CS445 - Extra Credit Assignment 5

(improved error handling)

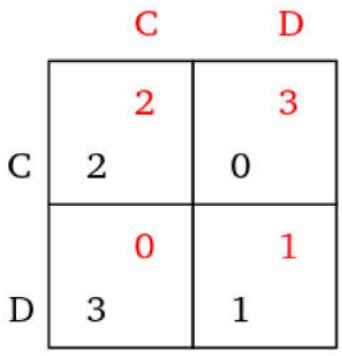
REWARD: 200 points

DUE: Wed Dec 14 at 5PM PDT

TEST DATA: None

Let's explore the idea of evolution without an explicit fitness function but rather a "fitness relation". I can only tell if one chromosome is better than another.

We will use the Prisoner's Dilemma from Game Theory. We will use the points version of the payoff matrix seen here:



In iterated Prisoner's Dilemma the game is played repeatedly an unknown number of times. The points are summed over all the games. A strategy is defined as a function that determines the next play (cooperate (C) or defect(D)) depending on the history of games. Memory is limited to a fixed number of most recent games and so the function takes the plays of both players in that history and predicts the next play. For example, if only looking one game into the past then there are the plays of each player (either C or D). So there are 4 possible plays in the previous game if you are talking about both players: CC, CD, CD, DD. There are 2^(2^2) plays for two games back in history and 2^(2^3)=64 possible 3 games. So ... a strategy for 1 game history is a table of 4 plays associated with CC, CD, DC, DD and a strategy for a 3 game history is a table to 64 plays. So, for example, if I look back in the last 3 games and see CCDCDC then my play should be f(CCDCDC), where f is the strategy function.

Write a program to evolve the best strategy as a function of the last 3 games. This should be a string of 64 (C's and D's) for the histories: CCCCCC through DDDDDD. Invent a comparison function and run an evolution algorithm of some kind. Turn in your software and a half page pdf report explaining what you did, how your algorithm worked including how you compared individuals, what your best strategy was and what you learned. Tell me whether you program found "Tit-for-Tat" strategy or something else.