CIS 603 -- Lab #1
Due: Thursday, 30 Oct 2014

Repeated Games

In this assignment, you will conduct and analyze round-robin and evolutionary computer tournaments in the following three repeated matrix games:

	c	d
c	3, 3	0, 5
d	5, 0	1, 1

Prisoner's dilemma

	c	d
c	3, 3	4, 6
d	6, 4	1, 1

Chicken

	c	d
c	4, 4	-5, 3
d	3, -5	2, 2

Stag hunt

You should hand in a report describing what happened in your computer tournaments. The report should describe your algorithm, the results, and an analysis of your results.

1. Design your own algorithm for repeated games, keeping in mind that (1) it will need to play the above three games, (2) you will want to win both round-robin and evolutionary tournaments with it, and (3) you will need to code it up. Your algorithm must differ in some way from the seven algorithms listed in the next step. In a short write-up, describe your algorithm. Explain why you selected it and why you think it might win. Note: the algorithm you create should not use information about the number of rounds in the repeated game.

2. Conduct and analyze a round-robin tournament for each of the three games listed above. That means you will need to write the code (in any programming language you want) to run the tournament, and you must code up the algorithms for the tournament. The tournament you conduct should adhere to the following guidelines:

   a. The tournament should include the following 8 algorithms:
      (1) tit-for-tat
      (2) tit-for-2-tats
      (3) random
      (4) always defect
      (5) always cooperate
      (6) maximin – plays its maximin strategy in each stage
      (7) win-stay, lose-shift – This algorithm begins by playing cooperate, and then repeats the action it played in the previous round if its last payoff was greater than or equal to its average payoff in the game (2.25 in the prisoner's dilemma, 3.5 in chicken, and 1.0 in the stag hunt).
      (8) The algorithm you created in step 1.

b. Each algorithm should play against itself and each of the other algorithms in a repeated game of 1000 rounds. To eliminate possible stochastic effects, take the average of 5 (or more) separate repeated games for each pairing.

c. Write up the results of your computer tournaments. In your report, discuss why certain algorithms did well in each game, and which ones did not do so well. Include in your analysis discussions of self play, kingmakers, etc. In other words, say why the winner won and why some of the other algorithms did not win. How did your algorithm do?

2. Use the results computed in part 1 to run an evolutionary tournament (using the replicator dynamic) of 1000 generations (or less if the population converges before then) in each of the three games. In the first generation, each algorithm should be equally represented. Report results of the evolutionary tournament, and analyze why things happened like they did. You should answer questions such as: Which algorithms were successful and which ones were not? Were there any attributes that defined the successful algorithms? Did the same algorithm win the round-robin tournament as the evolutionary tournament?

   You should implement the following pseudo-code to run the evolutionary tournament:

C Evolutionary Tournaments

Formally, let Ω be the set of algorithms in the population, and let $P^{\tau}(o)$ be the percentage of the population employing algorithm $o \in \Omega$ in generation τ. Also let, $S(o_i, o_j)$ be the average payoff achieved by algorithm $o_i \in \Omega$ when playing $o_j \in \Omega$ in a 1,000 -round repeated game. Then, an evolutionary tournament proceeds as follows:

1. For all $o \in \Omega$, set $P^1(o) = \frac{1}{|\Omega|}$, and set $\tau = 1$.
2. While $(\tau \leq 1000)$
3. $\quad$ For all $o \in \Omega$, $u^{\tau}(o) = \sum_{o' \in \Omega} P^{\tau}(o')S(o, o')$
4. $\quad$ For all $o \in \Omega$, $P^{\tau+1}(o) = \frac{P^{\tau}(o)u^{\tau}(o)}{\sum_{o' \in \Omega} P^{\tau}(o')u^{\tau}(o')}$
5. $\quad$ $\tau = \tau + 1$.