Technical University of Crete

Reinforcement Learning & Dynamic Optimization - plh423

Simple Poker Game

Presented By:
Georgios Skoulas 2018030148
Ioannis Peridis 2018030069

Environment Implementation

State Space

Deck: 20 cards : T,J,Q,K,A & 4 suits

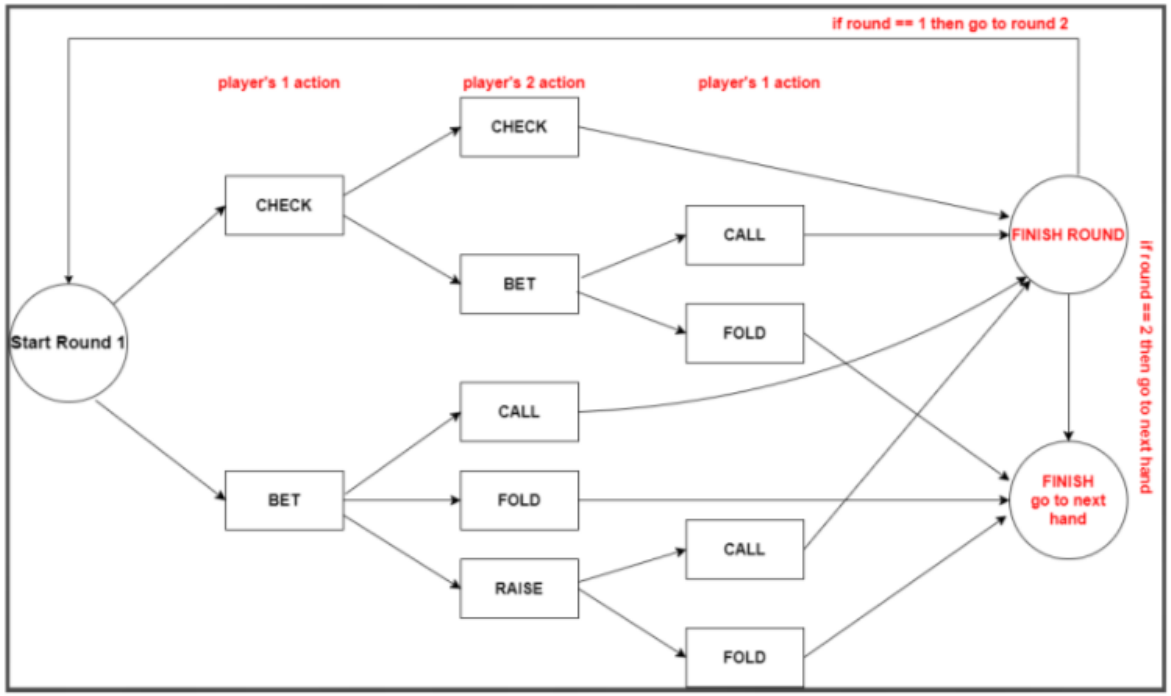
Player Hand: 1 card, Board: 2 cards

Blinds: 0,5 mandatory, Stack: 20 blinds

Actions

Fold, Check, Call, Bet, Raise

Actions Flow Diagram representing one game hand:



Heroes

Policy Iteration

Q-Learning

Villains

Random

Threshold Loose (aggressive)

Threshold Tight (passive)

Poker as a MDP

Poker as a Markov Decision Proccess

Actions: (0) fold, (1) check, (2) call, (3) bet, (4) raise

States: devided by round 1 or 2 and by hand strength

Total 21 states:

S(0-4) : round 1 states: T=s0, J=s1, Q=s2 ,K=s3, A=s4

S(5-19) : round 2 states: high card=s5-s9, one pair=s10-s14, set = s15-s19

S20 = terminal state

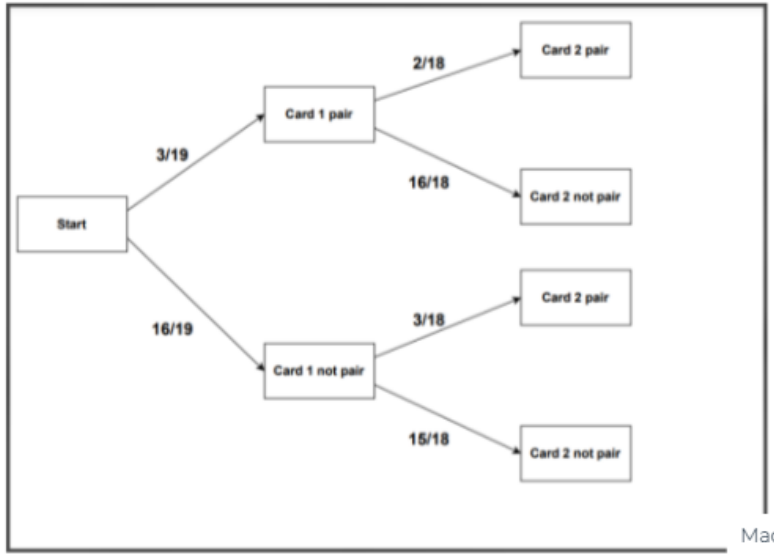
Transition Probabilities:

From S0-4 to S5-9: P=0,668

From S0-4 to S10-14: P=0,315

From S0-4 to S15-19: P=0,017

From S5-19 to S20: P=1



Policy Iteration

Strategy vs Opponents:

- vs Threshold Loose
  - Play high strength hands
  - Tight/Passive actions
  - Win few big pots
- vs Threshold Tight/Random
  - Play all hands
  - Aggressive actions/Bluffs
  - Win many small pots

Convergence:

Converges when new policy equals to old policy
Convergence after 2-3 iterations

Optimality:

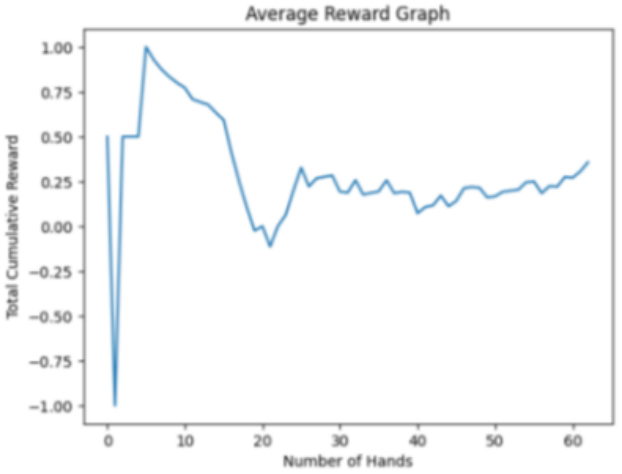
Optimal solution occurs from the reward system
PI agent decisions match the optimal policy

Policy Iteration - Rewards

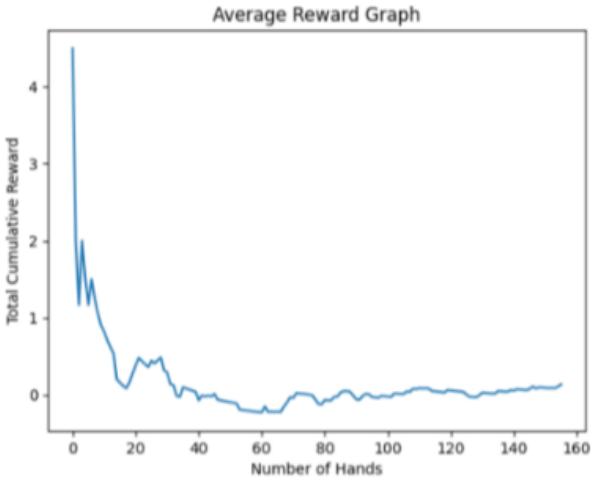
Parameters: discount factor (γ) = 0.98 convergence error (ε) = 1e-10

Average Reward vs Opponents

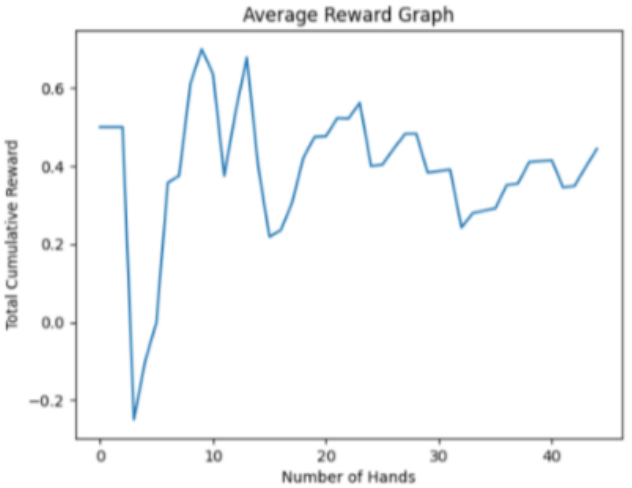
vs Random:



vs Threshold_Loose:



vs Threshold_Tight:



Experimental Results (10 games)

Win ratio:	Average #Hands:	Avg.Avg Blinds/Hand ratio:	Win ratio:	Average #Hands:	Avg.Avg Blinds/Hand ratio:	Win ratio:	Average #Hands:	Avg.Avg Blinds/Hand ratio:
100%	42,4	0,64	100%	73,3	0,49	100%	33	0,65


Q-Learning
Made with VISME

Q-Learning

Parameters

discount factor (γ) = 0.9
learning rate (α) = 0.1
exploration rate (ε) = episode^(-1/4) reduces over time
number of episodes = 20000

Convergence:

Convergence time is different against each opponent
Most time: vs threshold loose
Propably beacuse we lose from threshold loose

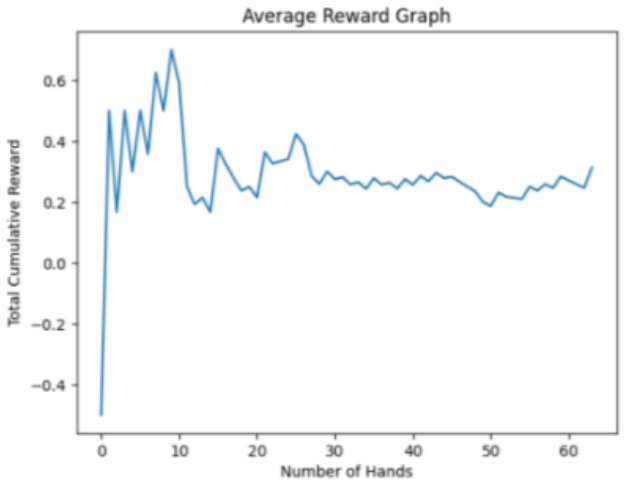
Optimality:

Tuning the parameters in different ways gives different results
By experimenting we chose parameters that gave the best results

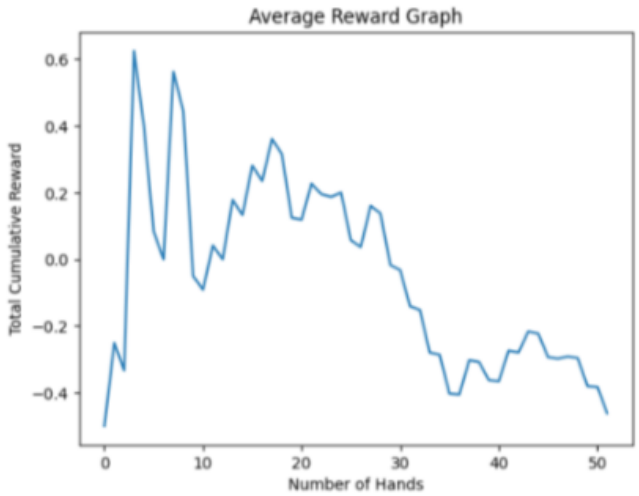
Q-Learning - Rewards

Average Reward vs Opponents

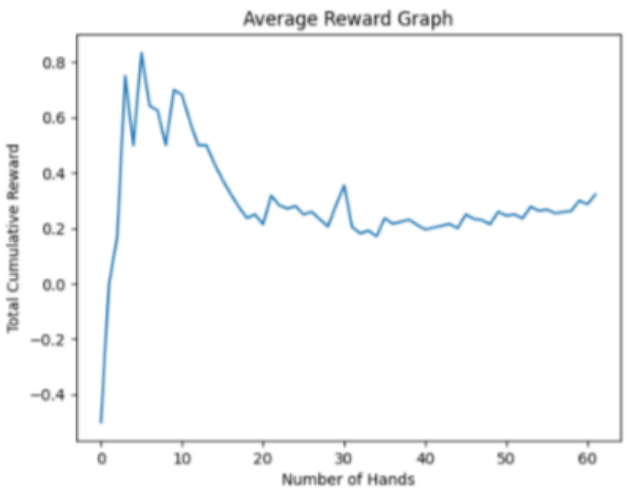
vs Random:



vs Threshold_Loose:



vs Threshold_Tight:



Experimental Results (10 games)

Win ratio:	Average #Hands:	Avg.Avg Blinds/Hand ratio:	Win ratio:	Average #Hands:	Avg.Avg Blinds/Hand ratio:	Win ratio:	Average #Hands:	Avg.Avg Blinds/Hand ratio:
100%	82,5	0,27	0%	40,6	negative0,5	90%	97,5	0,22

Problem

Q-Learning vs threshold loose

The agent loses to the threshold-loose bot

We could not find if there was some problem with the code
We assume maybe that the state space was not big enough