Assignment 8

Models of Cooperation in Social Systems

Write a report on the tasks given below.

Task 1(15 points)

1. Download *PD-Two-Person-Iterated.nlogo*. (This version fixes a bug that was in the original Models Library version.) Set the `computer-strategy` to "random" (i.e., randomly cooperate or defect). What are the approximate average scores of the human and the computer when the `human-strategy` is set to each of the six possible strategies? (Record the approximate average score for each after about 300 iterations.) Which is the overall best strategy for the human when the computer has strategy "random"? Why do you think this one gets the best results?

2. Propose a modified payoff matrix, different from the one given in the lectures. In *PD-Two-Person-Iterated.nlogo*, under the Code tab, implement your modified payoff matrix by changing the numbers in the `get-payoff` function. Repeat (1) above with this new payoff matrix, recording the approximate average scores of human and computer after about 300 iterations. Does your new payoff matrix make any difference from what you saw in question (1)? Why or why not?

3. Download *PD-N-Person-Iterated-New.nlogo*. Set the number of agents with each strategy to 20 (except set "n-unknown" to 0). Run the model for about 100000 time steps (set the speed bar to its maximum). Which strategy ends up having the highest payoff? Repeat this four more times. Do you always end up with the same strategy at the top of the "Average Payoff" plot? Why do you think you see these results?

4. Again, using *PD-N-Person-Iterated.nlogo*: Use the following settings:

   ```
   n-random: 0               n-tit-for-tat: 20
   n-cooperate: 20           n-unforgiving: 0
   n-defect: 20              n-unknown: 0
   ```

   In short, you're creating a tournament among cooperators, defectors, and tit-for-tat. Which strategy comes out on top this time? Experiment with varying the ratio of cooperators to defectors. What numbers of cooperators versus defectors allows tit-for-tat to come out on top?

Task 2 (25 points)

1. Open *PD-Two-Person-Iterated.nlogo*. Go to the code tab and find the procedure "`custom-strategy`". Change the code to implement your own strategy. How does your strategy fare against the other strategies in this model?

2. Open *PD-N-Person-Iterated.nlogo*. Go to the code tab and find the procedure "`unknown`". Implement your own strategy in this procedure. Again, how does your strategy fare against the other strategies in this model?

Task 3 (35 points)

Write and describe your own Prisoner's Dilemma strategy for a tournament between students ("*MyStrategy*" function in Python). Read the *Tournament* page in Moodle.