

Landlord-off Game AI

1 Xinru Xu

xuxr@shanghaitech.edu.cn

2 Miao Li

limaio@shanghaitech.edu.cn

3 Chuyi Zhao

zhaochyl@shanghaitech.edu.cn

4 Qingting Jiang

jiangqt@shanghaitech.edu.cn

Abstract—Fighting the Landlord is a famous Chinese poker game that is played under a stochastic, partially-observable environment. The aim of this paper is to implement an AI to play the Landlord-off game with a good performance. Two algorithms are discussed and implemented in this paper: A* algorithm and Minimax with Alpha-Beta pruning. The experiment simulates the game under different configurations. The results indicate that agents that use A* algorithm can have a better performance than normal agent.

Index Terms—Landlord-off, AI game

I. INTRODUCTION

This Paper is concerned with creating a computer to play Fighting the Landlord, a 3-player gambling poker game. Landlord-off has several features that are different than many other games like Pacman and chess. This game is played under a stochastic environment where they can only observe the discarded cards and the cards on their own hand. We create 3 different AI agents with different algorithms to play as the landlord. Then we test their performance by making them fight with each other as well as human players.

II. METHOD

A. Discard card

We discard cards to each player randomly, the landlord will get additional three cards.

B. Choose card

The most important step is to decide which cards to play at your turn. We need to first find all possible choices and then choose the best one.

- Normal agent.

For the normal agent, it will list all possible choices every time the card on its own hand changes. When it has the initiative, it will play cards legally and randomly. When it play secondly, it will just play the card with the least card value.

- Search agent.

For the search agent, it will first group its card legally, we denote a combination of groups as one possible choice. For example, it has cards 3,3,3,4,5,5, which can create a choice 333,4,55. The times we need to play to finish this choice is 3. This thought is realized by a A* algorithm with recursion. We aim at finding the optimal legal choice: the combination that finish playing whole cards using least times. Also, we cut the choice that needs more than 9 times since it seem unlikely to happen in real time. We assign a score to each choice at the same time.

The score is decided both by the original card value and the times it need. The higher the card value is, the higher the score it will get. The score of a choice will decrease with the increase of times. Whether the search agent has the initiative or not, it will choose the legal choice which has the highest score.

- Minimax agent.

For the minimax agent, we assume farmers will do the optimal choices. Things are different since our minimax agent is a cheat agent, it can know other players' cards. It can make precisely guess about other players' decision and then make its own move, just like the minimax search.

C. Results

We construct a test by first making normal agents play against each other and then making normal agents play against search agent. We repeat the test for 100 times and collect some data. We don't compare them with the minimax agent since it is unfair.

TABLE I
NORMAL AGENT VS NORMAL AGENT

Table Result	Character		
	Win	Lose	tie
Landlord (normal agent)	51	49	0
Farmer1 (normal agent)	49	51	0
Farmer2 (normal agent)	49	51	0

TABLE II
SEARCH AGENT VS NORMAL AGENT

Table Result	Character		
	Win	Lose	tie
Landlord (search agent)	69	31	0
Farmer1 (normal agent)	31	69	0
Farmer2 (normal agent)	31	69	0

D. Conclusion

In conclusion, the search agent is better than normal agent. The rate of winning rate for the search agent is improved by 18% , indicating our algorithm is effective.

E. Future work

- Speed improvement.

In search agent, the recursion step is time-consuming so we have to cut the choices that need more than 3 steps

to finish. The performance is influenced since we can't guarantee the optimality.

- Farmer character.

Our agents are played as landlord, we are interested in changing them to play as farmer.

- Network learning.

The score used in search agent is determined by ourselves with reference to the advice of some landlord-off master players, we are going to use a neural network to learn a more proper score.

REFERENCES

As the whole project is totally developed by ourselves, we decide to push our codes to github as a open source project.