Computation and Cognition - EX8 - Game Theory

Analytic Section

A. The payoff matrix is:

First hunter | Second hunter ->	Hare	Deer
Hare	(2,2)	(2,0)
Deer	(0,2)	(4,4)

B. In this game, there exists 2 Nash equilibrium points. When one player chooses the non deterministic strategy - hunt the deer, the other should choose the same to maximize reward. Let us assume one player chooses the other option - to hunt the hare. Thus, the other player would have to follow in order to change his reward from 0 to 2. Hence the points discussed above are (0,0), (1,1) probability wise.

C. Let p be the probability for first hunter hunts a deer. Let p2 be the probability for second hunter chooses deer. Observe the conditional expectation for the VR reward where first hunter hunter the deer:

$$E(reward) \ = \ 4p * p2 \ + \ 2(1-p2) \ => \ E(reward) \ = \ (4p-2) * p2 \ + 2$$

With p > ½ the reward expectation function with respect to p2 is monotonically increasing, and so to maximize reward second hunter should hunt a deer as well (p2 = 1). With p = ½ function is constant we would infer that there is no reward function in that case. I.e. that is a Nash non-deterministic point - (½,½). With p < ½ function is monotonically decreasing and so second hunter is better off hunting the hare (p2 = 0)

Programming Section annotation

Referral to given strategies (section a. and b.)

I've simulated the iterated game for 20 and observed the average values. It is seen that both u1 and u2 are equal 3.5 approximately with no strategy winning over the other. This can be explained by the patterns of behavior defined by mostly the first strategy. It starts with defect on the first run, then randomly pick a choice from the previous round and returns it, this pattern is prone to reach an equilibrium. The second strategy pattern is a bit more tricky, initially it starts with defect. On the next rounds the strategy weighs the number of previous defects and cooperations performed by the rival - if it is equal it gambles, else it counts the rival's former decisions and chooses to act in accordance with the choice which was made on more rounds. The fact that a very simple strategy performs as good as the other demonstrates that the behavior does not need to be complex in order to beat the opponent.

My chosen strategy:

A good strategy should consider the following factors:

Needless to say it needs to act under the premises of striving to maximize its reward on **multiple** games, hence punishing your rival backlashes at one's own reward. That per se does not mean it should be a deterministic strategy, I think it is also important sometimes to surprise your rival is to include a random random factor with a certain threshold, in addition it achieves the goal of breaking any predicted patterns of behavior made by a rival. Nevertheless setting a higher threshold could be morph the strategy into a "crazy" player strategy which is utterly unpredictable, but a crazy strategy generates results with a great variance - not something we wish to count on in game theory. Thus, I think threshold should be generated empirically after trials. Moreover a strategy should consider the history in terms of long patterns of behavior and short term pattern. The long and short patterns should be treated in a very different manner. Finally, the strategy needs to treat the rival in a "quid pro quo" approach. If he misbehaves, it will punish him and if he cooperates, he should be treated with the same approach.

Considering the guidelines above, I will introduce the following strategy key parts, a more detailed explanation on every decision is appears in code comments.

I've defined the long term memory mostly on the last 50 iterations of the game. For each game after the long term threshold has passed I try to evaluate the rivals long term strategy and choose the preferable counter strategy. If the rival is defecting too many times and in particular if he is the first one to defect within that time frame, the strategy morphs into a more punishing approach. Otherwise, I respond in a more peaceful fashion. If the strategy detects that the rival does not change his strategy after defecting a few times I change to a more aggressive approach. If we see that rival responds to that we change back to a peaceful approach but we probe every now and then based on a random threshold. In addition the strategy detects a crazy behavior as mentioned earlier, if that is the case we switch to a very aggressive strategy. After testing it a few times.