Player Agents Design

Game Theory: Assignment II

Kamil Akhmetov
B17-DS-01
Innopolis University
Innopolis, Republic of Tatarstan, Russia
k.ahmetov@innopolis.ru

Abstract—

I. Game description

The environment of the game is a field of 3 sections: A, B, C. They are initialized with value of 1. Two players simultaneously chose one of these fields to stand on. Next, one of the several paths is followed:

- If a field is empty (not chosen by any player), it increases its value by 1
- If a field is occupied by one of the players, it decreases its value X by 1 unless it is 0 and that player receives a value of $f(X) - f(0)$, where $f(X) = \frac{10*e^X}{10+e^X}$
- If a field is occupied by both of the players, it decreases its value by 1 and both players receive no payoffs.

Each game consists of K such sequential rounds.

II. Goals of assignment

In this work, we aim to develop and try to adapt and implement several strategies for an agent playing this game. In order to identify the best strategy we are going to use Tournament method.

III. Player strategies

A. Cyclic iterative

Start with strategies based on cyclic alternation of the fields to choose, including 2 options:

- `CF` - Cyclic Forward [0 1 2 0 1 2 ...]
- `CB` - Cyclic Backward [0 2 1 0 2 1 ...]

These strategies do not use any in-coming information, and basically have no memory and move regardless of what the opponent has done previously.

B. Random

Strategies in this set are based on random choice. This includes:

- `RAND` - Random choice
- `RANDNOTMIN` - `RAND` except the minimal value
- `RANDNOT0` - `RAND` avoiding 0 valued fields if possible
- `RANDCF` - `CF` with random first choice
- `RANDCB` - `CB` with random first choice

Random elements might be also used in other models.

C. What opponent just did?

Now, we start considering the moves of the opponent, at least use the last one to decide. Strategies:

- `COPY` - copy the last move of the opponent with random first choice
- `COPYNOT0` - `COPY` avoiding 0 valued fields if possible
- `ANTICOPYNOTMIN` - choose the next not minimal field after the last choice of the opponent

D. Cooperate or Defect?

Now, in order to move further, first let us define **cooperation** and **defection** for the game.

Definition 1: **Cooperation** - application of mutually beneficial actions to increase the scores of both players.

We suggest to consider ***alternating capture and concession of the maximum value fields*** to be cooperation for this game.

Definition 2: **Defection** - application of actions that do not imply any cooperation and are aimed solely at personal gain.

We suggest to consider ***attempts of a player to capture the maximum value field out of the turn*** to be defection for this game.

Let us consider the following strategies:

- `ALLD` - simply defect every time, just always attempt to take the maximum valued field
- `ALLMID` - always choose the middle valued field, between the maximum and the minimum cells. This can be viewed partly as a degenerate case of cooperation
- `ALLMIN` - always choose the minimal valued field. Also can be viewed as a degenerate case of cooperation.
- `ALLMIDMIN` - alternate between the middle valued and the minimum valued fields. This model introduces a concept of *state*.
- `ALLMIDMAX` - alternate between the middle (the largest except maximum) and the maximal fiend. Cooperative strategy.

E. More Advanced strategies

In this section we have tried to adapt for our game (given the above definitions of **cooperation** and **defection**) strategies described in the paper originally for Iterative Prisoner's Dilemma.

1) TFT: Tit for tat. Cooperate by default, but reflect defections with defections. This may be viewed as ALLMIDMAX which attempts to take maximum while it is not taken.

For better explanation see the Finite State Transducer like representation.

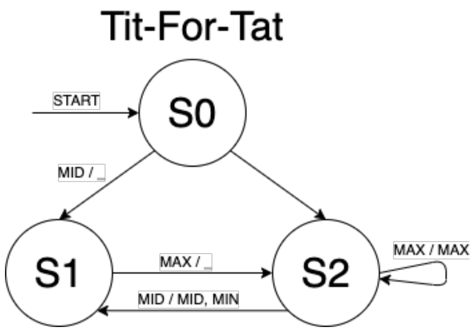

Fig. 1. TFT FST

2) Fortress: Agent with Fortress (N) [**?**] architecture tries to establish a N-times-defection handshake before starting a cooperation. So it maintains a counter for these defections during a handshake. We have tried different N parameter values:

- FORT5
- FORT9
- FORT45

3) Offence Keeper: I have tried to develop and implement a new strategy which I have called "Offence Keeper". The main point of it is to maintain the counter for the number of times the agent does not receive what it wants to receive.

- KEEPER - maintains offence counter
- PATKEEPER - KEEPER having initially negative offence, in other words it is "patient" at the start.

IV. Testing & Tournament

In order to find the best strategy we use the Tournament procedure. That is for N agents we have $\frac{N(N-1)}{2}$ games consisting of K rounds each. We have tested the N=42 agents with 21 different strategies (2 players each) with K=1000 rounds in each of the games.

From the Tournament results we can see that RANDNOTMIN and ANTICOPYNOTMIN strategies dominate the over the others. For that reason we chose ANTICOPYNOTMIN as the main strategy for the competition. An Interesting fact is that Random based strategy performs better than complicated ones.

Place	**Name**	**Score**
0	RANDNOTMIN	91192
1	ANTICOPYNOTMIN	91076
2	ANTICOPYNOTMIN	90955
3	RANDNOTMIN	89714
4	FORT9	83227
5	FORT9	81723
6	FORT5	81098
7	FORT5	80794
8	RANDNOT0	79380
9	RANDNOT0	79120
10	FORT45	70555
11	ALLMIDMAX	70082
12	COPYNOT0	69860
13	COPYNOT0	68841
14	ALLMIDMAX	67804
15	FORT45	67354
16	ALLD	64998
17	ALLD	64779
18	RAND	62656
19	RAND	62176
20	TFT	61662
21	KEEPER	61583
22	KEEPER	61508
23	TFT	61508
24	PATKEEPER	56246
25	PATKEEPER	53267
26	RANDCF	49096
27	RANDCB	48210
28	CB	46422
29	RANDCB	43931
30	RANDCF	43758
...	...	...

TABLE I: Tournament results

V. Code

All the code is written in self-explanatory manner, but logic of agents might be unclear, that is why we comment on them in the current report.

VI. Results & Conclusions

To conclude, we have learned how to design and implement agents for a game, designed environment, game and tournament simulations. We have learned different strategies, and also tried to design new ones. By putting experiments, we learned to determine better strategies in a set of several ones.

All in all, it was an interesting journey, thank you!

References

[1] Brown, Joseph and Ashlock, Daniel, "Domination in Iterated Prisoner's Dilemma", Canadian Conference on Electrical and Computer Engineering, pp. 1125-1128, May 2011.