

人工智能基础实验报告

刘砺志

(2014 级计算机 1 班 22920142203873)

本文是人工智能基础——智能五子棋游戏的实验报告。

1 实验概述

本次实验要求使用 $\alpha - \beta$ 剪枝技术，开发一个智能五子棋程序。

2 实验原理

2.1 $\alpha - \beta$ 剪枝 (Alpha-beta pruning)

下面的伪代码描述了使用 $\alpha - \beta$ 剪枝技术的极小极大算法 (minimax) 的流程：

ALPHA-BETA(*node*, *depth*, α , β , *maximizingPlayer*)

```
1  if depth = 0 or node is a terminal node
2      return the heuristic value of node
3  if maximizingPlayer
4      v :=  $-\infty$ 
5      for each child of node
6          v := max(v, Alpha-Beta(child, depth-1,  $\alpha$ ,  $\beta$ , FALSE))
7           $\alpha$  := max( $\alpha$ , v)
8          if  $\beta \leq \alpha$ 
9              break //  $\beta$  cut-off
10     return v
11 else
12     v :=  $+\infty$ 
13     for each child of node
```

```

14         v := min(v, Alpha-Beta(child, depth-1,  $\alpha$ ,  $\beta$ , TRUE))
15          $\beta$  := min( $\beta$ , v)
16         if  $\beta \leq \alpha$ 
17             break //  $\alpha$  cut-off
18     return v

19 // Initial call
20 Alpha-Beta(origin, depth,  $-\infty$ ,  $+\infty$ , TRUE)

```

2.2 五子棋模型

五子棋最常见的基本棋型大体有以下几种：连五，活四，冲四，活三，眠三，活二，眠二等，下面一一介绍。我们假设当前黑子执棋，0 表示空，1 表示黑子，2 表示白子。并且我们考虑对称性而不重复讨论一些情形。

2.2.1 连五

即五颗同色棋子连在一起，其表达式形如

11111

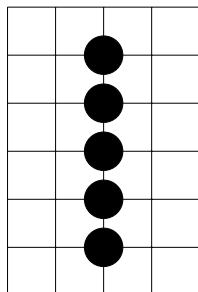


图 1 连五

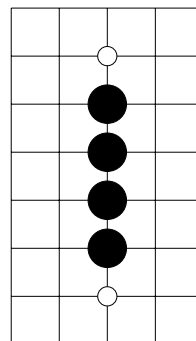


图 2 活四

2.2.2 活四

活四指有两个连五点（即有两个点可以形成五），图中白点即为连五点。其表达式形如

011110

2.2.3 冲四

即有一个连五点。其表达式形如

011112

10111

11011

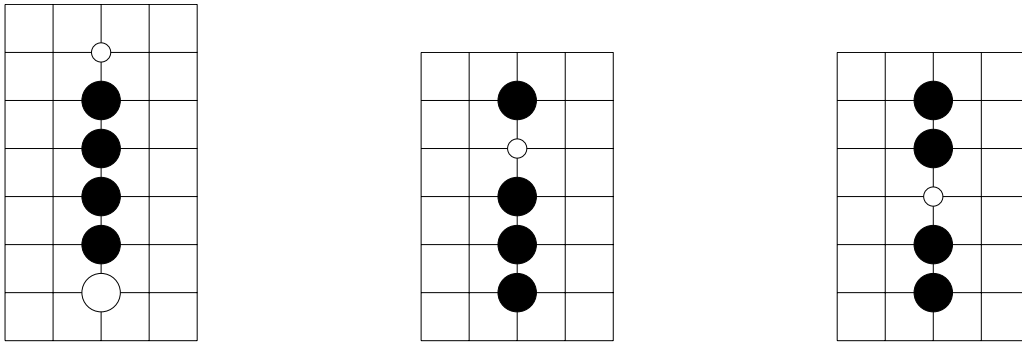
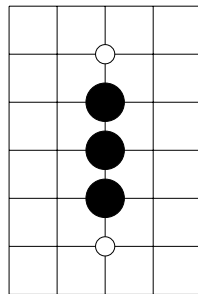


图 3 冲四

2.2.4 活三

指可以形成活四的三。其表达式形如

01110



1011

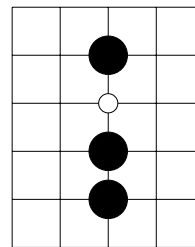


图 4 活三

2.2.5 眠三

表示只能够形成冲四的三。其表达式形如

001112

010112

011012

10011

10101

2011102

2.2.6 活二

即能够形成活三的二。其表达式形如

001100

01010

1001

2.2.7 眠二

能够形成眠三的二。其表达式形如

000112

001012

010012

10001

2010102

2011002

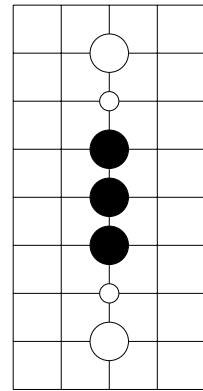
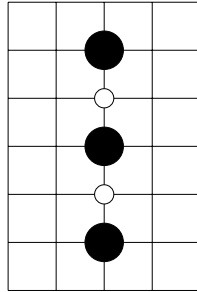
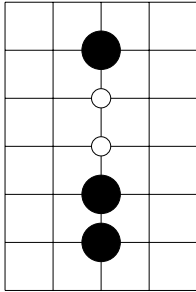
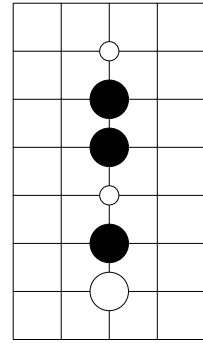
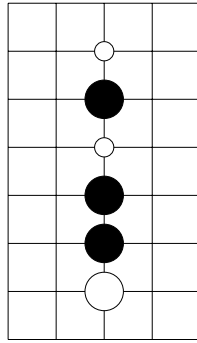
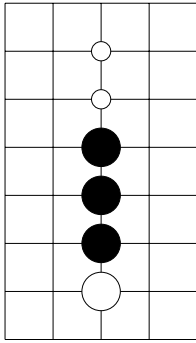


图 5 眠三

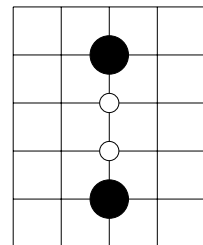
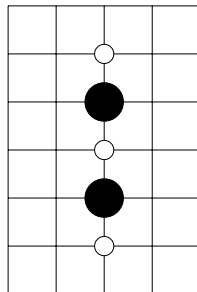
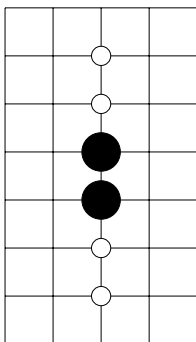


图 6 活二

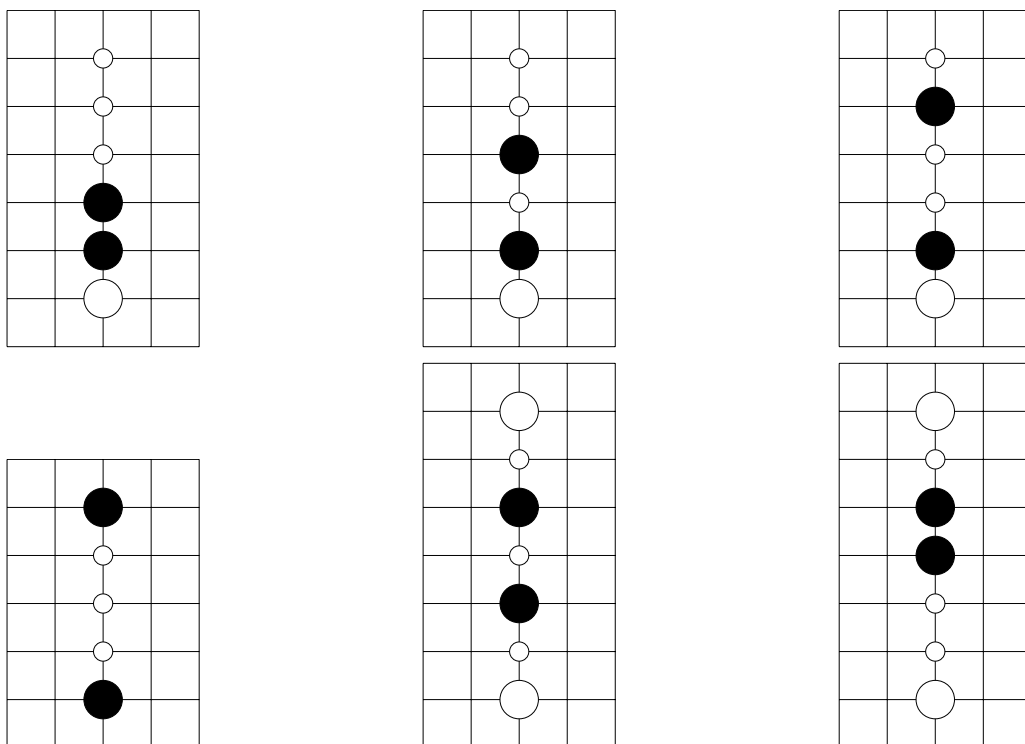


图 7 眠二

2.2.8 死四

两头都被封堵的四。其表达式形如

211112

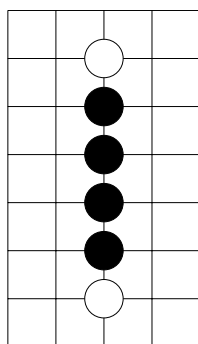


图 8 死四

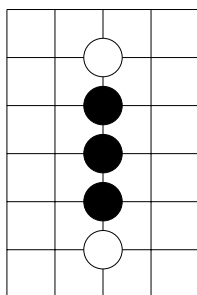


图 9 死三

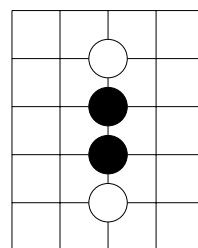


图 10 死二

2.2.9 死三

两头都被封堵的三。其表达式形如

21112

2.2.10 死二

两头都被封堵的二。其表达式形如

2112

2.3 估值函数

我首先考虑该子落下后，会在横、竖、撇、捺四个方向各形成什么棋型，然后进行打分；接着考虑落子的位置，一般同分情况下越靠近中心的点越好；并且还要考虑落子后会打破对方什么阵型，若该位置被对方占据会造成什么后果，来综合考虑攻防效果。

3 实验结果

我使用 Python 3.6.0 编写了所有程序，并且利用 John M. Zelle 博士设计的graphics.py库制作了图形界面 [4]，如图 11所示。

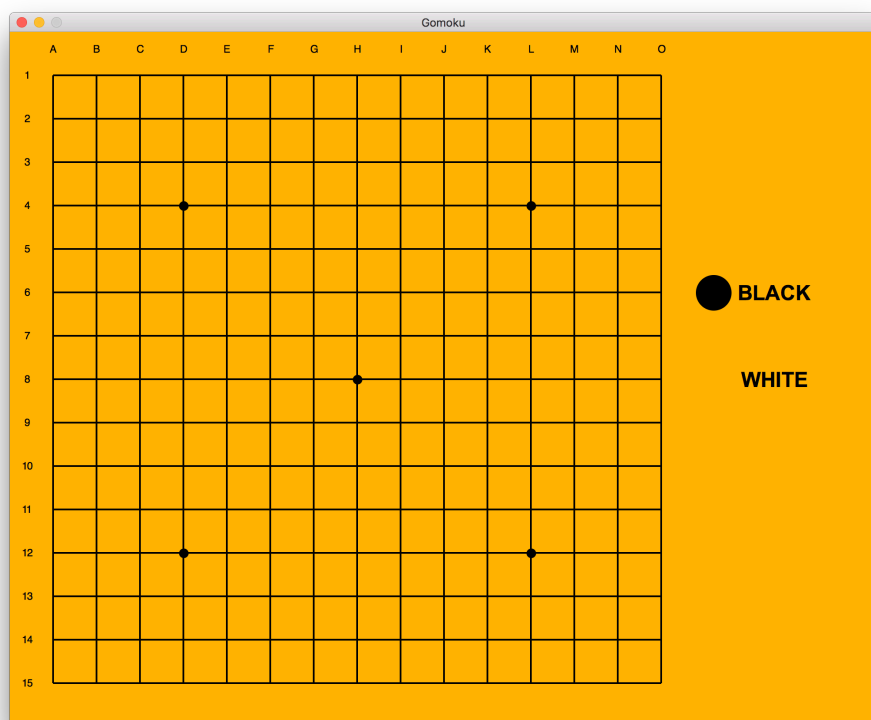


图 11 游戏开始界面

当玩家落黑子后，计算机会使用极大极小算法找到它认为的最佳落点，并置白子。两方对垒的效果如图 12所示。

若玩家赢得游戏，界面如图 13所示。

若计算机赢得游戏，界面如图 14所示。

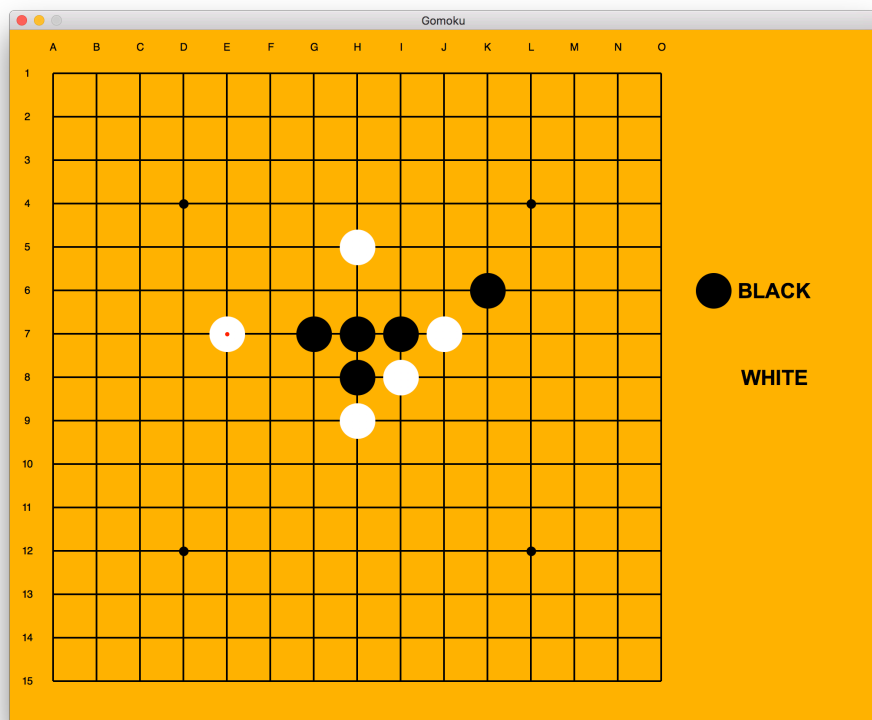


图 12 两方对垒过程效果

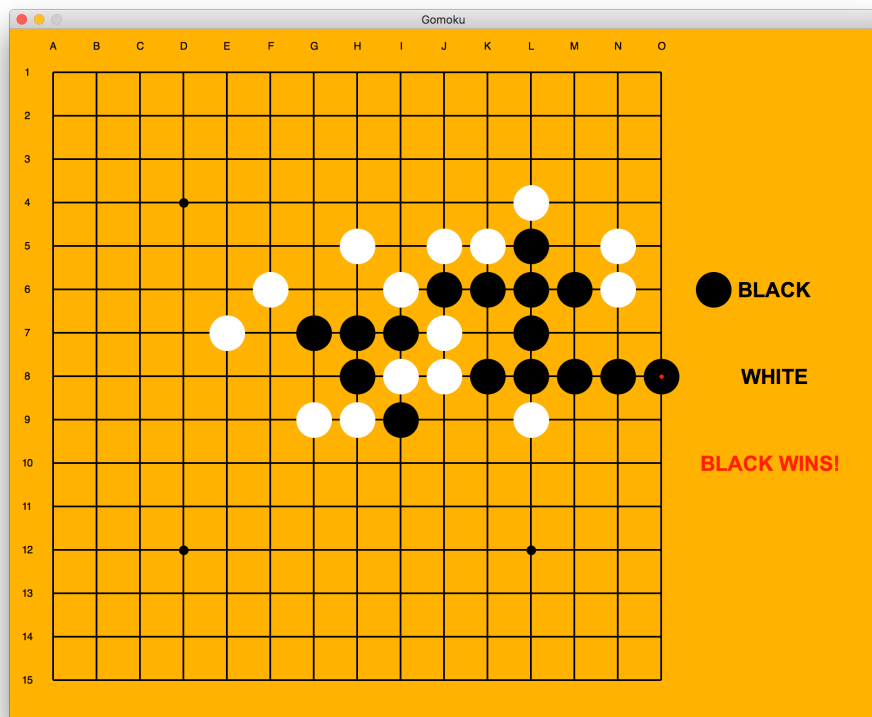


图 13 黑子赢得游戏

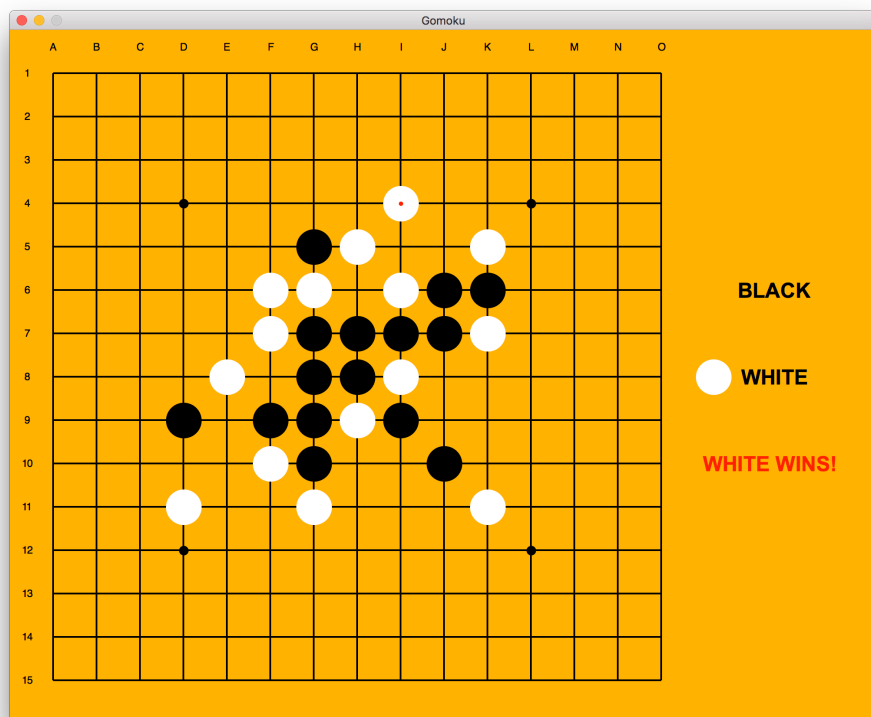


图 14 白子赢得游戏

参考文献

- [1] 人工智能, Nils J. Nilsson 著, 郑扣根, 庄越挺译, 潘云鹤校, 北京: 机械工业出版社, 2000 年 9 月
- [2] https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
- [3] <http://blog.csdn.net/cjianwyr/article/details/54911444>
- [4] <https://tieba.baidu.com/p/2180847383>
- [5] <http://mcsp.wartburg.edu/zelle/python/graphics.py>