The Second Practice Test on Game Theory Report

Author: Mahmoud Mousatat
Organization: *Innopolis University*
Group: DS-01
email: m.mousatat @innopolis.university

***Abstract*—This Report is based on the assignment2 Game theory. the perpose of this assignment is to creat best agent which will let you minimizethe payoff of the game based on the number of balls remained on your field.**

- Introduction

A snowball fight is a vigorous game that is most frequently played in the winter when there has been enough snowfall. Players typically divide into teams to attack other teams or their fields. To better understand the nature of the game, we will model the snowball game environment in this simulation.
Three territorial areas—designated A, B, and C fields—make up the setting where participants engage in the snowball game. There is one participant on each side in fields A and B. Since there are no players on field C, it should be regarded as a hot field where snowballs instantly melt and vanish. The participants must shoot snowballs from the field they are on because they are unable to move. You can suppose that N=100 represents the starting number of snowballs in both the A and B fields. There is just one Snowball Generating Machine in each of fields A and B. (SGM). These devices cause the number of snowballs N in both fields to rise by 1 every hypothetical minute.
Assume that both SC initially had a history of shooting within the last minute. The number of snowballs in your field must be kept to a minimum after 60 minutes (also known as rounds) in order to maximize your reward.
Eventhough, this problem seems simple but it hide inside its exception a lot of bugs and wrong results till you get the best result.

- Technique/Method

I created five Agents classes:
1.greedyMaxAgent: This Agent act like greedy agent it does not care about what the opponets acts, so This Agent Always Shot to both Sides after waiting number of minutes.
I tried to test multible variace of this agent with changing the number of minutes that the agent should wait before shoting, evntually the best number of minutes is Three.
2.wise_Random_Agent: This Agent is more advanced random agent, intsead of randomly choose how many snow ball The agent should thow, it choose randomly if the agent want to throw fixed number of balls or never throw any balls in this minute. then if the random.
IF the result of the randomization is Shot. the Agent will shot the maximum allowed number of snowballs to Both side.

3.completlyRandomAgent: This Agent is Shot completly random number of balls with following the restrictions, and the number of balls shots to opponents will be choosen randomly deferent than the number of balls shots to hot field.
4.agent_Shot_to_Only_Hot_Field: This Agent only Shot To Hot Field after number of minutes

5.agent_Shot_to_Only_opponent_field: This Agent only Shot To oponnent Field after number of minutes

- The functions in the testing code

1.
public static int[] tournament(Player a,Player b)
This Function return array of two elements which represent the final number of snowballs remains in Field A and B.
It take two agents a and b and test them on both Field A and Field B.
Inside the function there is a loop from 0 till 60 which represent the time increasing till the end of the round. inside the Loop every turn we detect the change in the snowball number inside both fields and and minutes that passed after last shot for both Agents.
2.
public static int[] multibletests(Player a, Player b)
This Function is for testing the agents on multible game rounds, because the random startegies usually give deferent results. So it will be better to test and get average payoff.
It return array of two elements which represent the avarage final number of snowballs remains in Field A and B after test this two agents 100000 times.

- Result

The output of the Source Code which used for my testing and testing cases is as following:
1.compare between the greedy Agent which is object from class greedyMaxAgent and wiseRandom Agent
2.compare between the greedy Agent which is object from class greedyMaxAgent and fullyRandom Agent which is object form class completlyRandomAgent.
3. compare between the greedy Agent which is object from class greedyMaxAgent and toOnlyHot which is object form class agent_Shot_to_Only_Hot_Field.
4.compare between the greedy Agent which is object from class greedyMaxAgent and toOnlyOpponent which is object form class agent_Shot_to_Only_opponent_field.

```
greedy Agent vs wise Random
The avarage payoff Of first Agent 7 : The avarage payoff Of second Agent 94

greedy Agent vs fully Random Agent
The avarage payoff Of first Agent 6 : The avarage payoff Of second Agent 166

greedyAgent vs agent Shot to Only Hot Field
The avarage payoff Of first Agent 3 : The avarage payoff Of second Agent 105

greedyAgent vs agent Shot to Only Opponent Field
The avarage payoff Of first Agent 11 : The avarage payoff Of second Agent 160
```

as we see from the output the Greedy Agent always win against other againt becuase of that I choosed it as my best agent.

As the output show the Greedy agent is the best agent which is object of the class greedyMaxAgent.