A
Project Report
on
REPEATED PRISONER'S DILEMMA
by
Bhakti Khude and Vardhini Kundale

Abstract

Situations involving cooperation and defect can be modelled using the model of Prisoner's Dilemma. The equilibrium outcome for such one shot game is that both players defect. Interesting strategies arise when the game is played iteratively for finite number of rounds. The idea is that two players may choose to cooperate by the threat of punishment by the other player in future for cheating in the present period. This report adds another strategy for Repeated Prisoner's Dilemma problem. It is based on co-operation with like-minded opponent.

I) INTRODUCTION

Game theory is the study of interactions among intelligent decision makers (players). In a game, each player has a set of strategies from which he can choose his move or action. Due to interactions, the payoff or return of a player, not only depend on his action but also on all the players' actions. Thus, players choose strategies which will maximize their payoffs. The analysis of such interactions to find optimal or equilibrium outcomes lies at the heart of the subject. Game Theory has tremendous applications in political science, ethics, psychology, philosophy, economics, computer science and evolutionary biology [1].

One of the most well-known example of a game is that of a Prisoner's Dilemma. It can model various social situations such as Duopoly, Arms Race, Common Property, Tariff wars between countries, etc. It involves two players who both have two strategies – "cooperation" (C) and "defection" (D). The problem can be modelled as N players interacting via Prisoner's Dilemma (two at a time) repeatedly. The interesting and rich situations arise when the game is played iteratively. At every point of time, each player can decide his action depending on the history (set of previous outcomes). Thus, a strategy of a player takes the form of a decision rule which is a function of previous histories [1].

II) PAYOFF MATRIX

		Player 2	
		C	**D**
Player 1	**C**	**(20 , 20)**	**(0 , 30)**
	D	**(30 , 0)**	**(10 , 10)**

Table 1: A representative payoff matrix for Prisoner's Dilemma Game

Two player strategic form game can be represented with the help of a payoff matrix (See Table 1). If there are two players, 1 and 2, then rows represents strategies of player 1, while columns represents a strategy for player 2.

III) STRATEGY DESCRIPTION

In the first 3 rounds the strategy would be D, D, C. From 4th round onwards all the outcomes of the opponent and self will be examined for like-mindedness. Also, opponents individual C's and D's

will be examined. If like-mindedness is observed and the previous outcome of ours and his are same (either CC or DD) or if he has co-operated in previous round while we defended (outcome of ours and his is DC) and if his number of C's are greater than D's, then we will put C for the next round. If like-mindedness is not observed then all the individual outcomes of the opponent are checked to see if he has shown willingness to co-operate. If the number of C's are greater than D's for all his previous entries observed, it will be a C for the next round. For all other conditions it will be D.

IV) C-CODING

The strategy has been implemented by using C-Programming to compete with other strategies for perfomance. The C-code for the strategy as described in Section III is mentioned below.

```
char g4(char g4_q)    /* Function definition */
{
 static char g4_M[g4_n]={'Z'}, g4_O[g4_n];
 static long int g4_Iternum,g4_j=0;/*,k,a,b,C,D;*/
 long int k,a,b,C,D;

 g4_O[g4_j]=g4_q;
 if(g4_M[0]=='Z')
 {
  g4_Iternum =1;
  g4_M[0] = 'd';
  g4_M[1] = 'd';
  g4_M[2] = 'c';
  return g4_M[0];
 }

 if(g4_M[g4_Iternum-1]=='c'|| g4_M[g4_Iternum-1] =='d')
 {
  g4_Iternum = g4_Iternum + 1;
  g4_j = g4_Iternum -1;  /* Decrementing g4_Iternum to get correct array locations */
 }

 if(g4_j>=3)     /* Iteration 4 onwards */
 {
  a =0;b=0;C=0;D=0;
  for(k=0;k<g4_j;k++)
   {
    if(g4_M[k]==g4_O[k])
     { a = a+1;}         /* Like-mindedness */
    else
    { b = b+1;}          /* Differences */

    if(g4_O[k]=='c'){ C = C+1;}   /* C counter */
    else { D = D+1;}           /* D counter */
   }

  if((a>=b && g4_M[g4_j-1]==g4_O[g4_j-1] && C>D)||(a>=b && g4_O[g4_j-1]=='c' && C>D))
  { g4_M[g4_j]='c';}

  else if (b>a && C>D)
   { g4_M[g4_j]='c';}
```

```
    else { g4_M[g4_j]='d';}

     return g4_M[g4_j];
   }
     return g4_M[g4_j];   /* for returning g4_M[1] and g4_M[2] element */
}
```

V) PERFORMANCE WITH OTHERS

There were 11 teams competing with each other and also with themselves. The number of iterations was fixed to 10000 and scores were recorded for each team. The performance of our strategy was observed and eventually some modifications were made in order to maximize the total score. Two modifications have been tested to check if they are able to maximize the total score. The original G4 algorithm will be referred as Strategy-1, first modification as Strategy-2 and second modification as Strategy-3.

The score board when using Strategy-1 is dispalyed below:

Team	Without self	With self	Total score	Ranking
G1	1069500	100000	1169500	**3**
G3	1067590	199950	1267540	**1**
G4	1067430	100010	1167440	**4**
G5	1067340	100000	1167340	**5**
G7	1067290	200000	1267290	**2**
G8	1067240	100000	1167240	**6**
G11	1062940	101400	1164340	**7**
G10	1050520	100000	1150520	**8**
G12	1033480	100000	1133480	**9**
G9	1000000	100000	1100000	**10**
G2	801190	133320	934510	**11**

Table2: Score chart for Strategy-1

G4 algorithm performs satisfactorily when competing with other strategies. But it fails when it competes with itself. The tendency of G4 competing against itself should be of co-operating nature in order to maximize the score. But, because the first three iterations are fixed as D,D,C, the C-count governed by 'C' variable has value C=1 and the D-count governed by 'D' variable has value D=2. The other counters for like-mindedness governed by 'a' has value a=3 and for differences governed by 'b' has value b=0.

Though likemindedness is seen, but since C-count is less than D-count it puts D for all the further iterations.

VI) MODIFICATIONS TO STRATEGY-1

In order to improve performance with itself, two modifications have been tried and tested.

Modification 1- (Strategy-2)

The beginning sequence is changed from D,D,C to D,C,C in order to capture the co-operating nature when played with itself.
The score board when using Strategy-2 is dispalyed below:

Team	Without self	With self	Total score	Ranking
G1	1069520	100000	1169520	**4**
G3	1067590	199950	1267540	**1**
G4	1067420	199990	1267410	**2**
G5	1067340	100000	1167340	**5**
G7	1067290	200000	1267290	**3**
G8	1067240	100000	1167240	**6**
G11	1062940	101400	1164340	**7**
G10	1050520	100000	1150520	**8**
G12	1033480	100000	1133480	**9**
G9	1000000	100000	1100000	**10**
G2	801190	133320	934510	**11**

Table 3: Score chart for Strategy-2

G4 algorithm, while competing with others with Strategy-2, suffers 10 points loss than previous Strategy-1. But it does substantially well when it competes with its own copy. As the co-operating nature is evident from the first three trials and like-mindedness is seen, both the G4 algorithms co-operate for all the next iterations and the score is maximized.

Modification 2- (Strategy-3)

In order to maximize the score when playing with self to its full potential, another modification was to start with a C from the first iteration itself. The beginning sequence of defining the output for first three iterations was changed to only the first iteration.
The score board when using Strategy-3 is dispalyed below:

Team	Without self	With self	Total score	Ranking
G1	1069500	100000	1169500	**4**
G3	1067590	199950	1267540	**1**
G4	1067430	200000	1267430	**2**
G5	1067340	100000	1167340	**5**
G7	1067290	200000	1267290	**3**
G8	1067240	100000	1167240	**6**
G11	1062940	101400	1164340	**7**
G10	1050520	100000	1150520	**8**
G12	1033480	100000	1133480	**9**
G9	1000000	100000	1100000	**10**
G2	801190	133320	934510	**11**

Table 4: Score chart for Strategy-3

The 10 points loss suffered in Strategy-2 is recovered as well as score when competing with its own copy is maximized to its full potential.

VII) CONCLUSION

The basic rules for co-operation that were recognized by Axelrod in the first competition are still valid: kindness, provocability, forgiveness and simplicity. The information carries the key role in intelligent strategies. Individuals with more information will have advantage in most of situations so the strategies that learn about the opponents and adjust their own behaviour will certainly have an increasingly important role in the future.

REFERENCES

1. Sandeep Pathak, *"Repeated Prisoner's Dilemma"*, Retrieved from http://www.physics.iisc.ernet.in/~psandeep/Talks/RepGames.pdf , December 2007.

2. Marko Jurisic, Dragutin Kermek and Mladen Konecki, *"A review of iterated prisoner's dilemma strategies"*, MIPRO, 2012 Proceedings of the 35th International Convention, p. 1093 – 1097, May 2012.

3. Prisoner's Dilemma, Retrieved from http://en.wikipedia.org/wiki/Prisoner%27s_dilemma, October 2014.