Prisoner's Dilemma with Optional Cooperation and N Participants

Matt Lee
TJHSST Computer Systems

Abstract

This project is designed to simulate the classical Prisoner's Dilemma with a large number of participants and set options to cooperate with others or not. The purpose of this project is to allow the Prisoner's Dilemma to have variable parameters so that a variety of situations and settings could be tested. The result that is expected is a variety of simulations that will show how a specific situation can turn out when given options to cooperate, backstab, or 'join forces'.

Background

The Prisoner's Dilemma has been implemented a large number of times. There have even been competitions held to see who could make an algorithm that would maximize the payout for their specific participant. As stated by Robert Axelrod, the author of 'The Complexity of Cooperation' the best strategy for maximizing payoff is to use 'tit for tat'. Tit for tat is a strategy where the participant mimics the last move played by the opponent, which in the long run, enables the user to maximize his payout at the end of the 'game' of Prisoner's Dilemma. However, interestingly enough, when both participants initiate tit for tat, it doesn't become the optimal strategy. From here, a large number of variations have been made to the Prisoner's Dilemma, including implementing "N" participants instead of just two.

```
if(truthcount==falsecount)
                {
                    CooperateBetray(prison
ers);
                }
                else
if(truthcount>falsecount)
                {
                    CooperateCooperate(pri
soners);
                }

                else
if(falsecount>truthcount)
                {
                    BetrayBetray(prisoners
);
                }
This portion of code should be able to
  be implemented for future development
  with "N" prisoners.
```

Progress

As of now, my version of Prisoner's Dilemma runs similarly to the classical version. There are two participants and both are trying to maximize their outputs in a total of six rounds of play. The code had originally been utilized to create two 'prisoners' and use them for play. The two would simply make a random choice which would in turn, determine their payout. However, this random decision strategy is merely for testing purposes, other strategies including tit for tat will be utilized later.

Results

The results I expect are a variety of scenarios to be implemented by the Prisoner's Dilemma that will include "N" participants of the user's choice and option to enable morality.