

AI Project1: Development of a superb AI Program for Reversed Reversi

叶璨铭, 12011404@mail.sustech.edu.cn

1 Introduction

- 1.1 Problem background
- 1.2 Literature review
 - 1.2.1 Normal Reversi program
 - 1.2.2 Reversed Reversi program
 - 1.2.3 General board game program

2 Preliminary

- 2.1 Problem formulation and notations
 - 2.1.1 Enviroment E
- 2.2 Basic theorems and corollaries
 - 2.2.1 Environment theorems
 - 2.2.2 Performance theorems

3 Methodology

- 3.1 General work flow
- 3.2 Assumptions
- 3.3 Model design
- 3.4 Model analysis

4 Experiments

- 4.1 Experiment 1: 探究当算法为贪心算法时不同评估函数对Agent理性程度的影响
 - 4.1.1 Experiment principle and experiment hypothesis
 - 4.1.2 Experiment setup and experiment steps
 - 4.1.2.1 Sub-Experiment 1.1 Baseline verification
 - 4.1.2.2 Sub-Experiment 1.2 AHP combination for better
 - 4.1.2.3 Sub-Experiment 1.3 Genetic programming for local search
 - 4.1.3 Experiment results and experiment analysis

5 Conclusion and discussion

1 Introduction

In this project, we try to develop a **high-level *Reversed Reversi* program** that are compatible to beat our classmates' programs. Our purpose is **not seeking to prevail over others, but to master the essence of AI system development by learning from and communicating with each other.**

As Kronrod says, computer board game is the "fruit fly in Artificial Intelligence".¹ This interesting computer game project will definitely provide me with **a better understanding of the knowledge I learnt in the AI course** via my **hand-by-hand practice and experiments** in the procedure of accomplishing this project.

1.1 Problem background

Reversi, also called Othello, is a deterministic, turn-taking, two-player, zero-sum game of perfect information.¹ Reversi is not popular in China before the development of the Internet.² Although it may not often appear as a board game, it is indeed popular in the research of computer game because of its relatively small search space.¹ Computers have always excelled in Reversi because average human players cannot envision the drastic board change caused by move³, and because human players dislike to risk taking a seemingly bad but actually best move.⁴

Reversed Reversi, also called the anti mode of Othello, shares the same dynamics of the chessboard environment as Reversi, while the objective is the opposite.⁴ In the game, each player places a piece of his color on the board, flipping any opponent's pieces that are bracketed along a line. The object of the **Reversi** is to have the **most discs** turned to display your color when the last playable empty square is filled, while **Reversed Reversi** expects the winner to have the **least discs**.⁵ A formal definition of the game rules of Reversed Reversi will be presented in Part 2.

1.2 Literature review

Before we start to develop our own algorithms for Reversed Reversi, it is necessary for us to do a literature review on how previous researchers develop programs that play Reversi, Reversed Reversi and other board games.

1.2.1 Normal Reversi program

The first world class Reversi program is *IAGO* developed by Rosenbloom in 1982. This program effectively quantified Reversi maxims into efficient algorithms with adversarial search techniques.³ Later in 1990, 李开复 developed another program called *BILL*, which far surpassed the *IAGO* by introducing dynamic evaluation function based on Bayesian learning.^{3 6} Although *IAGO* and *BILL* are best computer programs that play Reversi at their times, the top human players are not beaten until the occurrence of the program *Logistello* developed by Buro in 1997.¹ The main new idea of *Logistello* is that it automatically learns from previous plays.^{2 4} **After that, it is generally acknowledged that humans are no match for computers at Reversi.**¹

In 1997-2006, Andersson developed a practical Reversi program called *WZebra*. It has big opening books, state-of-the-art search algorithms, pattern-based evaluation scheme that can stably run on Windows 2000 to even today's Windows 11 platforms.⁷ While it gains a better performance and stability by applying several techniques of C Programming Language, the basic ideas of *WZebra*, however, are no more than *BILL*'s or *Logistello*'s.

1.2.2 Reversed Reversi program

While it is often the case that reversed board game is easy and boring, such as reversed Chinese Chess, Go and Chess, Reversed Reversi is worth playing and it is an art to play it well.⁴ According to MacGuire, much of the strategic thinking behind the classic game can also be applied to the reverse game, though sometimes in reverse.⁸

Tothello, a program developed by Pittner, is believed to be the best program in the world playing Reversed Reversi until 2006. ⁴

1.2.3 General board game program

2 Preliminary

In the last part, we have known the background of the problem and found some useful references. Next, we need to **formulate** the problem in formal language to **disambiguate the potential confusion**. ⁵ With the formal logic system, we can then **derive some basic theorems and corollaries** that any Reversed Reversi game must logically follow. With these knowledge in our minds, it is clear how to design our models and experiments in the next sections.

2.1 Problem formulation and notations

Informally, the Reversed Reversi problem is to build a program playing Reversed Reversi with some kinds of high *Intelligence*.

Formally, this problem can be formulated as a **task environment**, which is specified by a tuple (P, E, A, S) , where P is the performance measure, E is the environment, A is the actuators, and S is the sensors.

Besides problem, we also need to formulate the program. The program for this problem can be formulated as an **Agent**, which is specified by a function G , which maps the agents' percept histories to its action.

Now we formulate P , E , A , S and G respectively.

2.1.1 Environment E

Environment E is defined by the following notations:

Notation	Meaning
St	The chessboard state at round t .

2.2 Basic theorems and corollaries

2.2.1 Environment theorems

Theorem 1.

2.2.2 Performance theorems

3 Methodology

3.1 General work flow

After having a literature review and doing a formulation, I find these problems to be vital in this project:

- Online
 - How to design an efficient and effective algorithm to **search adversarially** on the game tree? How to write it in Python with no bugs?
 - How to design a good **evaluation** function?
- Offline

- How to **measure** the intelligence of a Reversed Reversi program?
- How to utilize **local search** algorithms to find the best weight in evaluation function?
- How to generate opening books and the weights for pattern-based evaluation scheme by inverse **adversarial searching**?

Therefore, our general workflow

3.2 Assumptions

3.3 Model design

3.4 Model analysis

4 Experiments

Experiment is very important in Computer Science Research, at least as important as it is in natural science in my opinion.

The following experiments were performed under the following environment conditions:

- Systeminfo: AMD Ryzen 7 4800H with Radeon Graphics, Microsoft Windows 11 专业版, RAM 16G
- Python: Python 3.10.5, Pycharm 2022.1.3 (PE), numpy 1.22.4, numba 0.55.2

4.1 Experiment 1: 探究当算法为贪心算法时不同评估函数对Agent理性程度的影响

4.1.1 Experiment principle and experiment hypothesis

4.1.2 Experiment setup and experiment steps

As we said in 4.1.1, one of the best ways to evaluate an RR agents` rationality is to evaluate their ranking in the round robin. We need to

4.1.2.1 Sub-Experiment 1.1 Baseline verification

4.1.2.2 Sub-Experiment 1.2 AHP combination for better

4.1.2.3 Sub-Experiment 1.3 Genetic programming for local search

4.1.3 Experiment results and experiment analysis

5 Conclusion and discussion

References

1. Stuartj. Russell, PeterNorvig, 诺维格, 罗素, 祝恩和殷建平, 《人工智能:一种现代的方法》, 清华大学出版社, 2013, doi: [9787302331094](https://doi.org/10.13700/j.cnki.issn.1009-3095.2013.01.001). [↩](#) [↩](#) [↩](#) [↩](#) [↩](#)
2. “黑白棋,” 百度百科. <https://baike.baidu.com/item/%E9%BB%91%E7%99%BD%E6%A3%8B/80689> (accessed Jul. 16, 2022). [↩](#) [↩](#)

3. K.-F. Lee and S. Mahajan, “The development of a world class Othello program,” *Artificial Intelligence*, vol. 43, no. 1, pp. 21 – 36, Apr. 1990, doi: [10.1016/0004-3702\(90\)90068-B](https://doi.org/10.1016/0004-3702(90)90068-B). ↵ ↵ ↵
4. Pittner. “Reversed othello games.” http://www.tothello.com/html/reversed_othello_games.html (accessed Jul. 16, 2022). ↵
↵ ↵ ↵ ↵
5. Teachers and assistants in SUSTech. Project-Reversed-Reversi-EN.pdf ↵ ↵
6. 彭之军, 《计算机博弈算法在黑白棋中的应用》, 现代信息科技, 卷 5, 期 17, 页 73-77+81, 2021, doi: [10.19850/j.cnki.2096-4706.2021.17.018](https://doi.org/10.19850/j.cnki.2096-4706.2021.17.018). ↵
7. Andersson. “Gunnar’ s Othello page.” <http://www.radagast.se/othello/> (accessed Jul. 17, 2022). ↵
8. MacGuire. “Strategy Guide for Reversi & Reversed Reversi.” <https://www.samssoft.org.uk/reversi/strategy.htm> (accessed Jul. 16, 2022). ↵