Game Theory
Assignmet II

Maxim Evgrafov,
BS-18-DS-01,
m.evgrafov@innopolis.university

Abstract

This paper describes the creation process of player agent for Moose Game. Method explanation, design, and testing process are provided in separate sections.

Index Terms

Game Theory, Moose Game

I. Introduction

MOOSE Game is a game, where two players compete against each other. There are three fields, where agents able to get their payoffs. Each field has its X value which is used to calculate the payoff for agents. The payoff for an agent on the field k with X value X_k is:

$$payoff_k = \frac{10 * e^{x_k}}{1 + e^{x_k}} - 5$$

If there are two agents on the field, then they both get payoff equals zero. X values are changing after each round. If there are no agents on the field, then the X value of this field increases by 1, otherwise, it decreases by 1. In the case of X, value equals zero X value remains its value.

In Moose Game Tournament all agents that are enrolled in the competition play against each other. A total score is the sum of all scores that were gained in games. The problem is to choose such a strategy that will get as many points as it can, even if you do not know the strategies that it gonna play with.

There will be two main sections in the paper: the strategy section and the test section. The strategy section contains information about algorithm explanation and Java implementation. The test section describes some common strategies and testing tournament results, that show the effectiveness of my strategy.

March, 2021

II. Strategy explanation

The main idea of my strategy is to choose the best field with the probability equal to the frequency of the opponent's best field choice. Also, when it is not important for us which field has the maximum payoff, i.e. they approximately equal, we decrease our probability of choosing the best. In this case, we should choose the field randomly, but taking into consideration their coefficients.

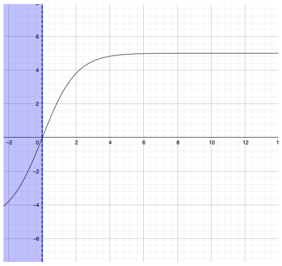

Fig. 1. Payoff Graph

First of all, let's assume that the Best Move strategy is the simplest strategy that can be applied in this game. That is why during the creation of my agent I assumed that in the tournament will be agents with this or a similar strategy. But the problem with the Best Move strategy is that when they are playing with each other average score will not be as high as for other strategies.

Secondly, cooperation in this game means that the agent will not always choose the field with the highest payoff, giving the opponent ability to get it. So, my agent also should be able to cooperate.

Thirdly, after the X value equal to 5 difference between payoffs is not that important, because the agent will get approximately the same number of points (see Fig. 1).

Taking into consideration all points that are described above, I created the strategy that can play with Best Move strategies with a higher average score and cooperate with strategies that want to cooperate with me.

III. Strategy implementation

For text simplification here and after MyAgent will mean a strategy that is implemented and chosen by me as my tournament player. The agent player which I am playing against will be called Opponent.

To get information about the Opponent's best moves rate MyAgent saves all Opponent's moves into the list. Also, MyAgent always has information about previous X values to get knowledge about the status of the previous Opponent's move (best move or not).

At the start of the game, MyAgent randomly chooses between all the fields, because it doesn't affect first-round profit. In the next round, it calculates the best moves Opponent's rate, saves information about the fields (X values). Then it checks if all the fields have X values bigger than 5. If so, it divides the best move rate by 2 for getting bigger chances for cooperating. After that it chooses the best move with probability equals to calculated rate. On the other hand, MyAgent chooses the field randomly depended on their X values. Chance of moving to the field k (k, i, j are the fields) with X value x_k is p_k:

$$p_k = \frac{x_k}{x_k + x_i + x_j}$$

Full implementation code can be accessed by this link [1]. To get access, please, write to the e-mail that is written at the beginning of the paper.

IV. Test opponents' strategies

Testing requires the implementation of different types of opponents, which can be seen in the tournament. Some of these strategies are described below.

A. *Best Move Opponent*

This type of player always chooses the best field, i.e. field with the highest payoff. It is a type of greedy algorithm. The problem is that in a tournament with several greedy algorithms, their average score will be less than for other algorithms. The scoreboard for such a tournament is provided in Appendix A. Average payoff for agents here is 166. The game contains 100 rounds, so the average payoff per round is 1,6.

B. *Second Best Opponent*

This type of player always chooses the second-best field. It is the type of "silly" cooperative algorithm, which goal is to gain as much as it can when it's playing with greedy algorithms. In this case, an opponent would increase its score, but the average score per game will be higher because they choose a different field.

C. *Holder Opponent*

This is another "silly" type of cooperator which goal is to get as much as it can from one field. When the coefficient for the field, where it is located equal to zero it chooses the best field and stays there until it becomes zero again. While holder stays on one field other fields can increase their profit, so it is more stable to be there for a long time and then move to another cell, where income will be high for a long time.

D. *Random Opponent*

This player randomly chooses between the fields. I don't think that a real person will choose this type for a tournament, but for testing reasons, it is interesting to add such a player.

E. *Tit for Tat Opponent*

At the beginning of the game, this agent plays randomly. After that, it repeats the move that his opponent did in the previous round. Intuitively, the score of this agent will not be very high, because it always will get less than his opponent or zero.

F. *Weighted Random Opponent*

This type of agent can be named improved random algorithm because it chooses the field randomly depending on the field's feature (X value, payoff, squared payoff). For example, if fields have the same X value, the probability of choosing each field will be the same. If fields A, B, and C have X values 1, 1, 2 respectively, the probability of choosing C will be 0.5, A - 0.25, B - 0.25.

Also, the dependent value is an important parameter for the agent, because it affects overall results. For example, tests showed that the X value is a more successive parameter than the field's payoff.

V. TESTING

Here I want to show the results of the test tournament. There were 16 players: each test opponent strategy doubled and two MyAgent strategies. They are playing with each other for 100 rounds between each pair. Sorted results can be observed in Appendix 2.

VI. RESULTS

The strategy that I implemented become the first and third strategy in this tournament, so we can say that it statistically gain a higher score than other strategies. Also, strategy WeightedX, which uses weighted random depended on X value squared, becomes second. Because another player with this strategy gained not as much point as the first one we can say that the results for this strategy are not consistent and highly depended on luck.

VII. CONCLUSION

In this paper, we take a look at my implementation of the strategy for the Moose Game Tournament. I showed that results for my strategy are stable. The score for my strategy is usually higher than for other test strategies.

APPENDIX A
TOURNAMENT RESULT WITH BESTMOVE ONLY

	Best Move 1	Best Move 2	Best Move 3	Best Move 4	Best Move 5	Best Move 6
Best Move 1	0,0	180,3	194,2	148,9	189,7	178,9
Best Move 2	180,3	0,0	153,5	162,2	169,3	179,7
Best Move 3	194,2	153,5	0,0	159,3	141,0	141,0
Best Move 4	148,9	162,2	159,3	0,0	156,4	174,1
Best Move 5	189,7	169,3	141,0	156,4	0,0	177,1
Best Move 6	178,9	179,7	141,0	174,1	177,1	0,0

APPENDIX B
TOURNAMENT RESULTS WITH EACH TEST STRATEGY TWICE

Strategy	SUM	Average
MyAgent1	3183,2	212,2133333
WeightedX1	3182,9	212,1933333
MyAgent2	3149,3	209,9533333
BestMove2	3139,5	209,3
BestMove1	3082,6	205,5066667
WeightedX2	3038,7	202,58
WeightedProfit1	2926,5	195,1
WeightedProfit2	2906	193,7333333
Holder2	2279,7	151,98
Holder1	2233,9	148,9266667
Random2	2095,2	139,68
Random1	1901,7	126,78
TFT1	516,2	34,41333333
TFT2	500,2	33,34666667
SecondBest1	383,5	25,56666667
SecondBest1	330,2	22,01333333

REFERENCES

[1] https://github.com/MaEvGoR/GameTheoryAss2 Innopolis University, Innopolis, Russia