Introduction to Complexity

Unit 8 Homework

(optional)

You can choose whether you want to do the Beginner, Intermediate, or Advanced level. You are free to discuss this part of the homework with anyone, or to ask questions about it on the course forum. You can also use any resources you like, including our course NetLogo models. Note that homework in this course is not turned in or graded.

All of the exercises below are experimental and open-ended, so we will not be providing solutions for this homework set.

Beginner Level:

1. Download **PD-Two-Person-Iterated.nlogo** from the Course Materials page. (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 ``to 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** from the Course Materials page. Set the number of agents with each strategy to 20 (except set "n-unknown" to 0). Run the model for about 100,000 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-cooperate: 20

    n-defect: 20

n-tit-for-tat: 20

n-unforgiving: 0

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?

5. Download El-Farol.nlogo from the Course Materials page. Experiment with different parameter settings (memory-size, number-strategies, overcrowding-threshold). Which settings do you find result in the smallest percent of overcrowded days (after about 1000 iterations)? If you know how to use the NetLogo Behavior Space tool, this will help in experimenting with different parameter settings.

Intermediate Level:

1. Download PD-Two-Person-Iterated.nlogo from the Course Materials page. (This version fixes a bug that was in the original Models Library version.) Go to the code tab and find the procedure "to custom-strategy". Change the code to implement your own strategy. How does your strategy fare against the other strategies in this model?

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

3. In El-Farol.nlogo (on the Course Materials page), an agent's "best current strategy" is determined by testing each of the agent's strategies on each past week in the agent's memory. Put in a slider that allows the user to set how many past weeks are used to determine each agent's best current strategy. Experiment to see what the optimum number of weeks is, given values of other parameters (you can use the Behavior Space tool for this).

Advanced Level:

1. Read the paper "Evolutionary Games and Spatial Chaos" by Nowak and May

http://www.fas.harvard.edu/~ped/people/faculty/publications_nowak/Nature92.pdf

This paper describes agents playing the Prisoner's dilemma on a spatial array. Implement this model in NetLogo and see if your version has the same behavior as described in the paper. Now read "Evolutionary Games and Computer Simulations by Huberman and Glance ((http://www.pnas.org/content/90/16/7716.full.pdf). This describes what happens when Nowak and May's model uses asynchronous updates. Try this in your model as well. Finally, read Nowak et al.'s reply to Huberman and Glance, "Spatial Games and

the Maintenance of Cooperation" (http://www.pnas.org/content/91/11/4877.full.pdf), and try their parameter settings in your model as well.

2. A 2006 paper by Santos et al., "Graph topology plays a determinant role in the evolution of cooperation", links scale-free network topology with the conditions necessary to sustain cooperation in the Prisoner's Dilemma. Read this paper ((http://rspb.royalsocietypublishing.org/content/273/1582/51.full.pdf) and implement their model in NetLogo to see if you can replicate their results.

3. Extend El-Farol.nlogo to use a simple genetic algorithm to evolve new strategies for each agent. Do the new strategies do a better job of predicting the future?