Parley Pacheco Martins 1484000
AUCSC 460 – Artificial Intelligence
Winter 2016
Department of Science
University of Alberta, Augustana Faculty

Report Rock Paper Scissors

1 Game Agent

The first dumb agent (SimpleCycleAgent) is just a simple cycle made to beat the Rock, Paper, Scissors cycle. It doesn't react to the other agent and always play this cycle. This agent was made just to see the behavior (of both of the agents) when such a simple one is playing.

The second agent (BiggerCycleAgent) is another dumb cycle, but bigger. It chooses a random cycle size from 1000 to 10000 and randomly populates the cycle. Once it's done, it just iterates through the cycle, without any reaction. This was done to improve the challenge of finding a cycle, since this can be really tricky for other agents. Depending on the size of the cycle it's not really different from the RandomAgent.

The third agent (StatisticAgent) is "smarter", in the sense that it reacts to the other player. It counts the frequency of a movement and plays the action that would beat the most played action by the oponent. This was actually the first idea, but I think it would have been more successful against humans, since we are horrible at ramdonness.

2 Results

The three agents had the following results. These values are the mean value of three tournaments of 1000000 rounds each:

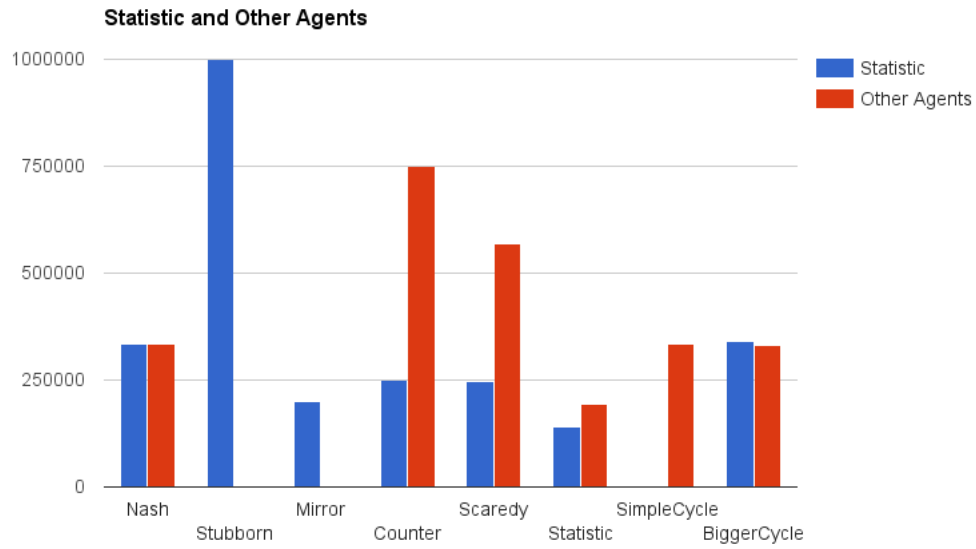

Figure 1: StatisticAgent performance

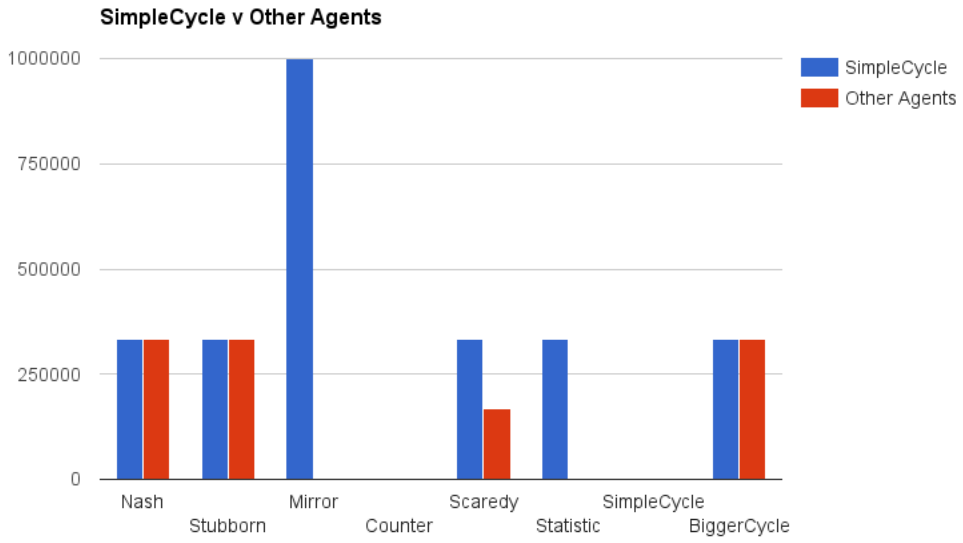

Figure 2: SimpleCycleAgent performance

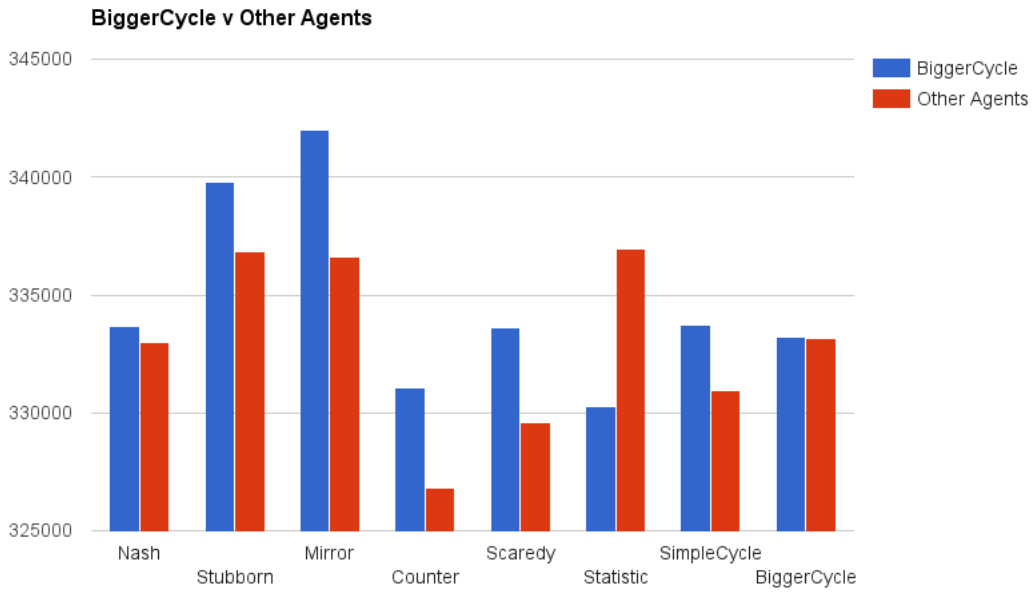

Figure 3: BiggerCycleAgent performance

3 Performance Discussion

When I first wrote the StatisticAgent, I thought that would be a good policy. However, as shown in figure 1, it was a clear disaster against agentslike Scaredy and Counter. Even against the mirror agent it didn't have a very good performance, ending up in tied games most of the times. That happened because of the reaction time of this agent. The only moment it changed its action was when the opponent most played action changed.

The Statistic showed its flaws especially against the SimpleCycle. With such a simple policy, without even react to the oponnents hand, this agent confused made the Statistic play Paper only when it also played Paper.

The SimpleCycleAgent was written just to check the behavior of all the other agents. I imagined that (almost) any policy would beat a simple Paper, Scissors, Rock cycle. It was made to win only in these conditions. However it showed a surpringly good behavior.

Observing figure 2, we see interesting performance against MirrorAgent and CounterAgent. That happened because Mirror generated the very cycle that Simple was suppose to defeat, while Counter generated the same cycle as Simple.

The BiggerCycle was an "improvement" based on the SimpleCycle. Any agent that tried to count the cycle could have beaten SimpleCycle, so I decided to make a bigger cycle. That would be almost the same as playing against the random agent. It became, indeed,

an improvement achieving the equilibrium of 33% of victories against all the other agents. That can be notice in the graph if you see that the biggest difference between win and lose is of 500 (out of 100000) rounds.

4 Other Approaches

There are several other approaches to this problem. One of them would be tring to find cycles and play the cycle that beats that. However, as I see, this would be a big problem. First of all you would have to assume a size or count the movements of your opponent. You would also have to store all the action taken and check the next one against all of them everytime. When would the cycle be done? The most important question in this approach is how to be sure that I'm not in a subcyle?

Maybe a better way of doing this would be by statistic (Intuitevaly, but not related to all rounds. Checking against a smaller total (maybe the last five, for example) and checking the most used in this interval seems more effective than what I did for the statistic, because it could have a big response time depending on the opponent's pollicy.

5 Conclusion

It's interesting to see how complex an Artifical Intelegence has to be even for such a small example as playing Rock-Paper-Scissors. This showed how important it is to