Exploration/exploitation heuristics can improve this
- But doesn’t guarantee optimality or convergence

- Paper suggests a leader-follower strategy pair where the leader tries to force the other best-response (Q-) learner into certain actions

Leader Strategies - Bully

Leader strategy:

- "What will the other player do when I do this?"
- "What is then the most beneficial for me"

Or, more formally: $i^* = \arg\max_i M_1(i, \arg\max_j M_2(i, j))$

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3.0 & 3.5 \\ 1.5 & 1.0 \end{bmatrix}$$

Leader Strategies - Godfather

Leader strategy steps:

- Determine security level of itself and its opponent
- Check if an offer exists that allows both players to score above security level: the targetable pair
- Start with own half of targetable pair
- Force opponent's security level if didn't play its half

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3.0 & 3.5 \\ 1.5 & 1.0 \end{bmatrix}$$

Leader Strategies - Godfather

Leader strategy steps:

- Determine security level of itself and its opponent
- Check if an offer exists that allows both players to score above security level: the targetable pair
- Start with own half of targetable pair
- Force opponent's sec. level if it didn't play its half

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3.0 & 3.5 \\ 1.5 & 1.0 \end{bmatrix}$$

Leader Strategies - Godfather

Leader strategy steps:

- Determine security level of itself and its opponent
- Check if an offer exists that allows both players to score above security level: the targetable pair
- Start with own half of targetable pair
- Force opponent's sec. level if it didn't play its half

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3.0 & 3.5 \\ 1.5 & 1.0 \end{bmatrix}$$

Experiments

- To measure and verify performance, we implemented an agent-agent simulator, capable of repeating the simulations done by Littman and Stone
  - 100.000 experiments for each player pair
  - Each experiment has 30.000 iterations
  - Analyze payoff of last 5.000 iterations

- However… We also implemented new strategies

Experiments

- Implemented 12 different agent types
  - Q_0 & Q_1 with either greedy policy or Boltzmann policy
  - The Random strategy
  - The Bully and Godfather strategies
  - Different Godfather extension combinations
  - A TF2T-like Godfather
- Results for every agent-agent combination for each of the 4 game types
- Only some results will be mentioned here…

Extensions - Best response exploit

- Bully-like extension to the Godfather strategy
- Improvement: Never execute a threat if the best response action of the opponent equals the opponent's half of the targetable pair
- Consequence: Opponent doesn't need to learn from the threat for some game types

$$M_1 = \begin{bmatrix} \textcircled{3} & 0 \\ 2 & \boxed{1} \end{bmatrix} \qquad M_2 = \begin{bmatrix} \textcircled{3} & 2 \\ 0 & \boxed{1} \end{bmatrix}$$

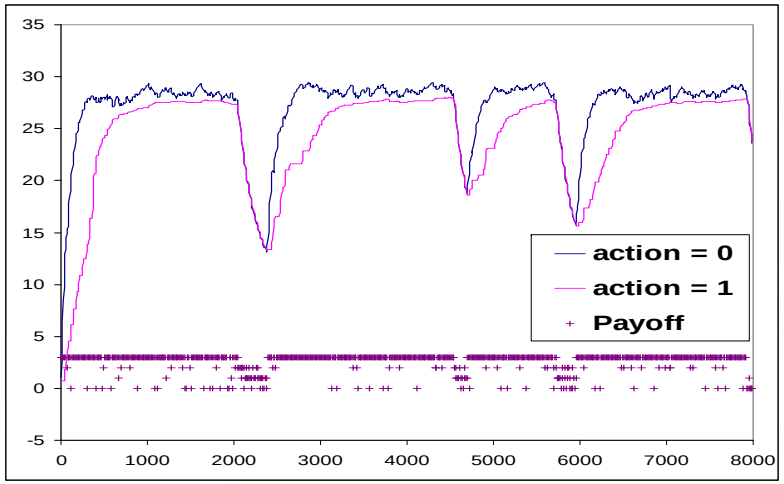
○ = offer

□ = security level

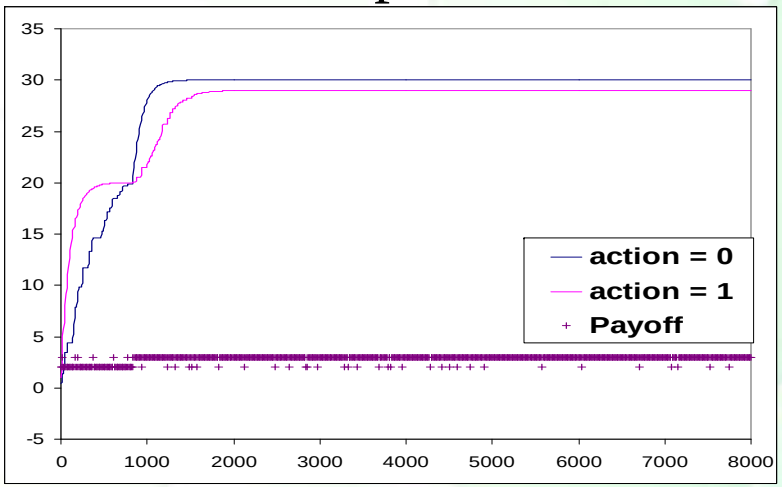
Extensions - Best response exploit

Example: Godfather vs Q_0 in the Assurance game

without self-punishment



with self-punishment



$$M_1 = \begin{bmatrix} \textcircled{3} & 0 \\ 2 & \boxed{1} \end{bmatrix} \qquad M_2 = \begin{bmatrix} \textcircled{3} & 2 \\ 0 & \boxed{1} \end{bmatrix}$$

○ = offer

□ = security level

Extensions - Prediction

- Godfather executes its threat one iteration after the opponent did't play its half
- Q_0 isn't able to learn from this threat execution
- Improvement: Use internal Q_0 predictor that learns and acts the same as a Q_0 opponent
- Consequence: Threats can be executed in the same iteration where the Q_0 opponent is expected to defect

Extensions - Prediction

The prediction strategy can be implemented in different ways:

1. If the last prediction was correct, execute direct punishment when predicted

2. If the running average of the predictor accuracy exceeds a certain threshold, allow punishment when predicted

Extensions - Prediction

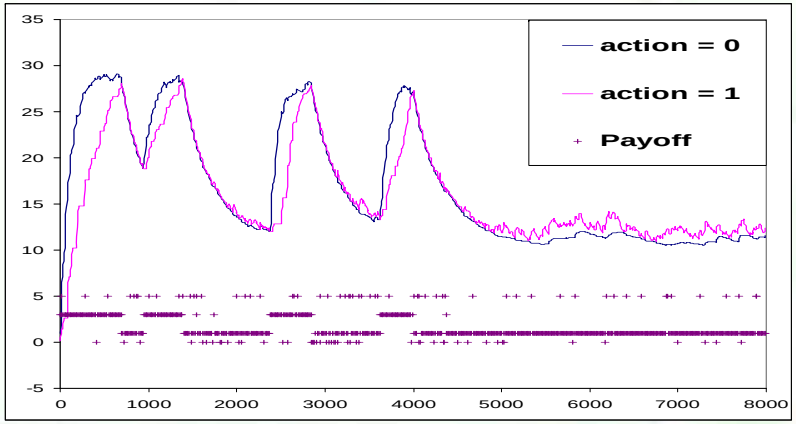
The prediction strategy can be implemented in different ways:

1. If the last prediction was correct, execute direct punishment when predicted

2. If the running average of the predictor accuracy exceeds a certain threshold, allow punishment when predicted

Q_0 opponent

Standard Godfather:



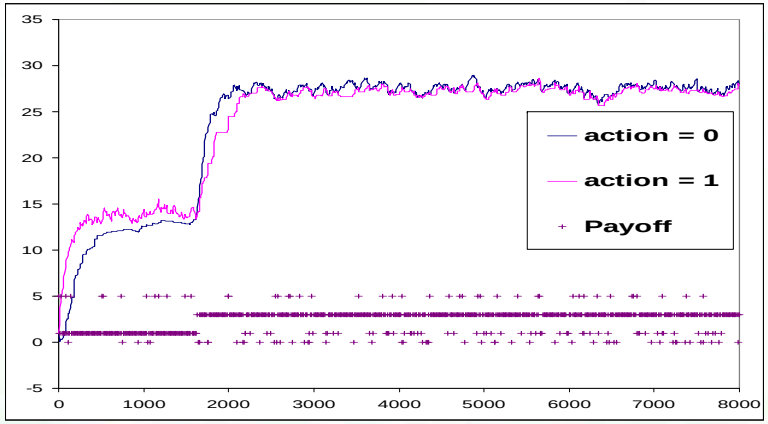
Extensions - Prediction

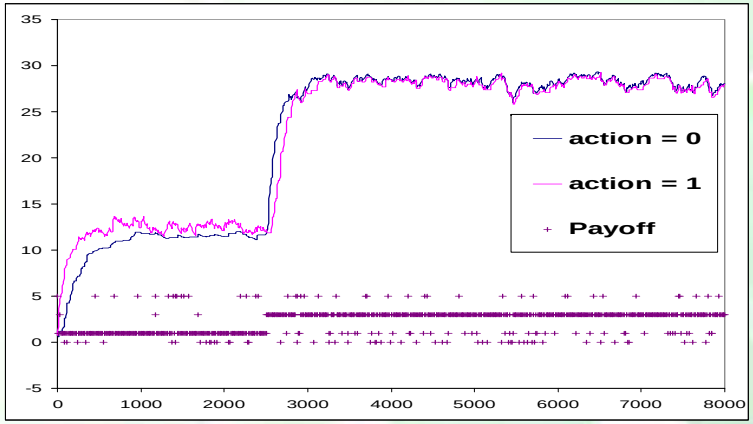
The prediction strategy can be implemented in different ways:

1. If the last prediction was correct, execute direct punishment when predicted

2. If the running average of the predictor accuracy exceeds a certain threshold, allow punishment when predicted

Q_0 opponent





Extensions - Prediction

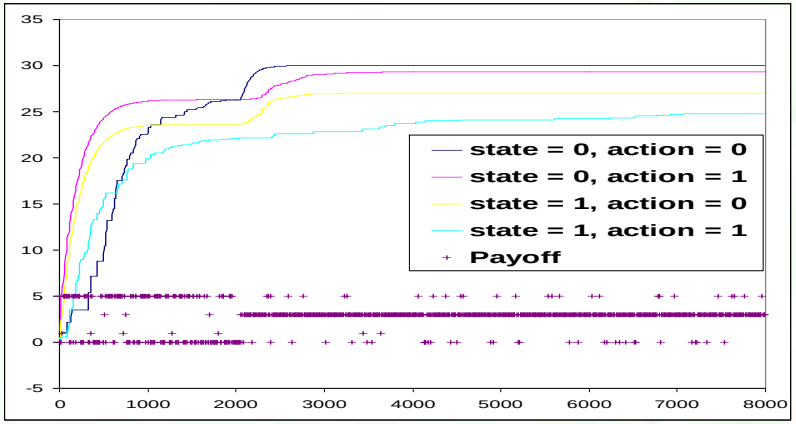
The prediction strategy can be implemented in different ways:

1. If the last prediction was correct, execute direct punishment when predicted

2. If the running average of the predictor accuracy exceeds a certain threshold, allow punishment when predicted

Q_1 opponent

Standard Godfather:



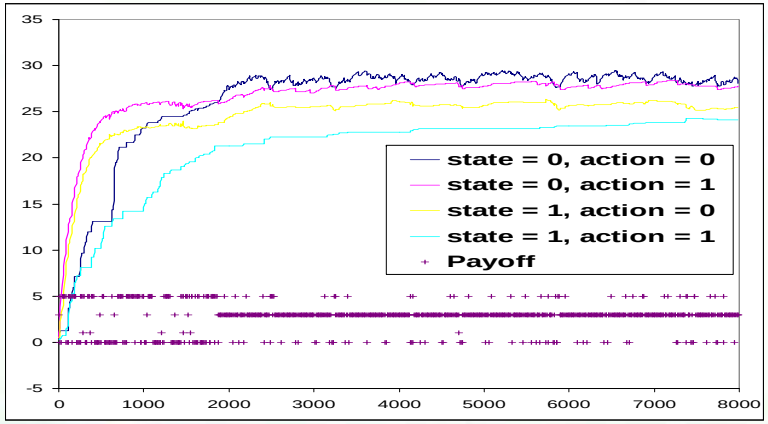
Extensions - Prediction

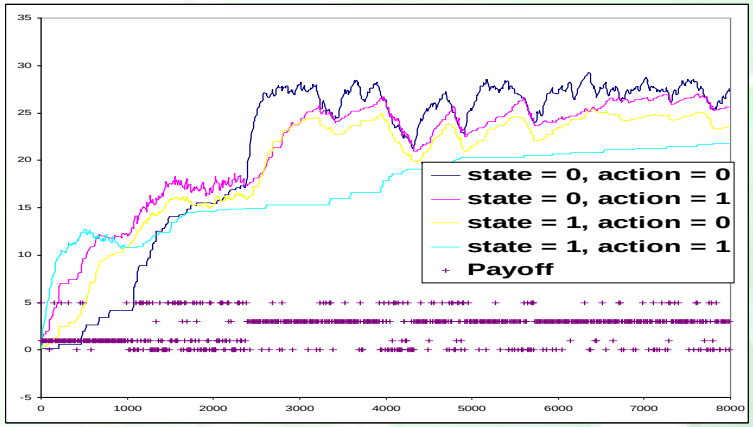
The prediction strategy can be implemented in different ways:

1. If the last prediction was correct, execute direct punishment when predicted

2. If the running average of the predictor accuracy exceeds a certain threshold, allow punishment when predicted

Q_1 opponent





Extensions - Prediction

The prediction strategy can be implemented in different ways:

1. If the last prediction was correct, execute direct punishment when predicted

2. If the running average of the predictor accuracy exceeds a certain threshold, allow punishment when predicted

Results:

- Both perform well against Q0
- Implementation 1. is more stable than 2. against Q1

Results - Deadlock Game

Player score / Opponent	Q0	Q1	Bully	Godf	Godf Ext
Q0	2.805	2.805	2.950	2.805	2.950
Q1	2.805	2.805	2.950	2.805	2.950
Bully	2.850	2.850	3.000	3.000	3.000
Godfather	2.805	2.805	3.000	3.000	3.000
Godfather Ext	2.850	2.850	3.000	3.000	3.000

Godfather's strategy

Bully's strategy

$$M_1 = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}$$

= best response

= security level

Results - Chicken Game

Player score / Opponent	Q0	Q1	Bully	Godf	Godf Ext
Q0	*2.432*	*2.479*	3.375	2.850	2.909
Q1	*2.440*	2.867	3.375	2.948	2.966
Bully	1.475	1.475	1.000	1.000	1.024
Godfather	2.850	2.948	1.000	3.000	2.492
Godfather Ext	2.845	2.881	1.119	2.492	2.498

= exploited = exploiting = unstable

Godfather strategy Bully strategy

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3.0 & 3.5 \\ 1.5 & 1.0 \end{bmatrix}$$

= best response

= security level

Results - Assurance Game

Player score / Opponent	Q0	Q1	Bully	Godf	Godf Ext
Q0	1.561	1.683	2.850	1.343	2.850
Q1	1.683	1.952	2.850	2.805	2.850
Bully	2.950	2.950	3.000	3.000	3.000
Godfather	1.343	2.805	3.000	3.000	3.000
Godfather Ext	2.950	2.950	3.000	3.000	3.000

= unstable

Godfather's strategy

Bully's strategy

$$M_1 = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}$$

= best response

= security level

Results - Prisoner's Dilemma

Player score / Opponent	Q0	Q1	Bully	Godf	Godf Ext
Q0	1.181	1.158	1.200	1.355	2.757
Q1	1.175	1.205	1.200	2.948	2.982
Bully	0.950	0.950	1.000	1.000	0.952
Godfather	1.355	2.948	1.000	3.000	2.491
Godfather Ext	2.726	2.786	1.190	2.491	2.473

= unstable

Godfather strategy

Bully strategy

$$M_1 = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3 & 5 \\ 0 & 1 \end{bmatrix}$$

= best response

= bully's choice

Conclusions

- Introducing Leader strategies can help convergence if its opponent uses a best response strategy
- Leader-leader experiments result in mixed performance. Dynamically changing role might improve performance.
- Bully's performance depends heavily on the game and opponent type

Conclusions

- Godfather's TFT-like approach works well in most situations
- Godfather can be extended with a best response exploit and/or a Q_0-predictor to futher improve its performance where possible
- Q's ε-greedy policy should only be used for easier comparison of results. Use Boltzmann for much more efficient convergence.

Questions ?

Nash Equilibrium

A game reaches Nash Equilibrium when neither player can perform better by changing only his own strategy

Example: Chicken

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 3.0 & 3.5 \\ 1.5 & 1.0 \end{bmatrix}$$

Boltzmann policy

An action is chosen with the following probability:

$$P(a) = \frac{\exp(\frac{Q(a)}{T})}{\sum_{i \in \mathrm{A}} \exp(\frac{Q(i)}{T})}$$

Where $T = T_{start} \cdot c^{iteration-1}$.

Our experiments used T_{start} = 16 and c = .9

Deadlock Game

Opponent \ Player score	Q0	Q1	Bu	Gf	Ra	Gb	Gp	Gn	Gu	G2	B0	B1
Q0 (Q0)	2.805	2.805	2.950	2.805	1.993	2.950	2.681	2.793	2.950	2.943	2.950	2.950
Q1 (Q1)	2.805	2.805	2.950	2.805	1.993	2.950	2.681	2.793	2.950	2.943	2.950	2.950
Bu (Bully)	2.850	2.850	3.000	3.000	2.010	3.000	2.858	2.858	3.000	3.000	3.000	3.000
Gf (Godfather)	2.805	2.805	3.000	3.000	1.898	3.000	1.048	2.756	3.000	3.000	3.000	3.000
Ra (Random)	2.553	2.553	2.670	1.898	1.898	2.670	1.860	2.279	2.670	2.415	2.670	2.670
Gb (GodfatherBully)	2.850	2.850	3.000	3.000	2.010	3.000	2.857	2.858	3.000	3.000	3.000	3.000
Gp (GodfatherPred)	2.766	2.766	2.953	1.048	1.892	2.952	1.090	2.671	2.952	2.952	2.953	2.953
Gn (GodfatherNonDetPred)	2.801	2.801	2.953	2.756	1.953	2.953	2.590	2.781	2.952	2.946	2.952	2.952
Gu (GodfatherPredBully)	2.850	2.850	3.000	3.000	2.010	3.000	2.857	2.857	3.000	3.000	3.000	3.000
G2 (Godfather2Treason)	2.848	2.848	3.000	3.000	1.973	3.000	2.857	2.854	3.000	3.000	3.000	3.000
B0 (B0)	2.850	2.850	3.000	3.000	2.010	3.000	2.858	2.857	3.000	3.000	3.000	3.000
B1 (B1)	2.850	2.850	3.000	3.000	2.010	3.000	2.858	2.857	3.000	3.000	3.000	3.000

Chicken Game

Opponent \ Player score	Q0	Q1	Bu	Gf	Ra	Gb	Gp	Gn	Gu	G2	B0	B1
Q0 (Q0)	*2.432*	*2.479*	3.375	2.850	1.871	2.850	2.909	2.907	2.909	2.824	1.575	1.575
Q1 (Q1)	*2.440*	2.867	3.375	2.948	2.004	2.948	2.966	2.978	2.966	2.255	1.575	1.575
Bu (Bully)	1.475	1.475	1.000	1.000	1.335	1.000	1.024	1.024	1.024	1.000	1.500	1.500
Gf (Godfather)	2.850	2.948	1.000	3.000	2.561	3.000	2.492	2.821	2.492	3.000	3.000	3.000
Ra (Random)	2.625	2.613	2.675	2.561	2.561	2.561	2.562	2.573	2.562	2.523	2.505	2.505
Gb (GodfatherBully)	2.850	2.948	1.000	3.000	2.561	3.000	2.492	2.821	2.492	3.000	3.000	3.000
Gp (GodfatherPred)	2.845	2.881	1.119	2.492	2.551	2.492	2.498	2.737	2.498	2.929	2.929	2.929
Gn (GodfatherNonDetPred)	2.900	2.805	1.119	2.821	2.428	2.821	2.780	2.840	2.780	2.922	2.929	2.929
Gu (GodfatherPredBully)	2.845	2.881	1.119	2.492	2.551	2.492	2.498	2.737	2.498	2.929	2.929	2.929
G2 (Godfather2Treason)	2.947	3.188	1.000	3.000	2.966	3.000	3.024	3.017	3.024	3.000	3.000	3.000
B0 (B0)	3.475	3.475	3.500	3.000	3.165	3.000	3.024	3.024	3.024	3.000	3.000	3.000
B1 (B1)	3.475	3.475	3.500	3.000	3.165	3.000	3.024	3.024	3.024	3.000	3.000	3.000

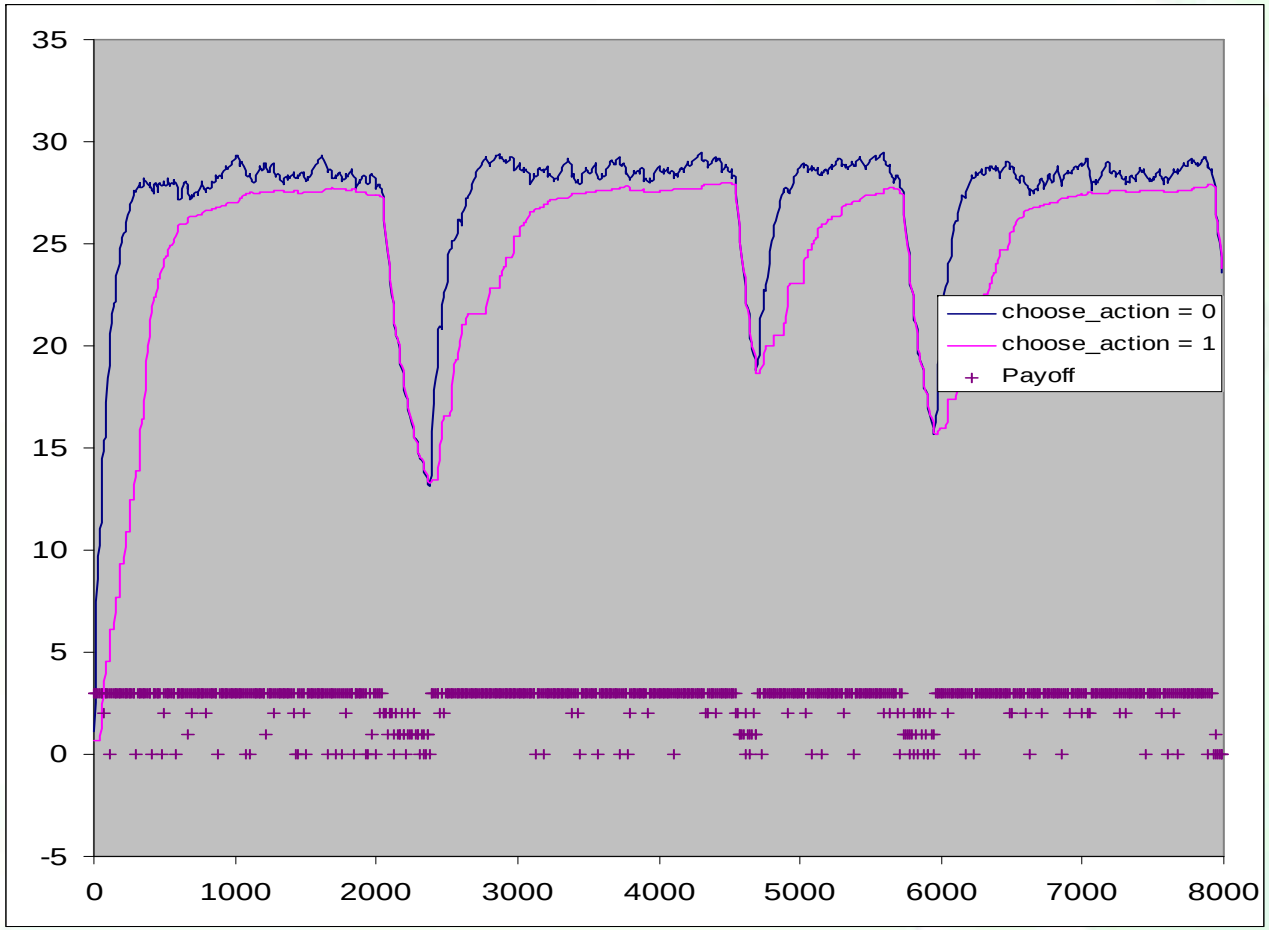
Assurance Game

Opponent \ Player score	Q0	Q1	Bu	Gf	Ra	Gb	Gp	Gn	Gu	G2	B0	B1
Q0 (Q0)	1.561	1.683	2.850	1.343	2.239	2.850	2.711	2.109	2.850	2.701	2.850	2.850
Q1 (Q1)	1.683	1.952	2.850	2.805	2.017	2.850	2.765	2.401	2.850	2.848	2.850	2.850
Bu (Bully)	2.950	2.950	3.000	3.000	2.670	3.000	2.952	2.953	3.000	3.000	3.000	3.000
Gf (Godfather)	1.343	2.805	3.000	3.000	1.898	3.000	1.048	2.732	3.000	3.000	3.000	3.000
Ra (Random)	1.947	1.915	2.010	1.898	1.898	2.010	1.888	1.925	2.010	1.973	2.010	2.010
Gb (GodfatherBully)	2.950	2.950	3.000	3.000	2.670	3.000	2.952	2.952	3.000	3.000	3.000	3.000
Gp (GodfatherPred)	2.635	2.676	2.857	1.048	1.828	2.857	1.117	2.538	2.857	2.857	2.858	2.857
Gn (GodfatherNonDetPred)	2.107	2.374	2.858	2.732	2.084	2.857	2.617	2.683	2.858	2.852	2.857	2.857
Gu (GodfatherPredBully)	2.950	2.950	3.000	3.000	2.670	3.000	2.952	2.953	3.000	3.000	3.000	3.000
G2 (Godfather2Treason)	2.796	2.943	3.000	3.000	2.415	3.000	2.952	2.944	3.000	3.000	3.000	3.000
B0 (B0)	2.950	2.950	3.000	3.000	2.670	3.000	2.953	2.952	3.000	3.000	3.000	3.000
B1 (B1)	2.950	2.950	3.000	3.000	2.670	3.000	2.952	2.952	3.000	3.000	3.000	3.000

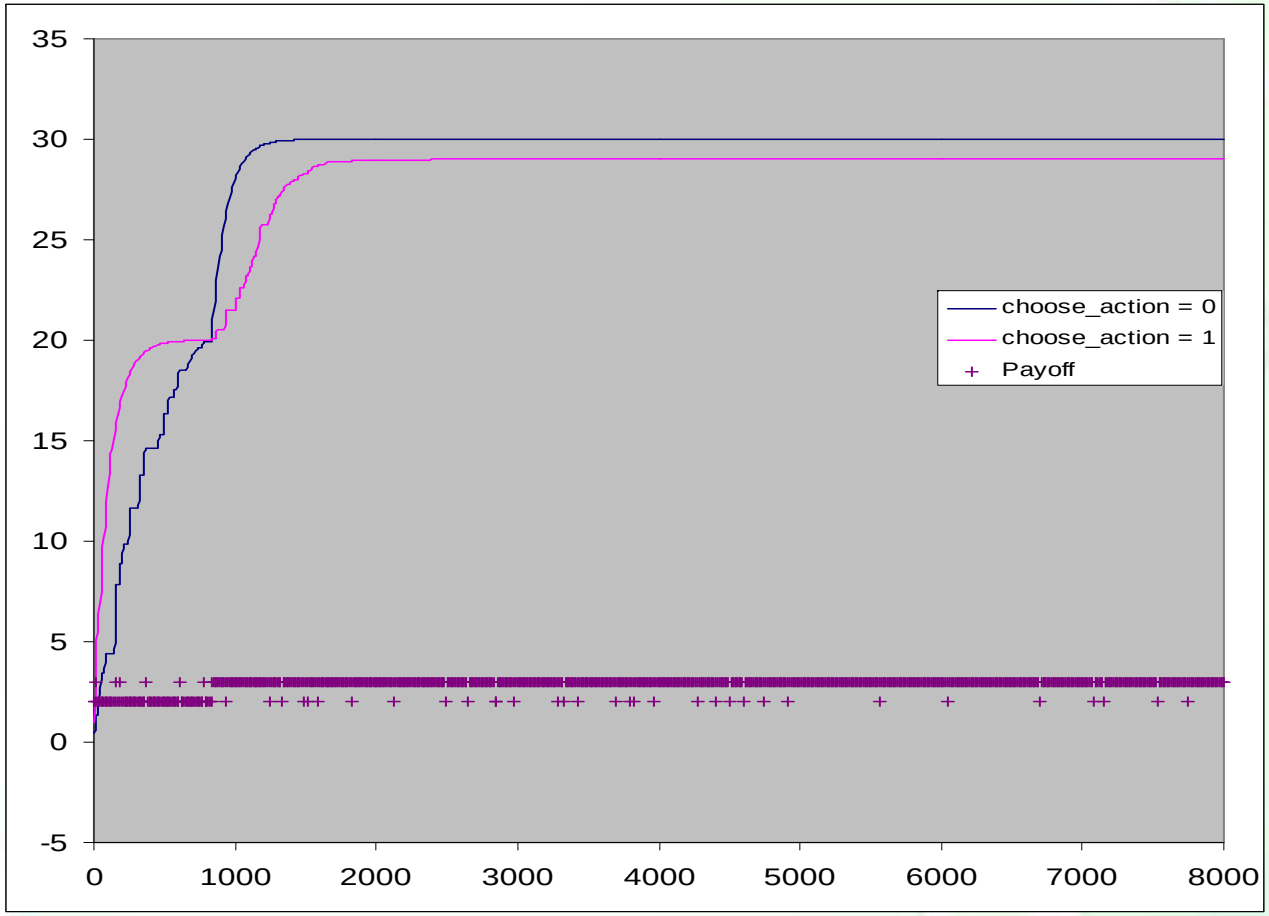
Prisoner's Dilemma

Opponent \ Player score	Q0	Q1	Bu	Gf	Ra	Gb	Gp	Gn	Gu	G2	B0	B1
Q0 (Q0)	1.181	1.158	1.200	1.355	0.536	1.355	2.757	2.805	2.757	1.486	0.150	0.150
Q1 (Q1)	1.175	1.205	1.200	2.948	0.583	2.947	2.982	3.073	2.982	1.546	0.150	0.150
Bu (Bully)	0.950	0.950	1.000	1.000	0.330	1.000	0.953	0.953	0.952	1.000	0.000	0.000
Gf (Godfather)	1.355	2.948	1.000	3.000	2.561	3.000	2.491	1.300	2.491	3.000	3.000	3.000
Ra (Random)	3.577	3.553	3.680	2.561	2.561	2.561	2.588	2.886	2.588	2.192	2.010	2.010
Gb (GodfatherBully)	1.355	2.947	1.000	3.000	2.561	3.000	2.491	1.301	2.491	3.000	3.000	3.000
Gp (GodfatherPred)	2.726	2.786	1.190	2.491	2.508	2.491	2.473	2.068	2.473	2.857	2.858	2.857
Gn (GodfatherNonDetPred)	2.792	2.573	1.190	1.300	1.913	1.301	1.903	1.477	1.903	2.821	2.857	2.857
Gu (GodfatherPredBully)	2.726	2.786	1.190	2.491	2.508	2.491	2.473	2.068	2.473	2.857	2.858	2.858
G2 (Godfather2Treason)	1.817	3.834	1.000	3.000	3.297	3.000	3.095	3.081	3.095	3.000	3.000	3.000
B0 (B0)	4.900	4.900	5.000	3.000	3.660	3.000	3.095	3.095	3.095	3.000	3.000	3.000
B1 (B1)	4.900	4.900	5.000	3.000	3.660	3.000	3.095	3.095	3.095	3.000	3.000	3.000

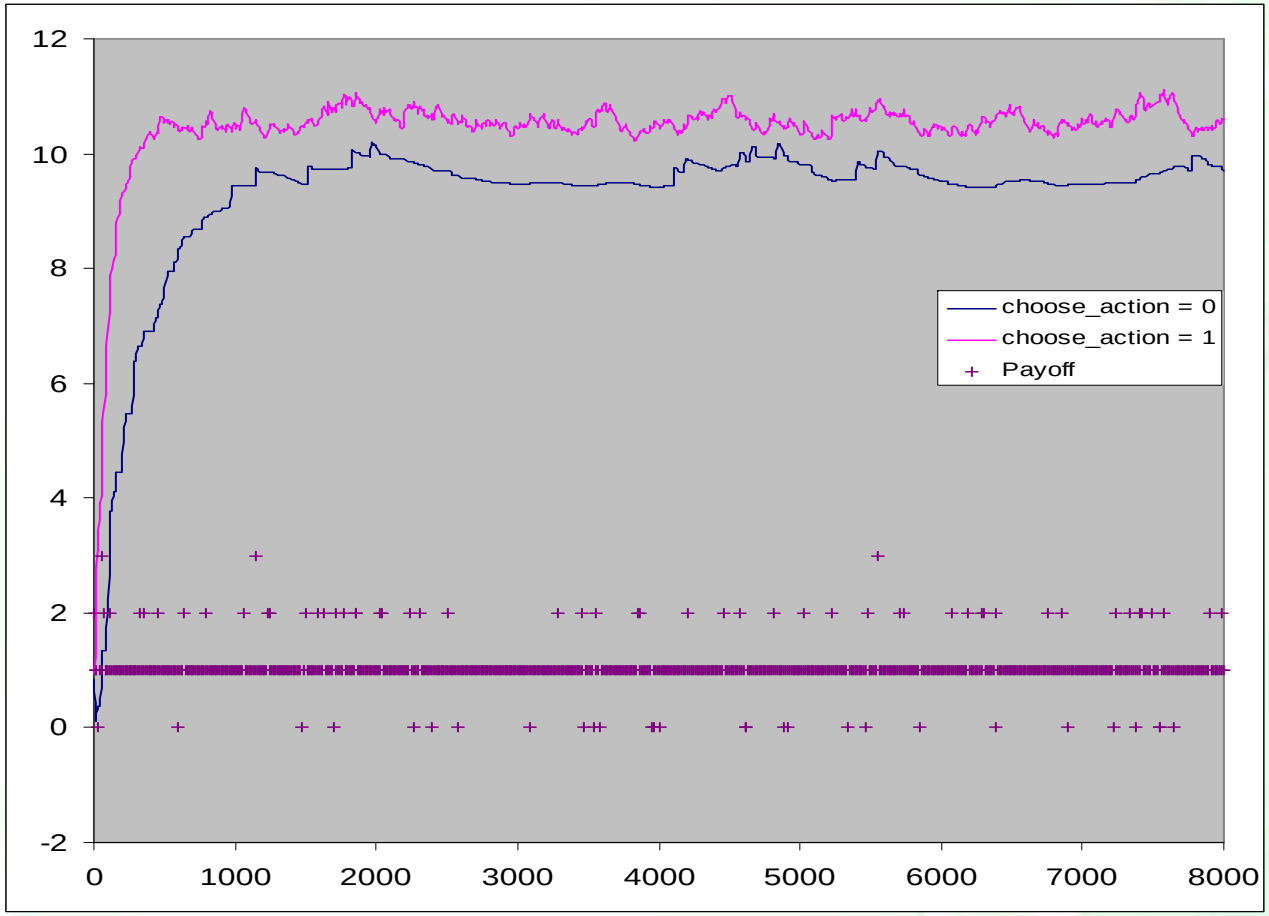
Assurance Graphs - $Gf\text{-}Q_0$



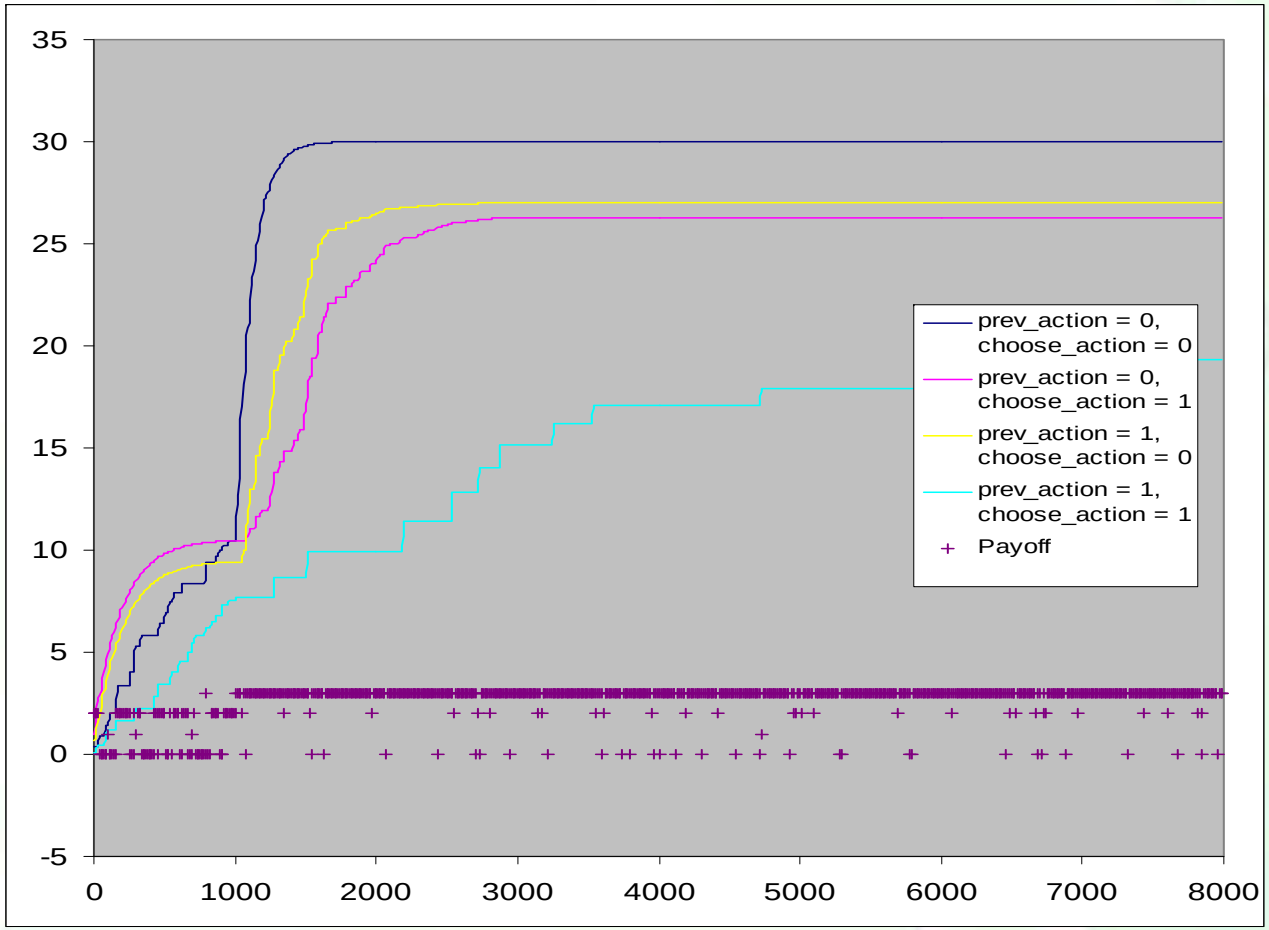
Assurance Graphs - Gu-Q_0



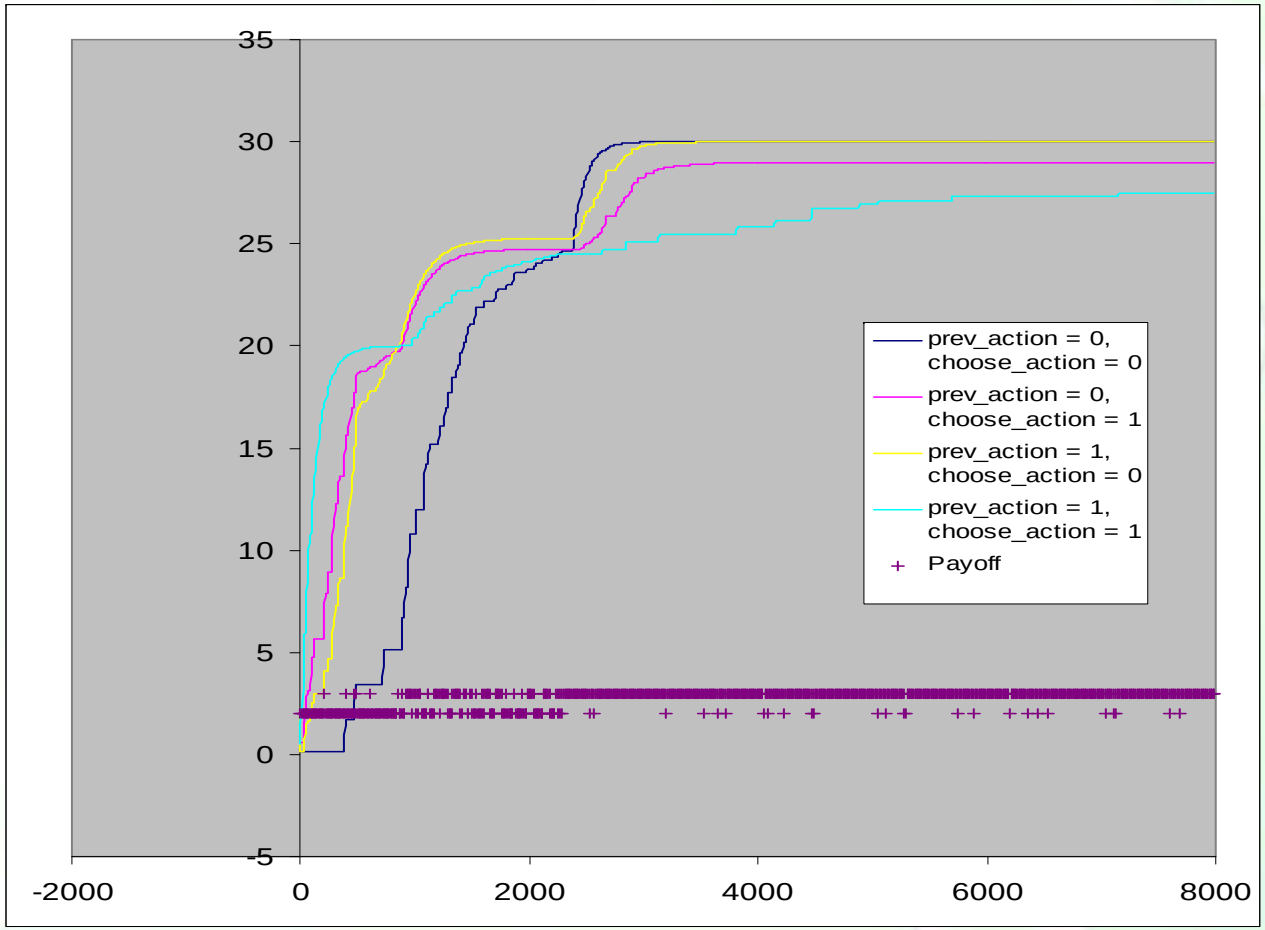
Assurance Graphs - $Gn\text{-}Q_0$



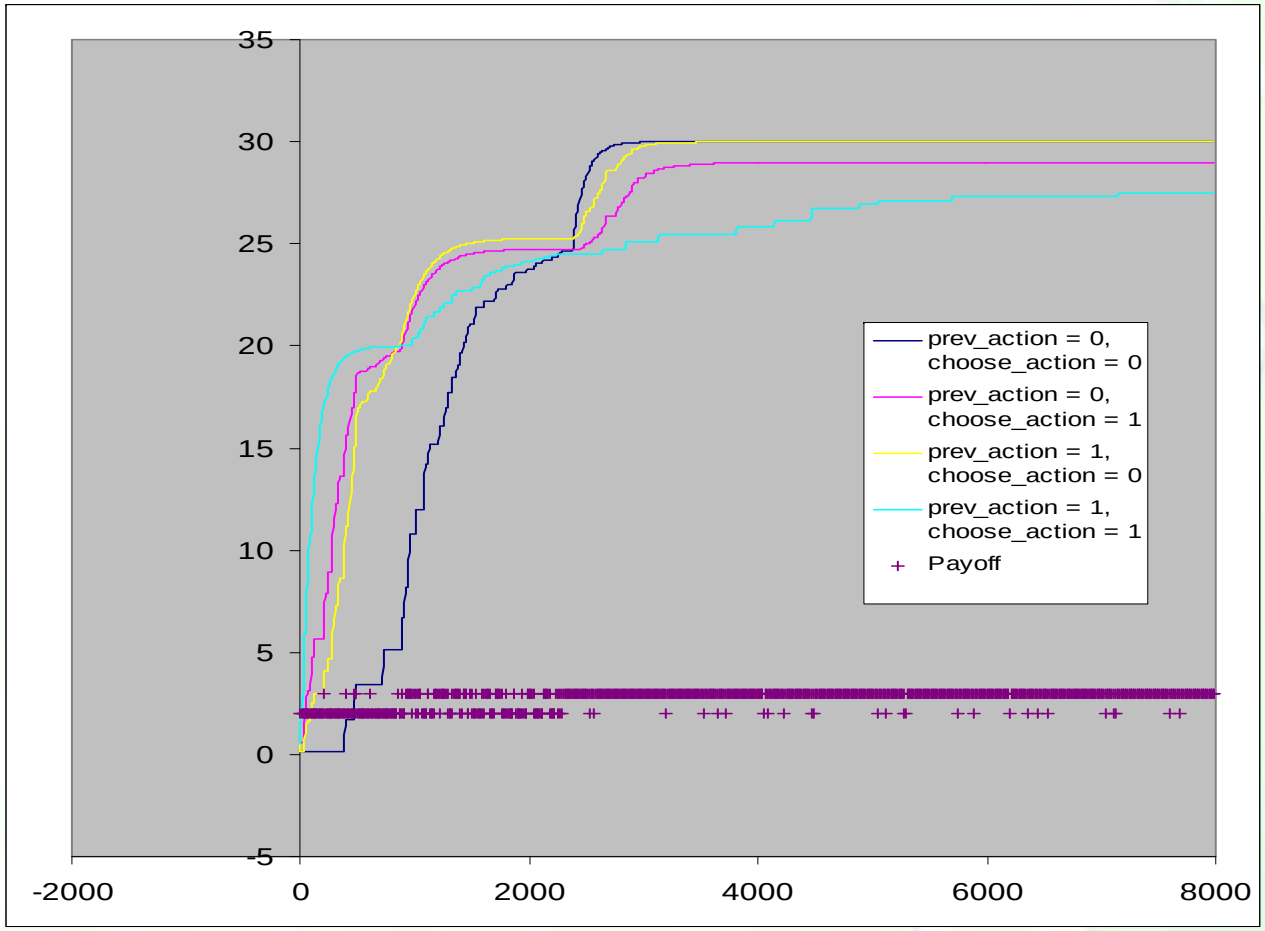
Assurance Graphs - $Gf\text{-}Q_1$



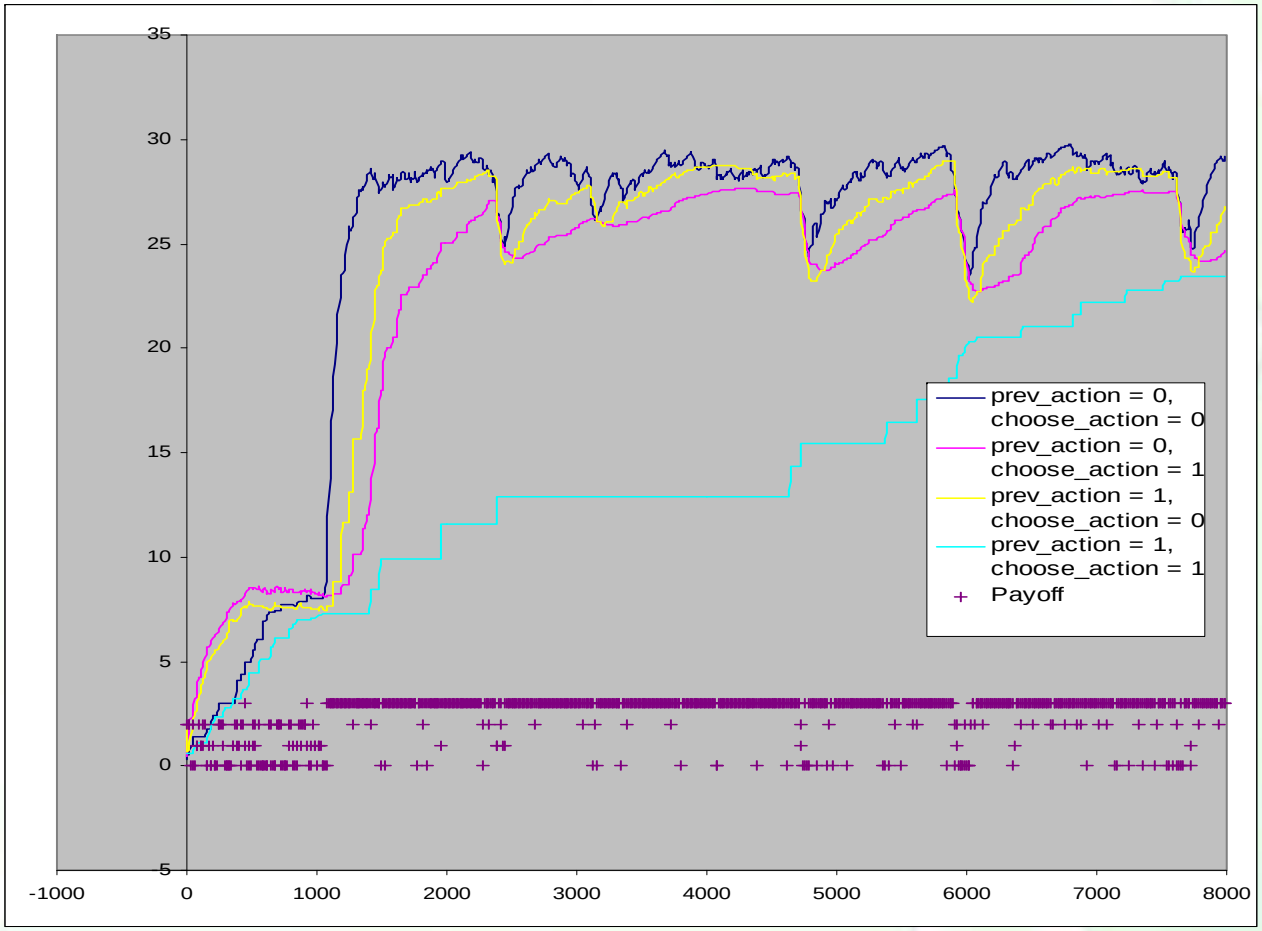
Assurance Graphs - Gu-Q_1



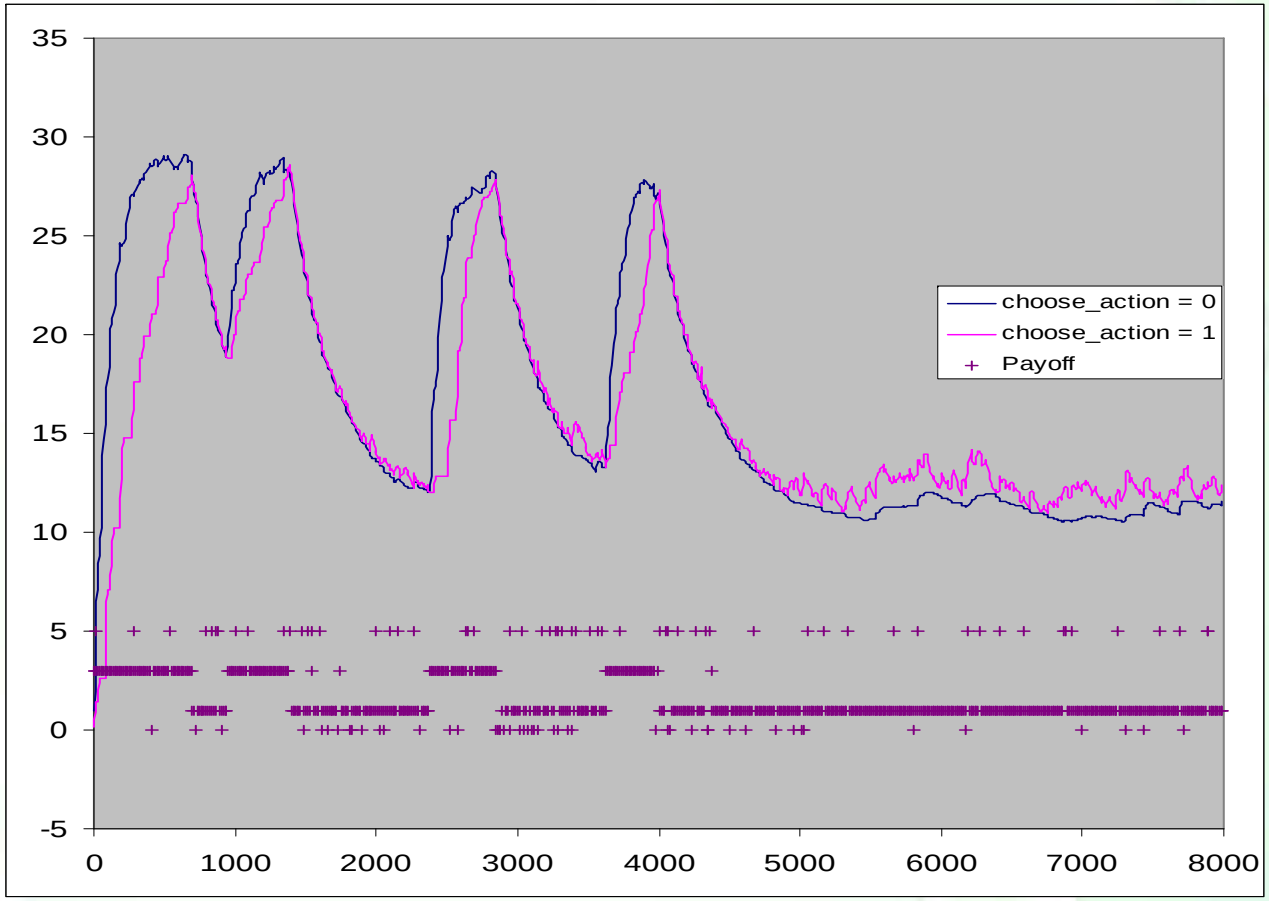
Assurance Graphs - Gu-Q_1



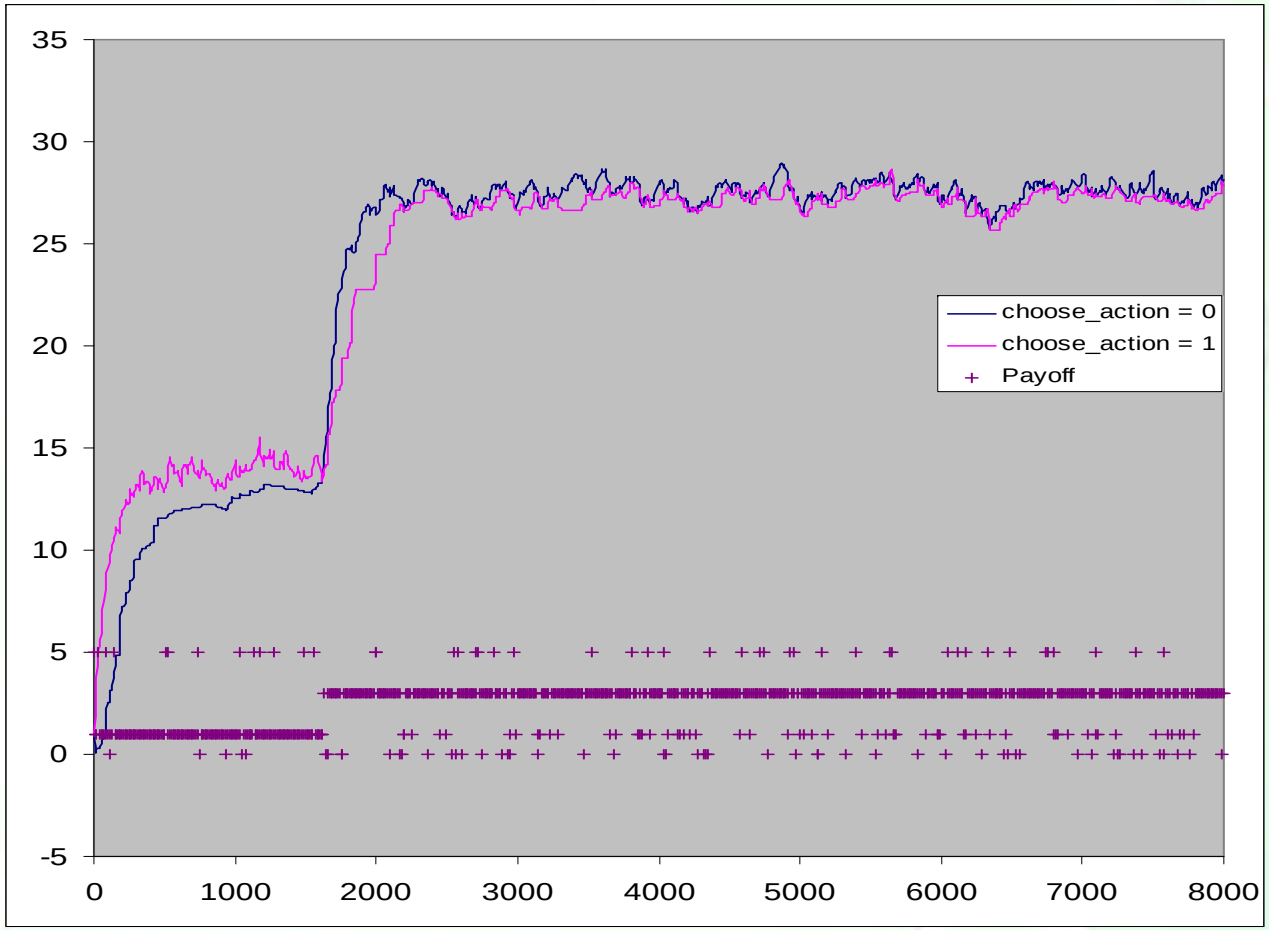
Assurance Graphs - $Gn\text{-}Q_1$



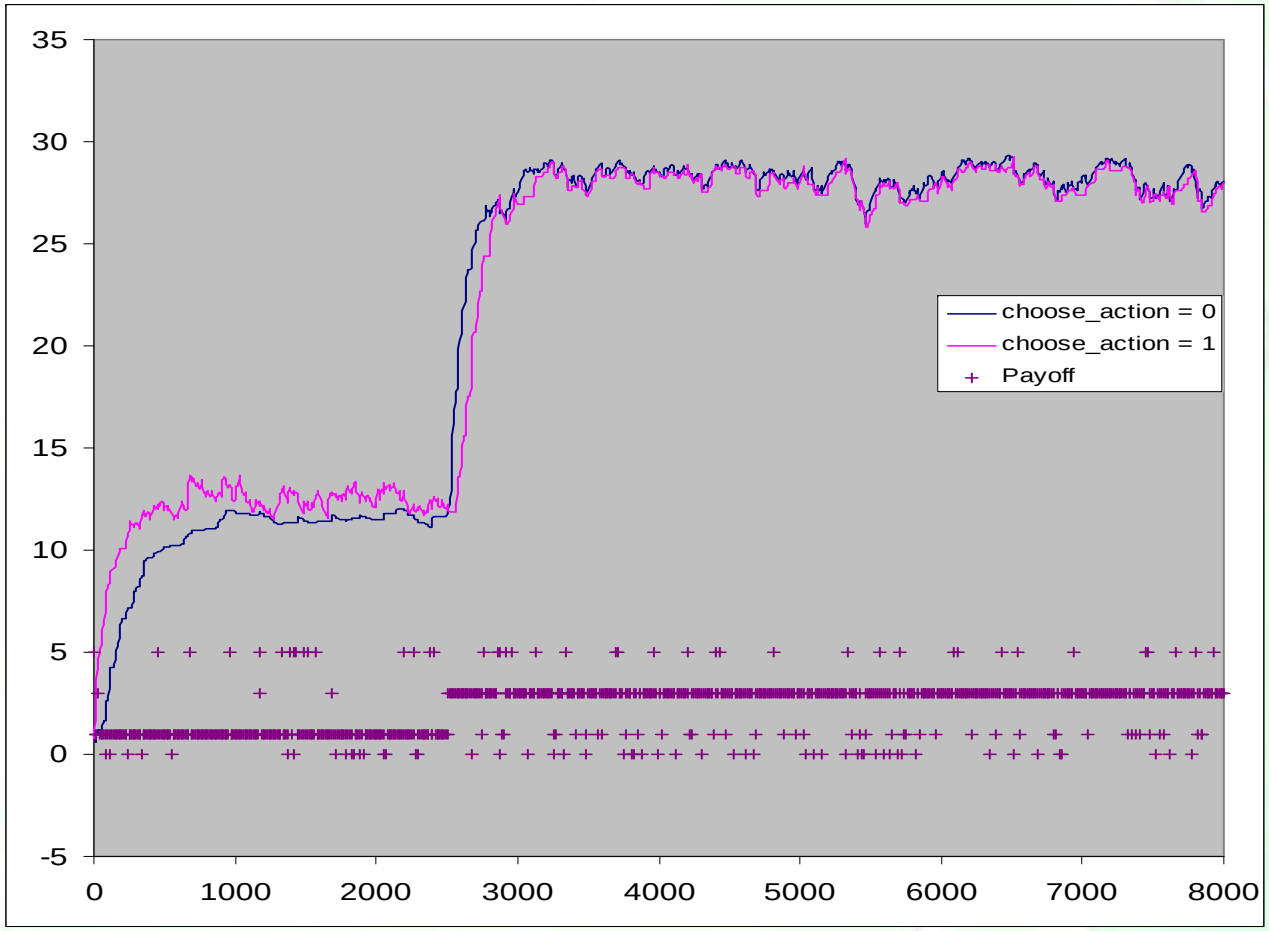
Prisoner Graphs - $Gf\text{-}Q_0$



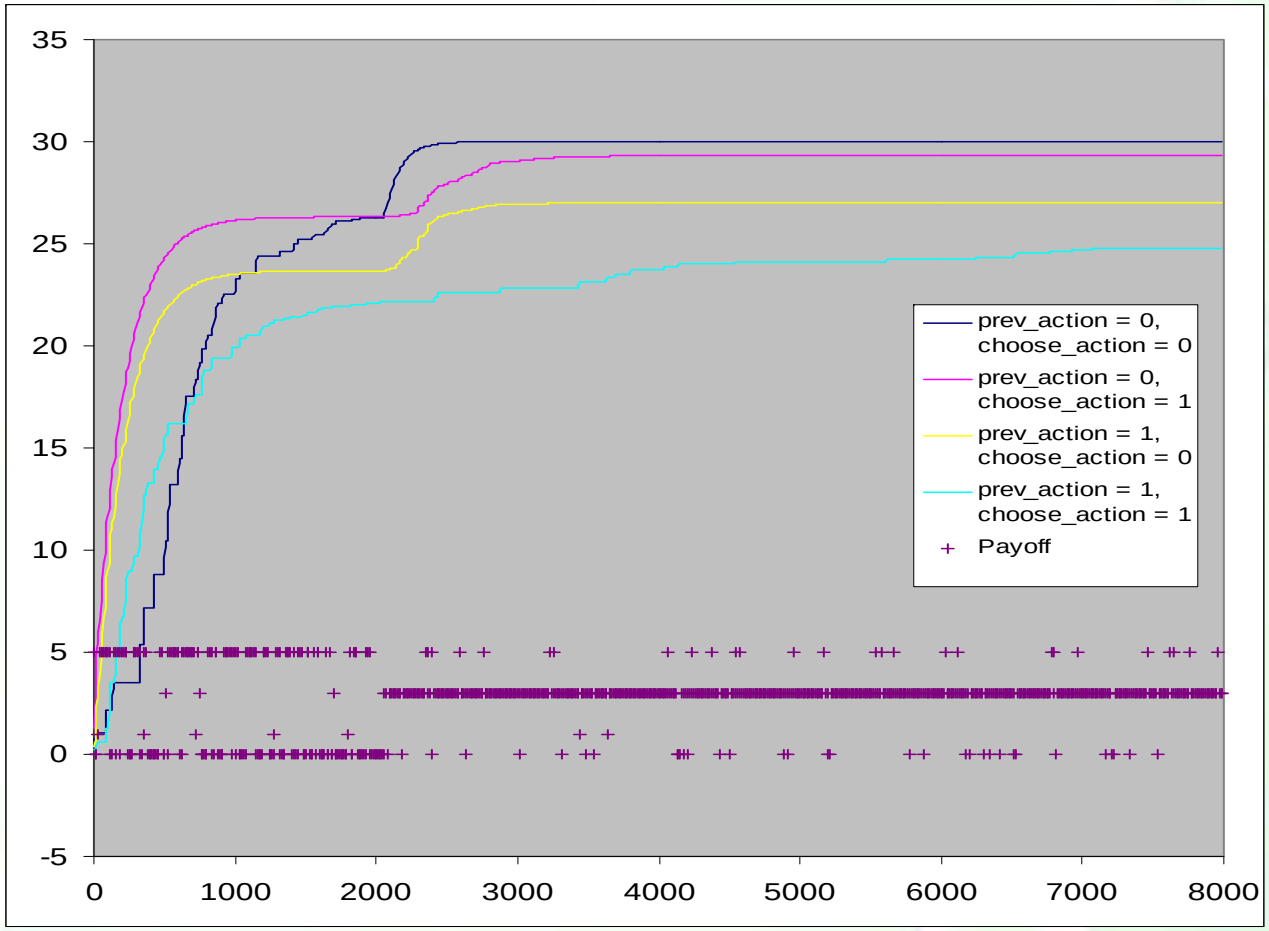
Prisoner Graphs - Gu-Q_0



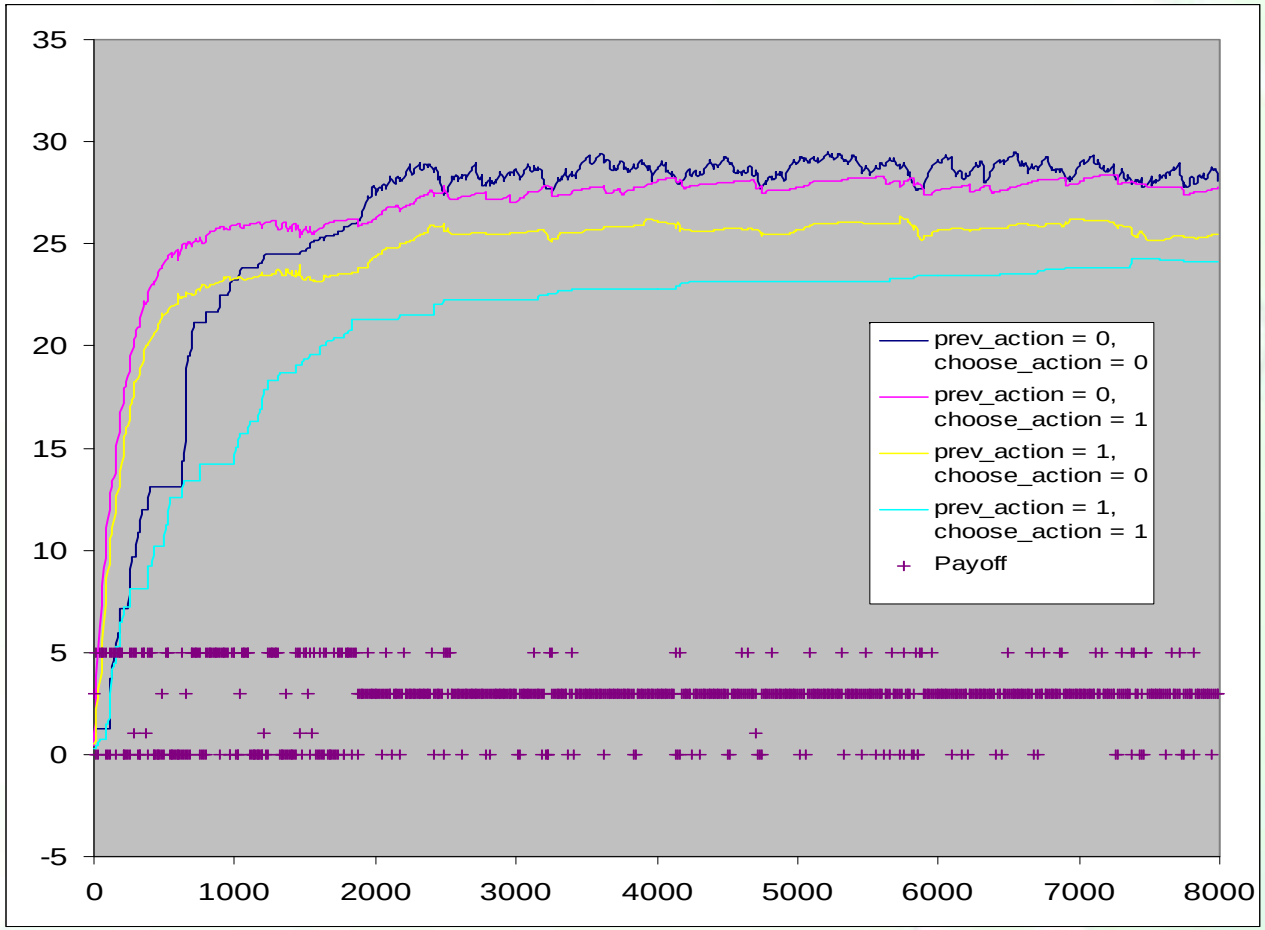
Prisoner Graphs - $Gn\text{-}Q_0$



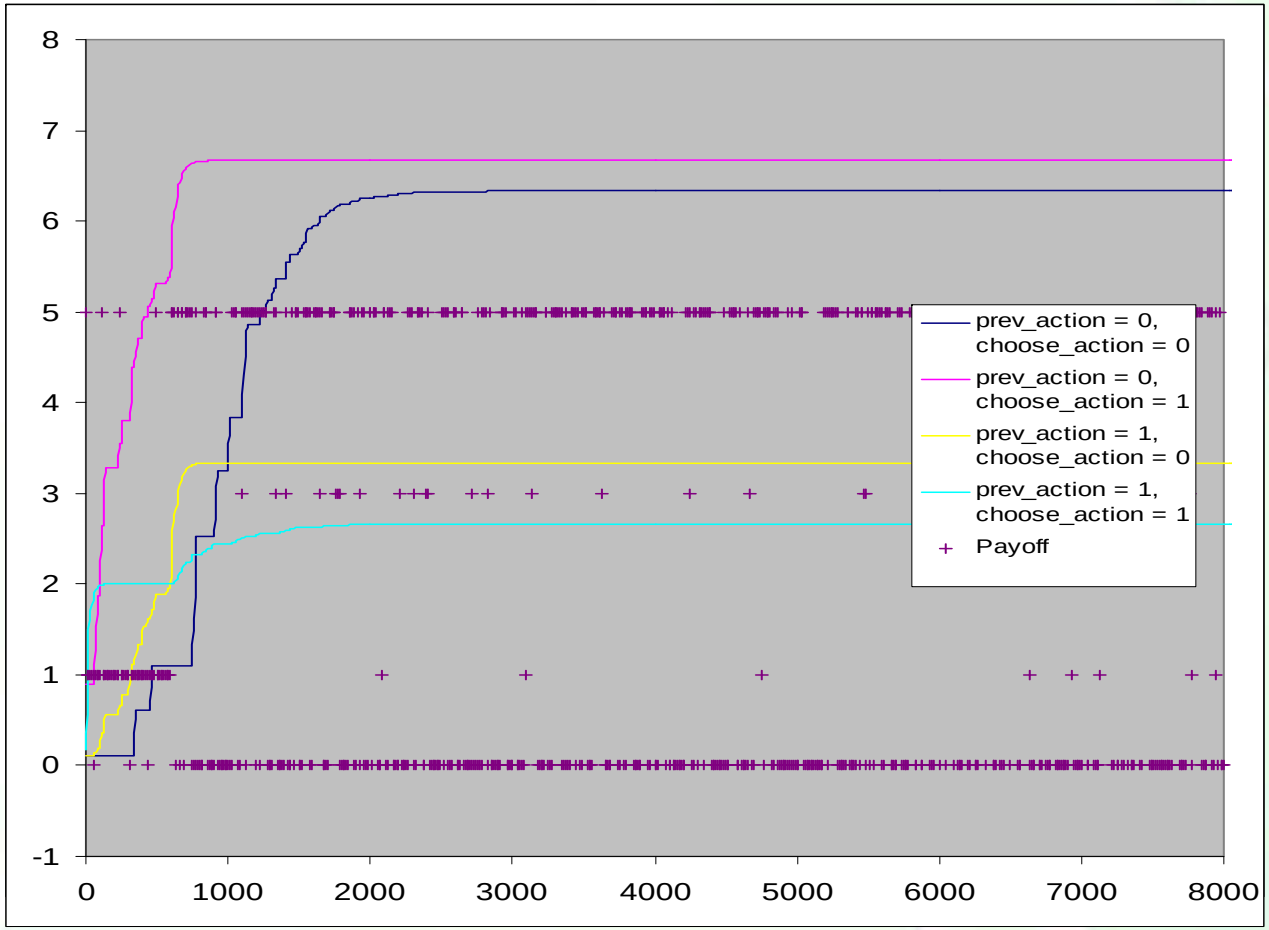
Prisoner Graphs - $Gf\text{-}Q_1$



Prisoner Graphs - Gu-Q_1



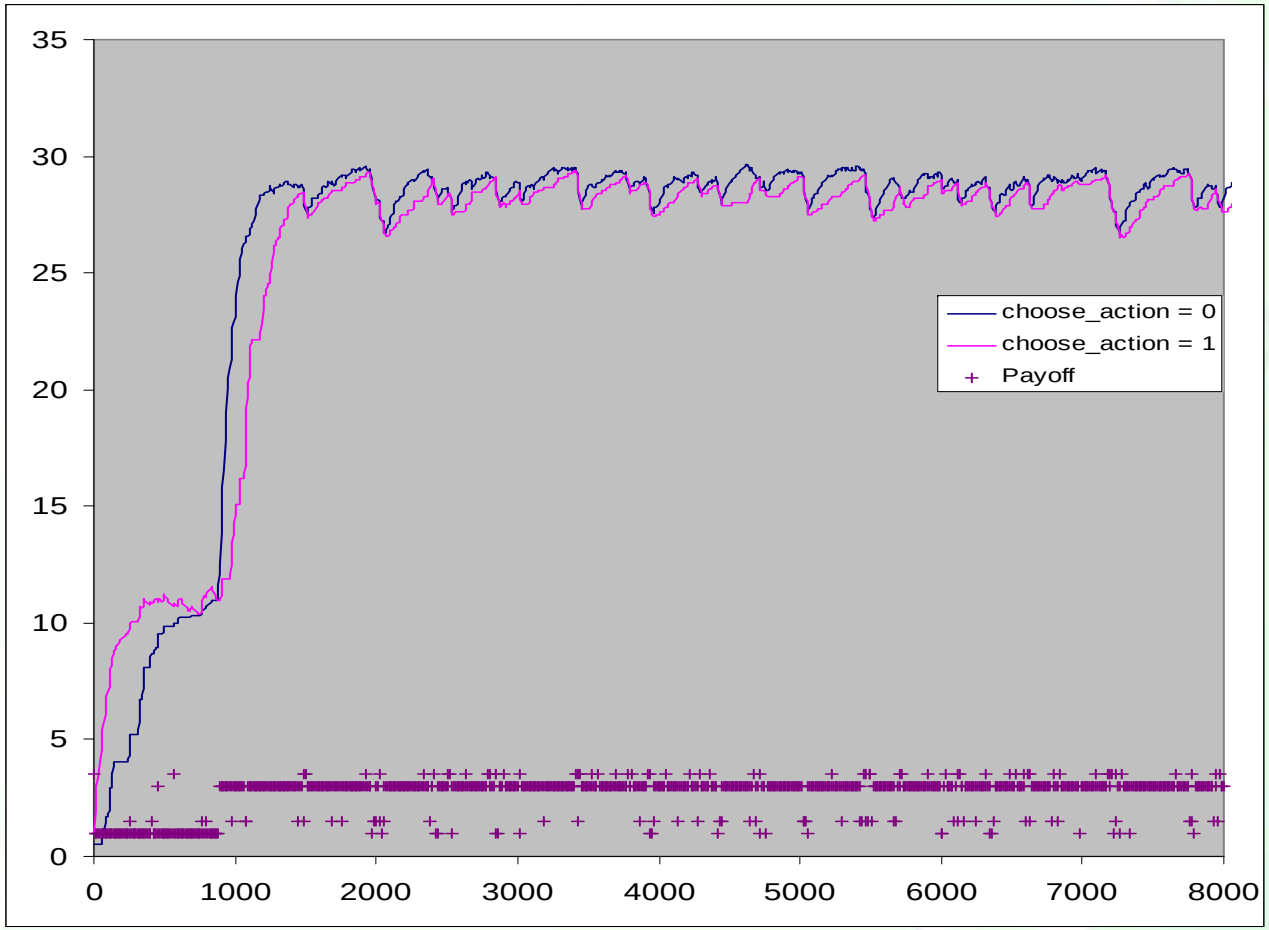
Prisoner Graphs - Gf-Q_1



alpha = .1 gamma = .5

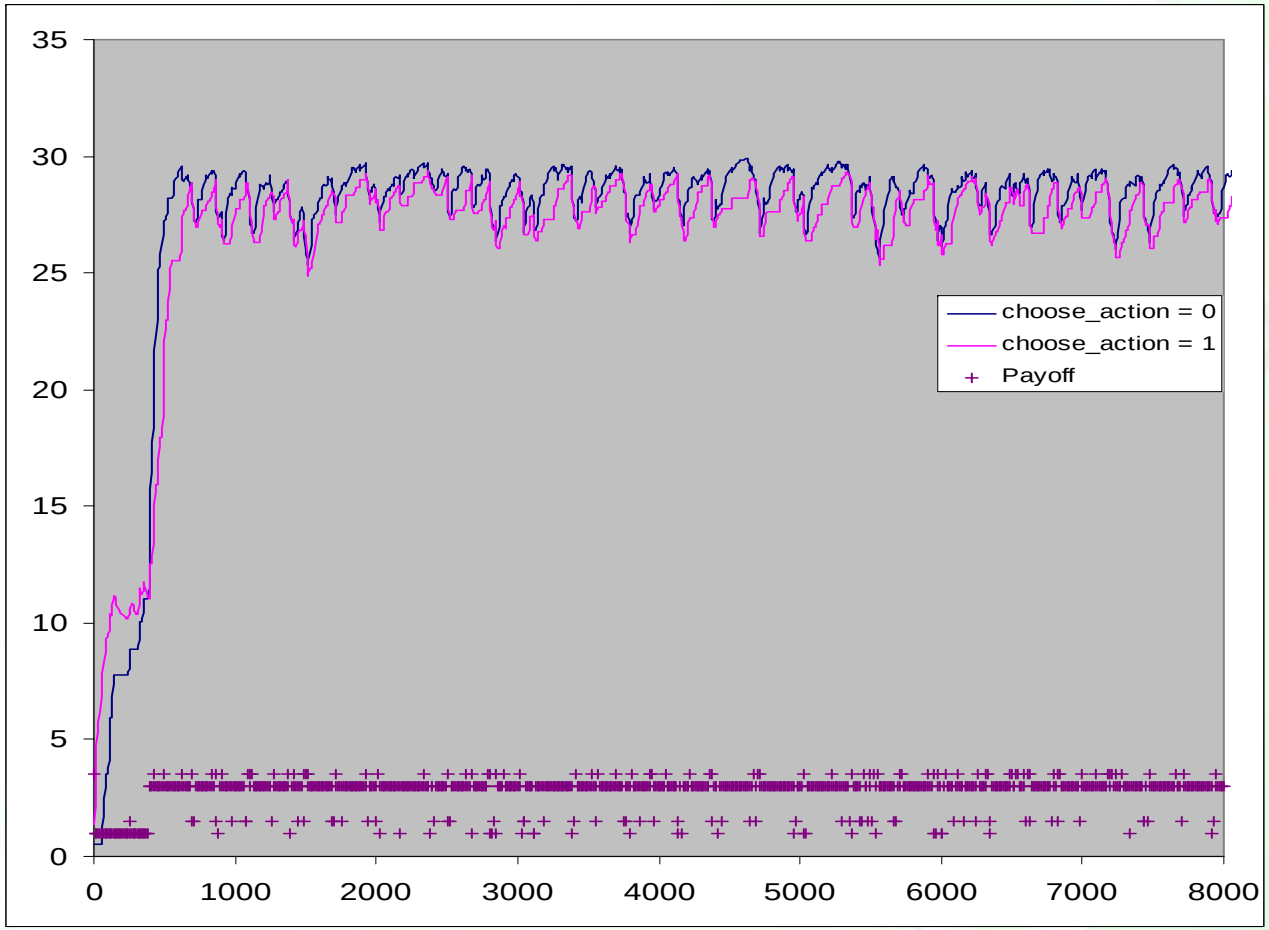
Prisoner Graphs - Gn-Q_1

Chicken Graphs - $Gf\text{-}Q_0$



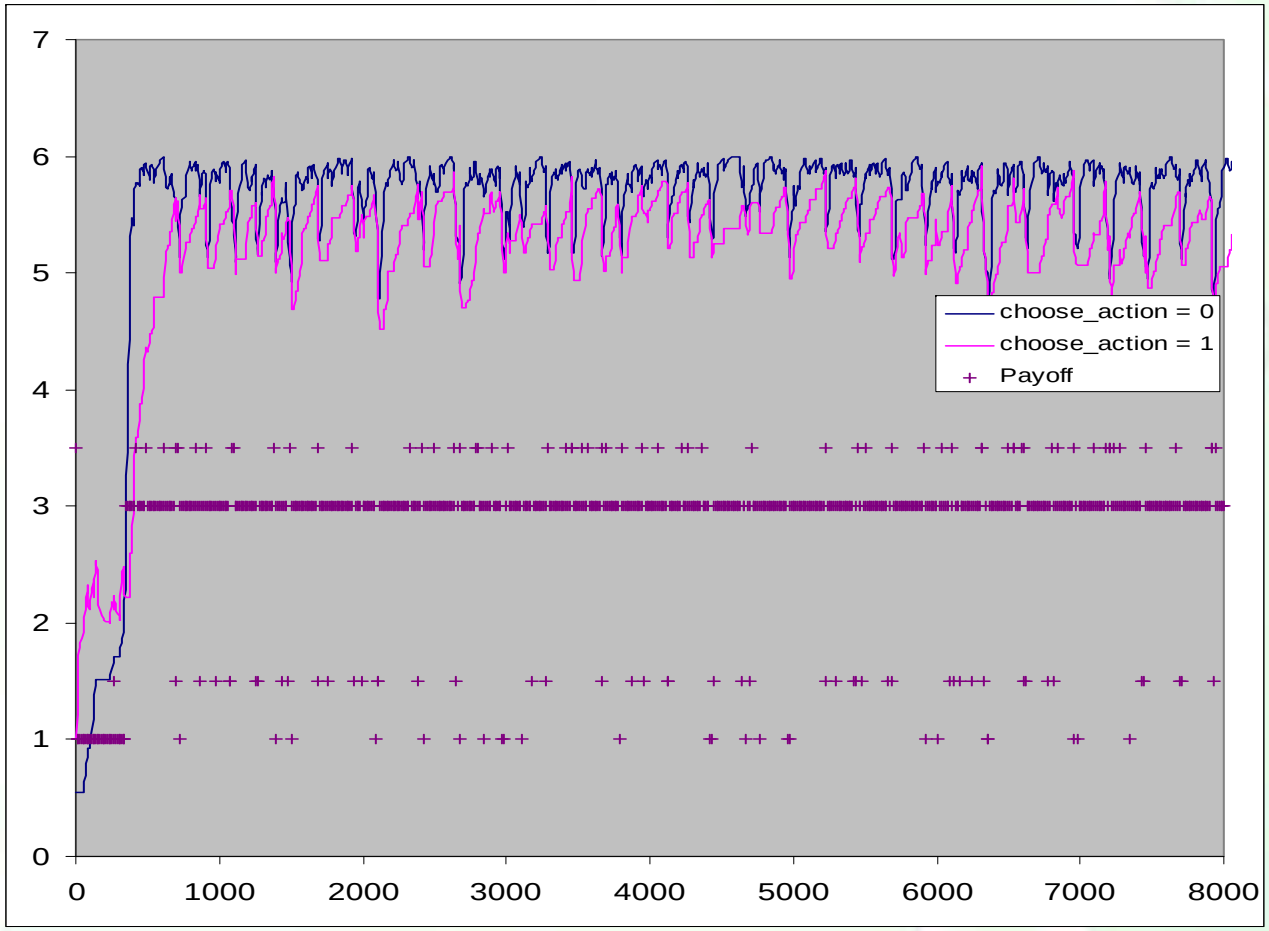
alpha = .1 gamma = .9

Chicken Graphs - $Gf\text{-}Q_0$



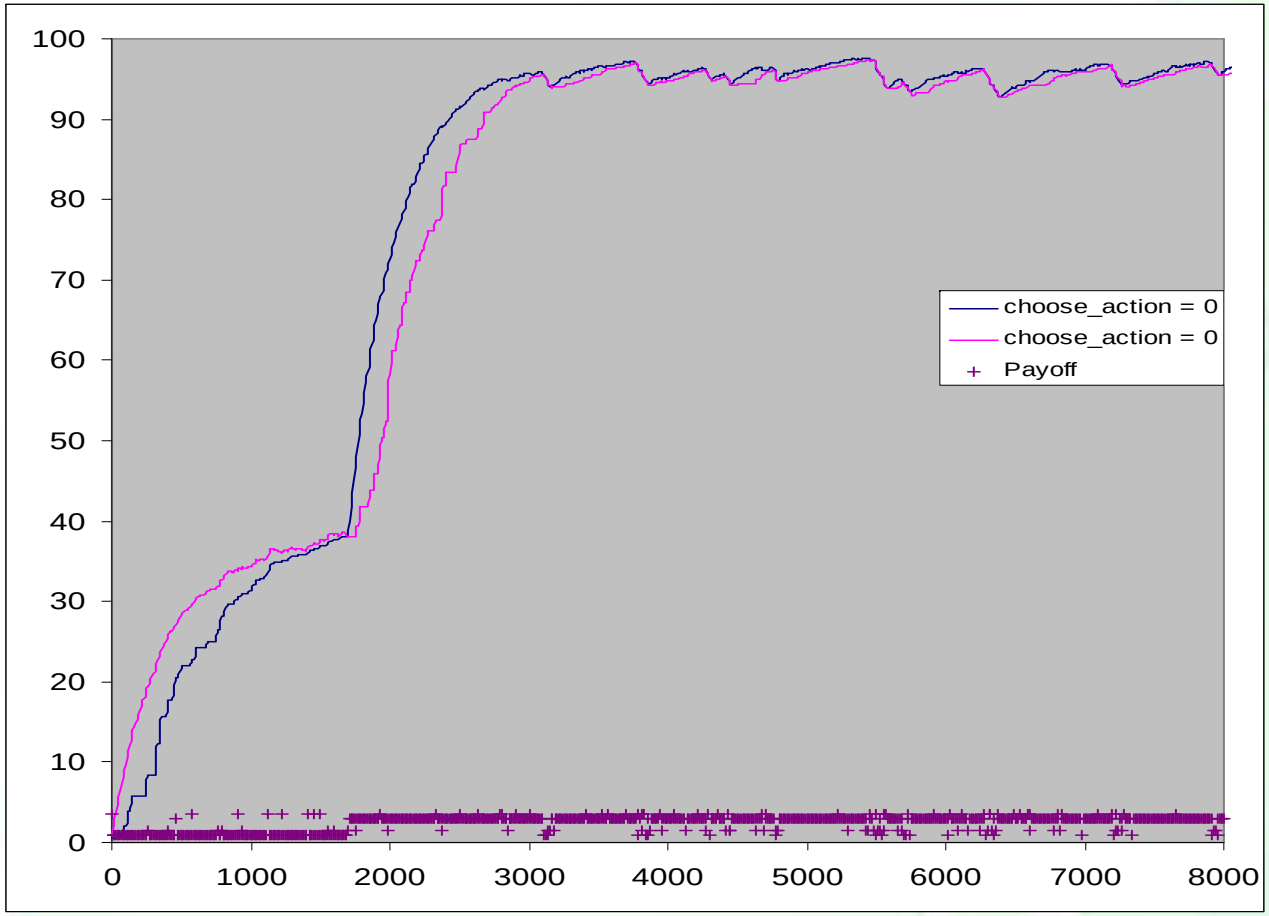
alpha = .2 gamma = .9

Chicken Graphs - $Gf\text{-}Q_0$



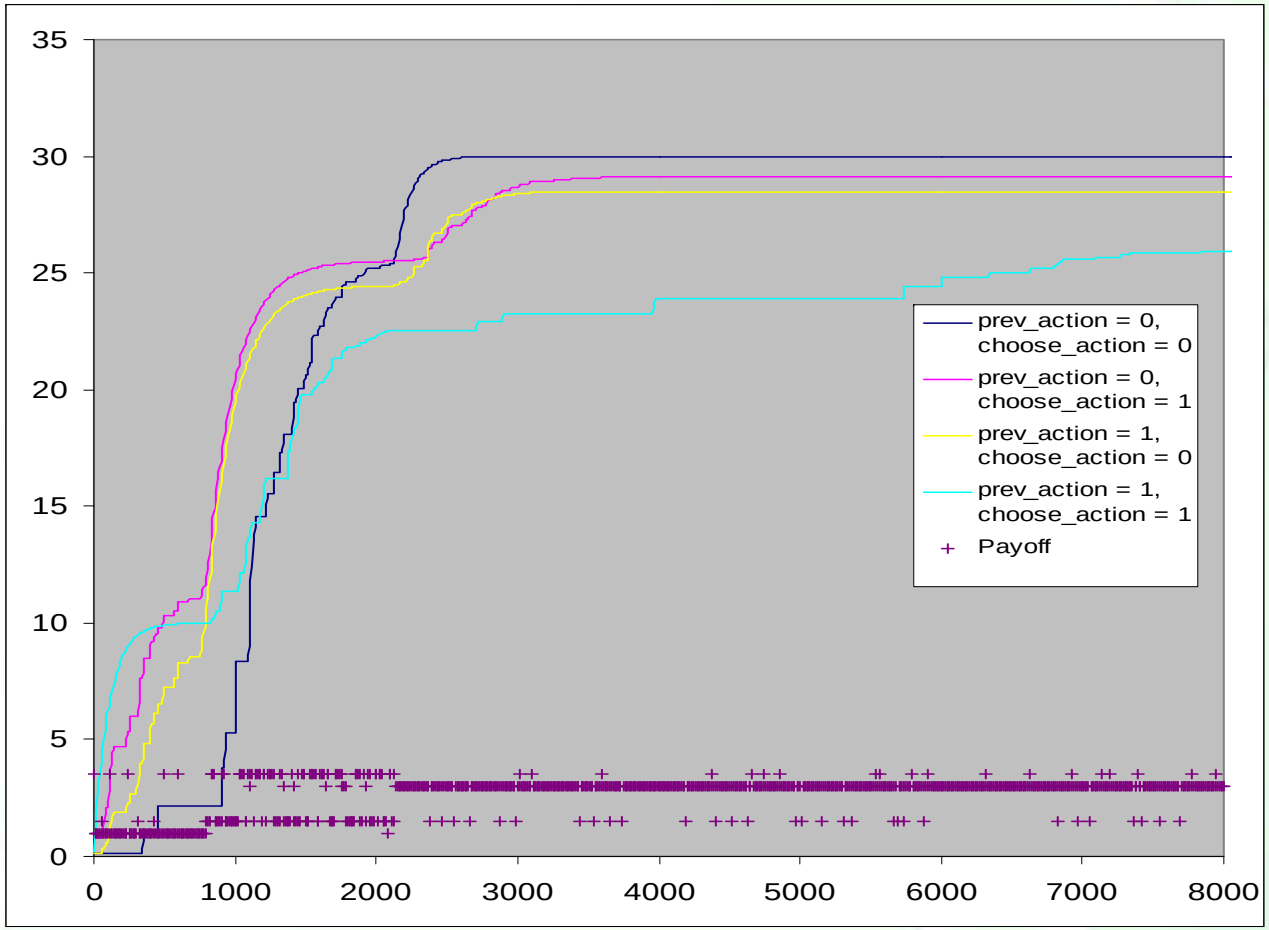
alpha = .1 gamma = .5

Chicken Graphs - Gf-Q_0



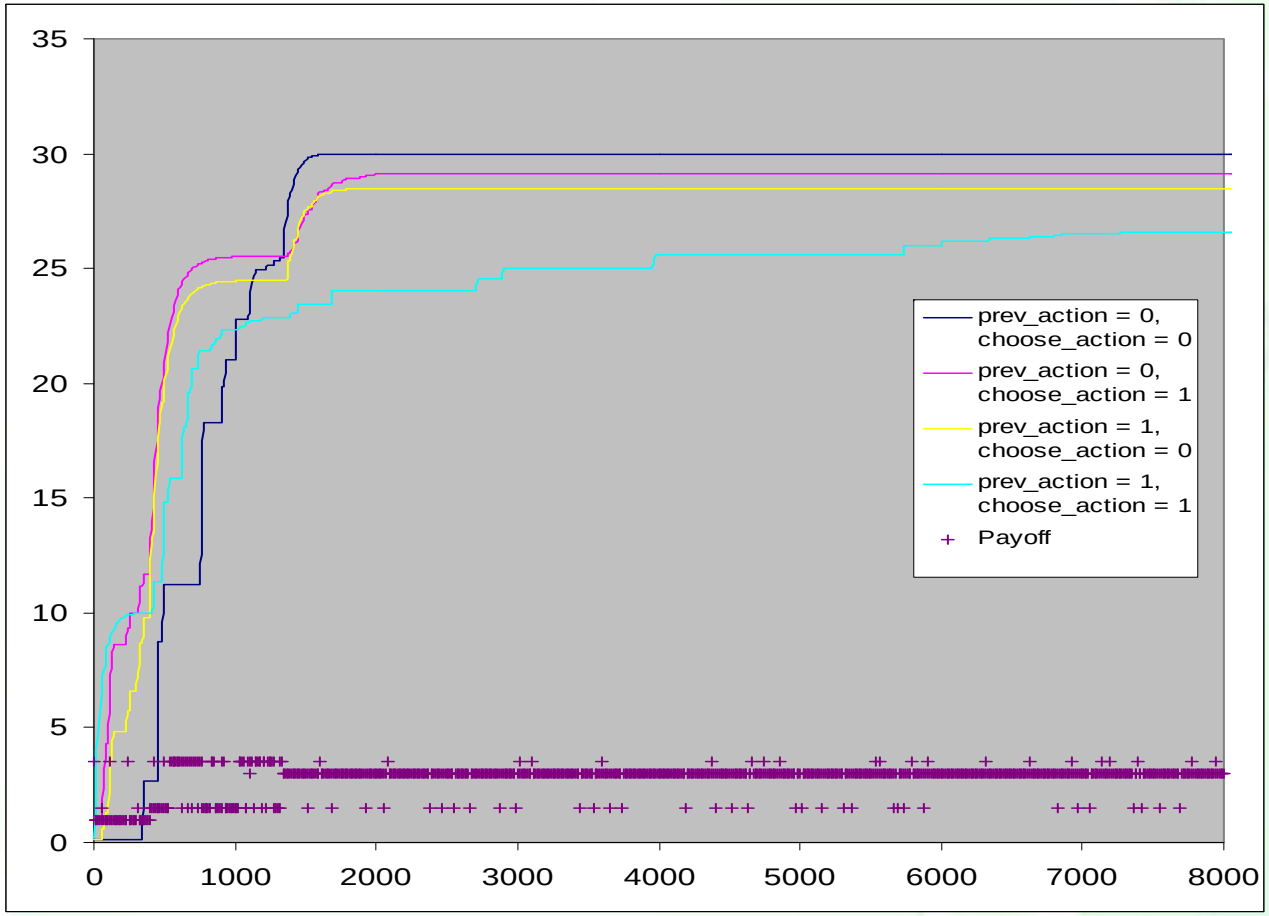
alpha = .1 gamma = .97

Chicken Graphs - Gf-Q_1



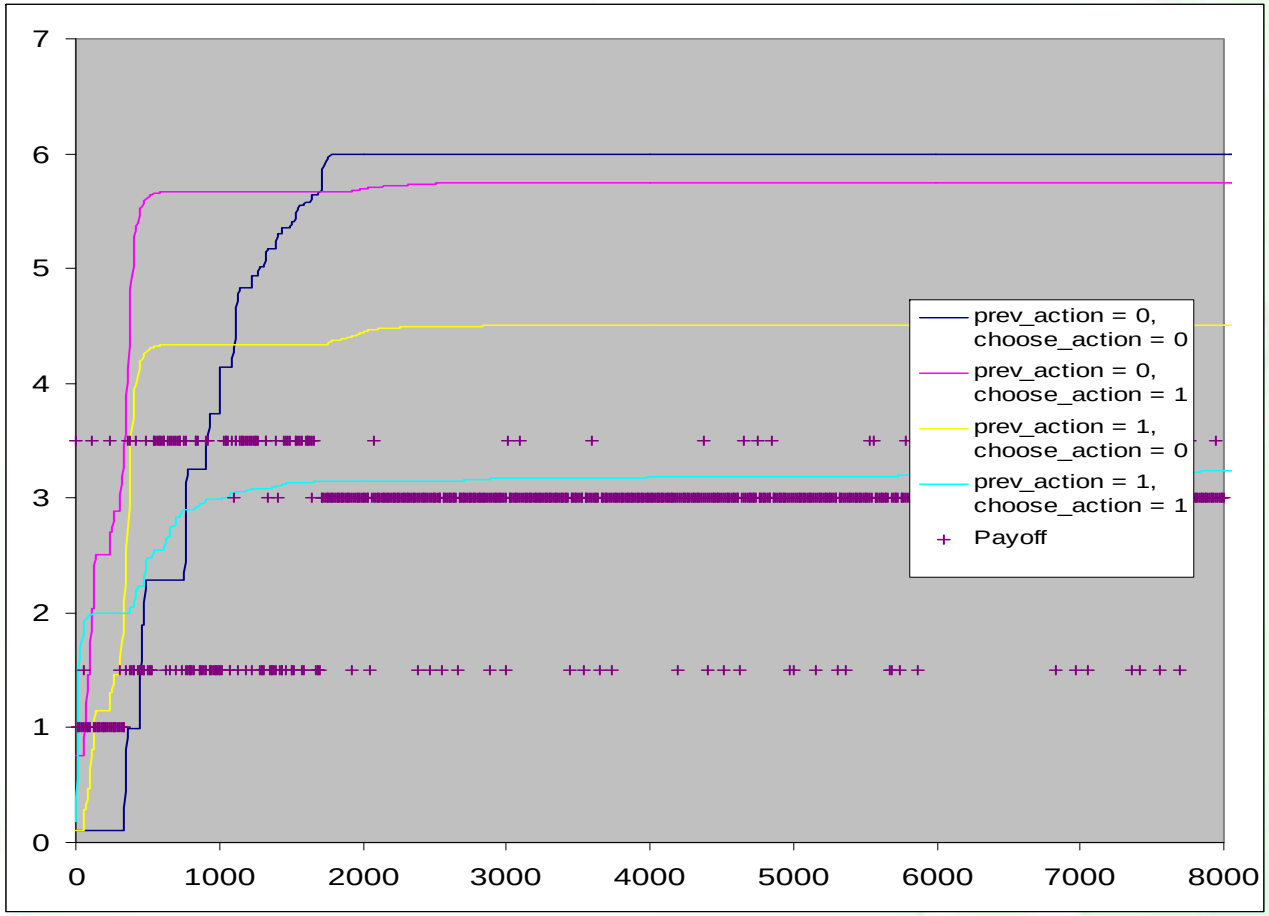
alpha = .1 gamma = .9

Chicken Graphs - $Gf\text{-}Q_1$



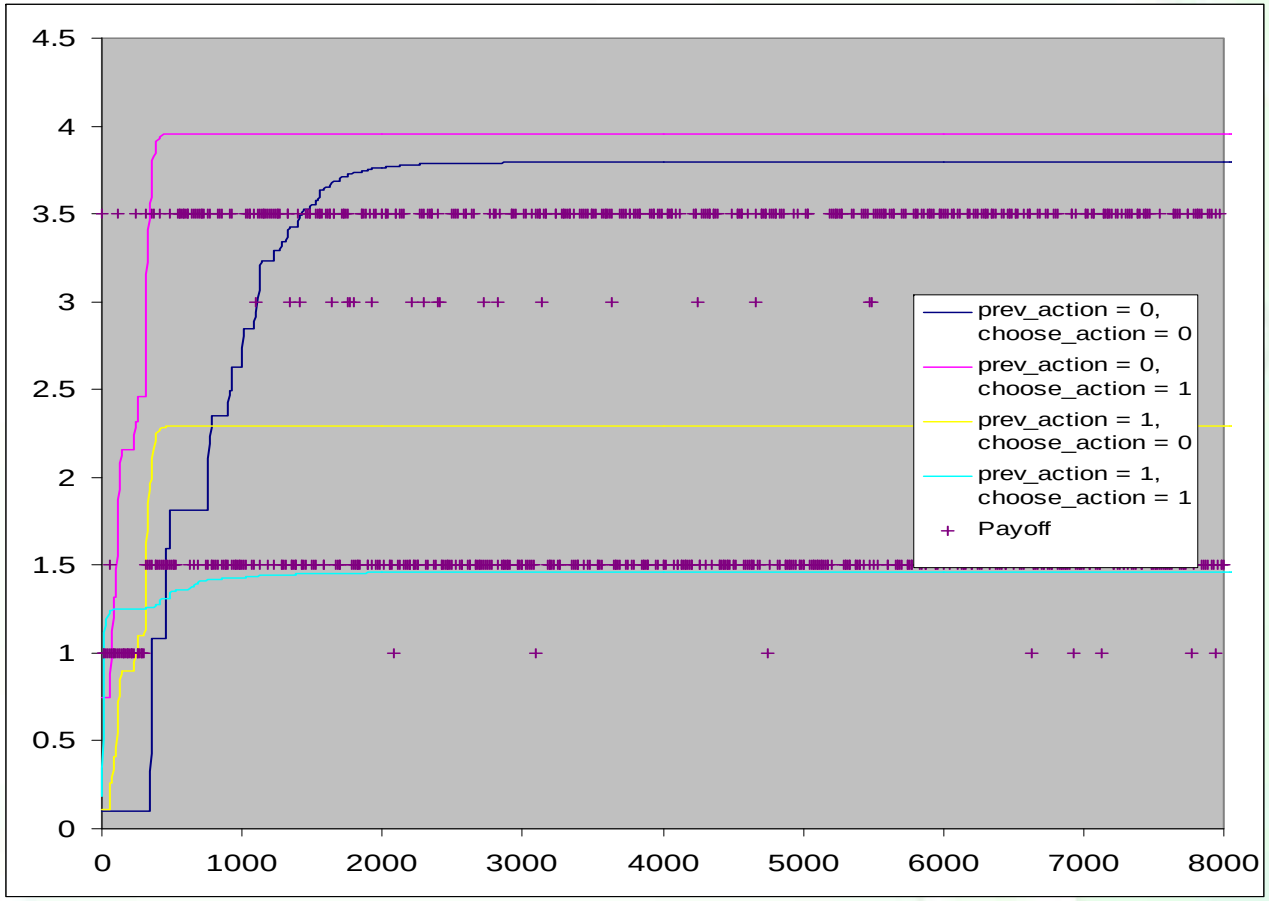
alpha = .2 gamma = .9

Chicken Graphs - $Gf\text{-}Q_1$



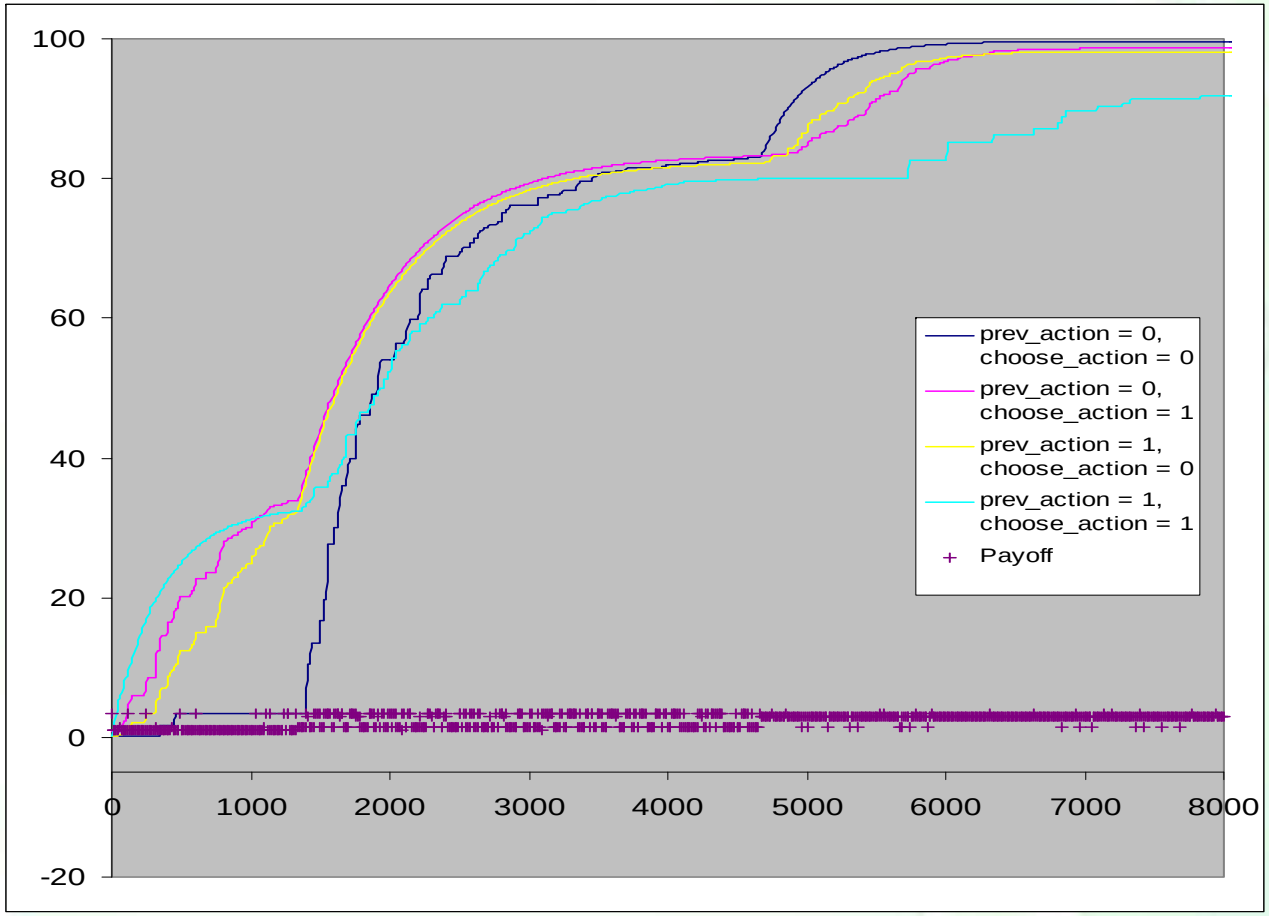
alpha = .1 gamma = .5

Chicken Graphs - Gf-Q_1



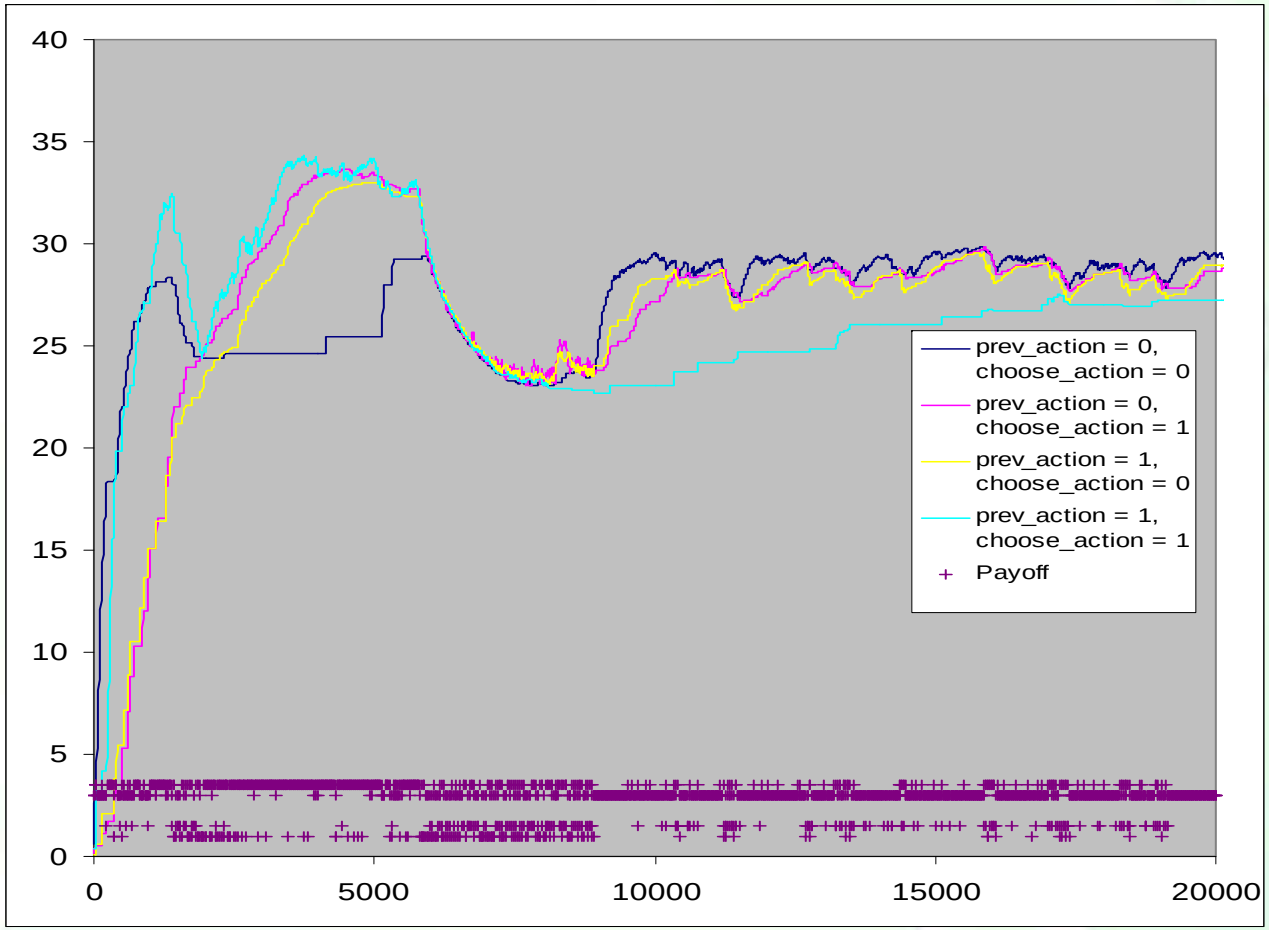
alpha = .1 gamma = .2

Chicken Graphs - Gf-Q_1

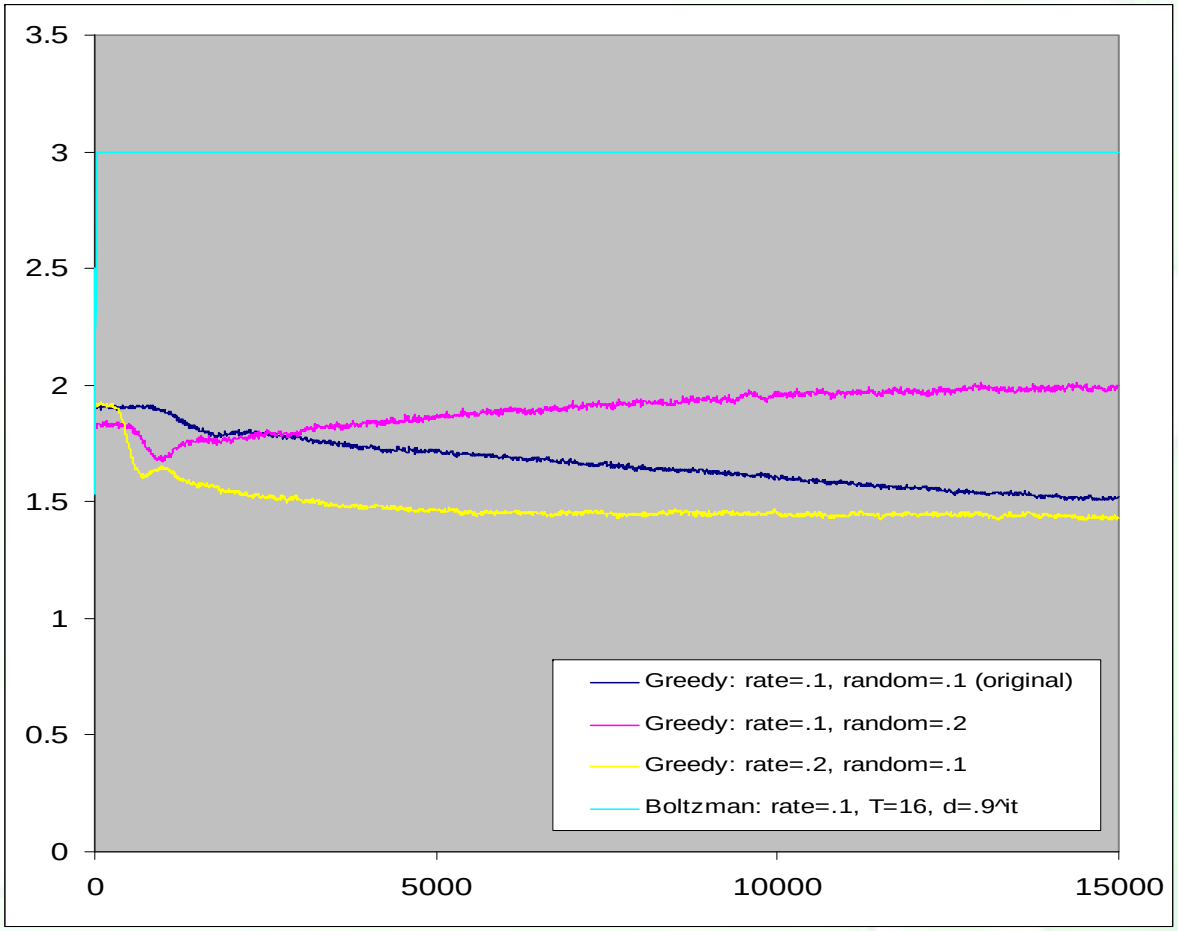


alpha = .1 gamma = .97

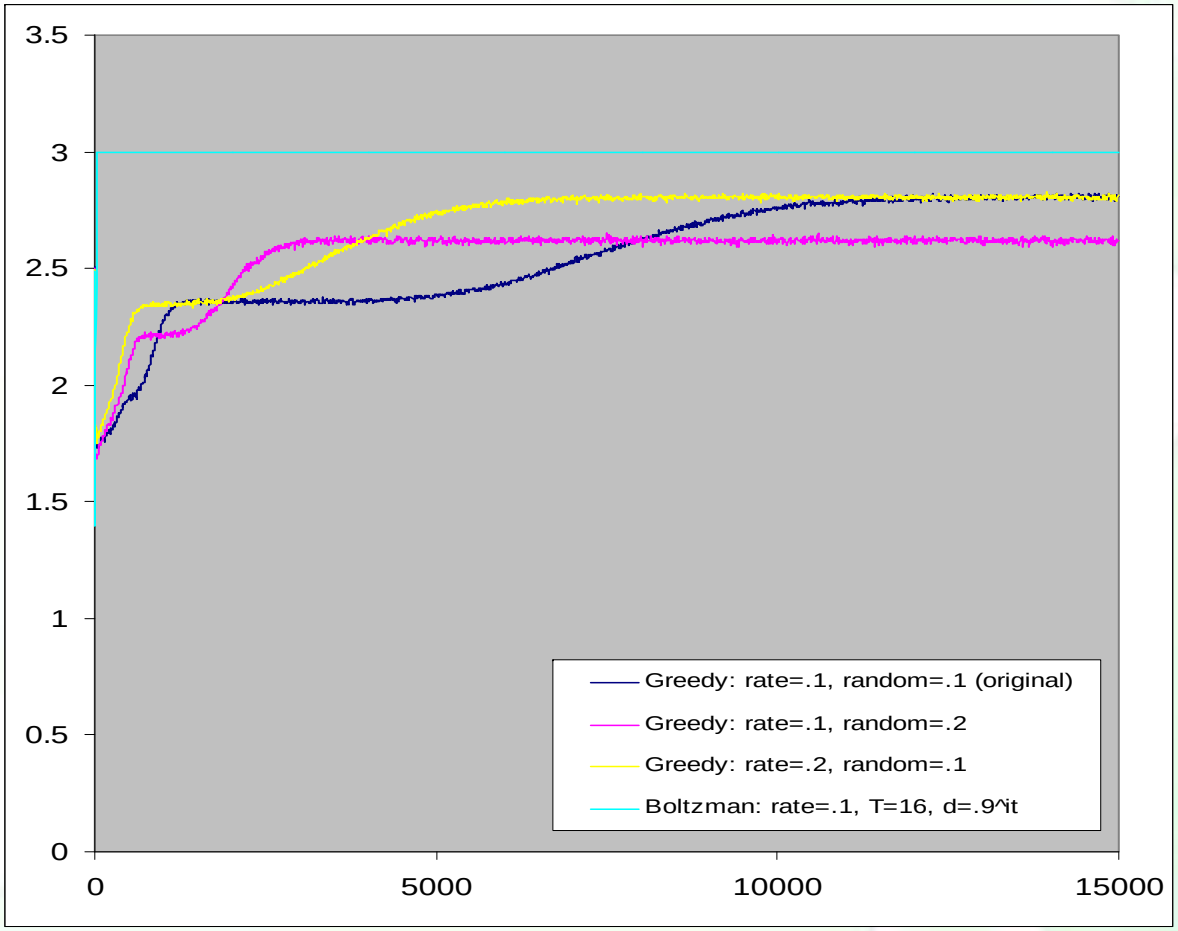
Chicken Graphs - Q_1-Q_1



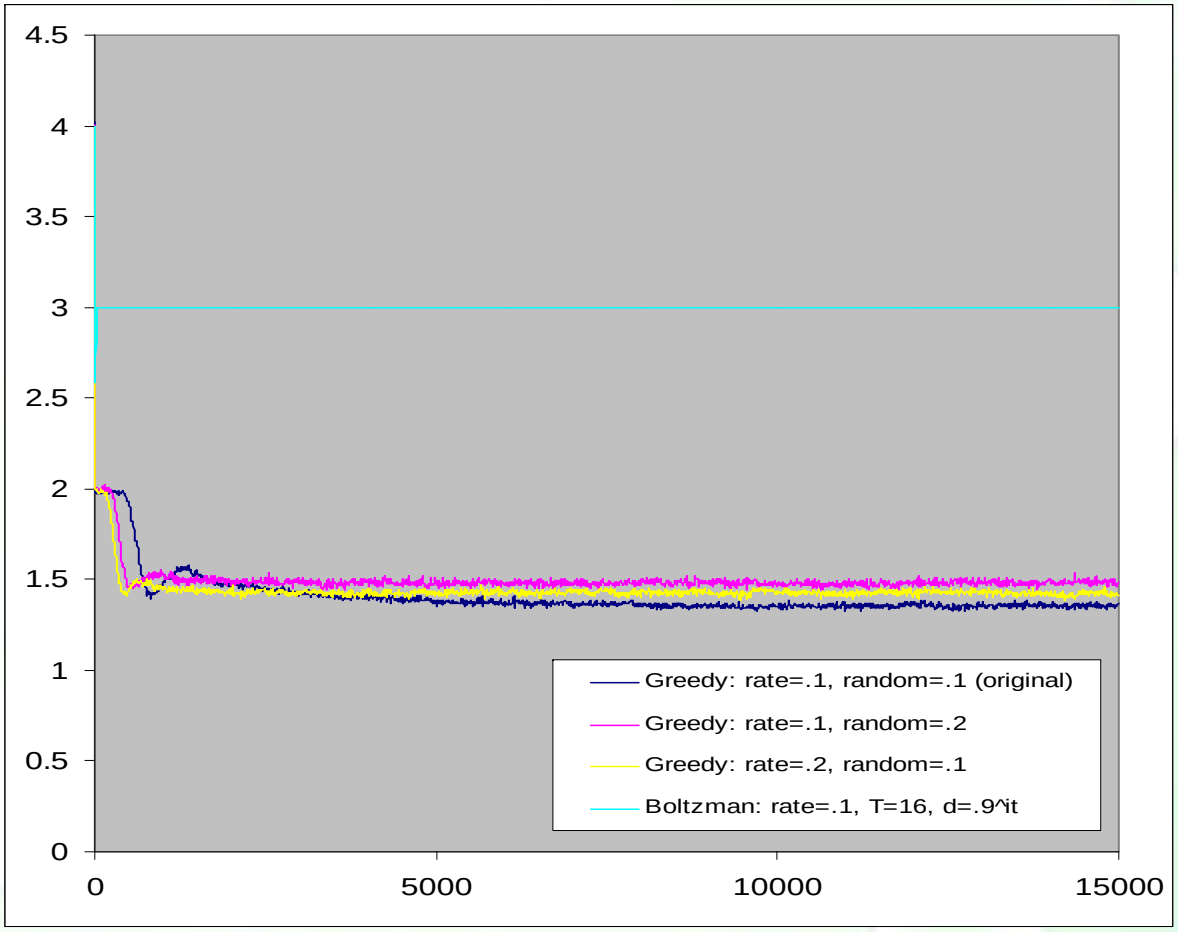
Assurance Average – Gf-Q_0



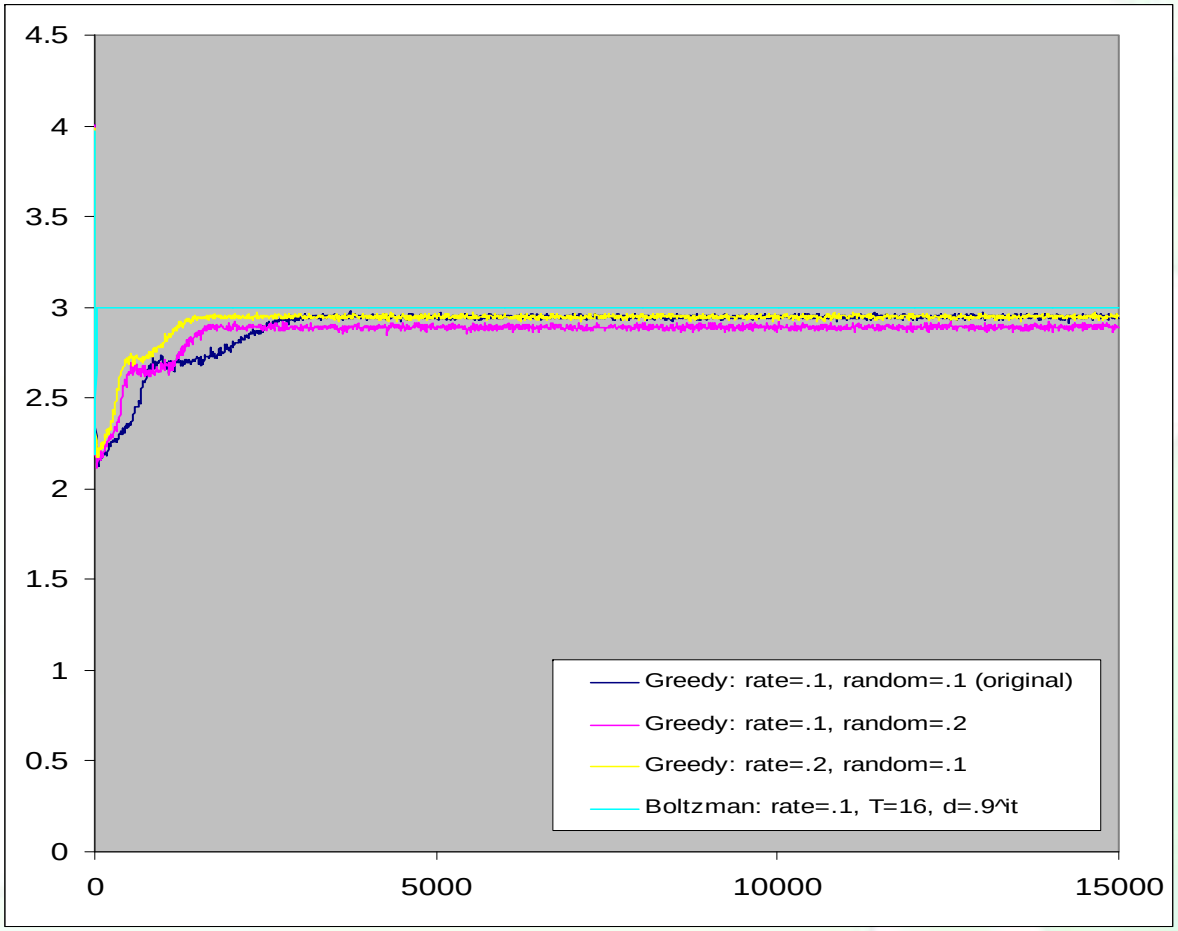
Assurance Average – Gf-Q_1



Prisoner Average – Gf-Q_0



Prisoner Average – Gf-Q_1



General-Sum Games

- Deadlock

$$M_1 = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}$$

- Prisoner's Dilemma

$$M_1 = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix}$$

- Assurance

$$M_1 = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}$$

- Chicken

$$M_1 = \begin{bmatrix} 3.0 & 1.5 \\ 3.5 & 1.0 \end{bmatrix}$$

General-Sum Games

- Deadlock

$$M_1 = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}$$

Best choice:

- Always cooperate

- Assurance

$$M_1 = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix}$$

Best choice:

- Match other player's action