

华东师范大学计算机科学技术系上机实践报告

课程名称：人工智能	年级：2016 级	上机实践成绩：
指导教师：周爱民	姓名：汪春雨	创新实践成绩：
上机实践名称：五子棋	学号：10152150127	上机实践日期：2018/4/24
上机实践编号：No. 3	组号：	上机实践时间：

一、 问题介绍

（本节介绍需要求解的问题是什么，为什么要采用我们介绍的方法求解）

1.1 概要

实现交互式五子棋程序，采用博弈算法实现

1.2 五子棋简介

五子棋（两人对弈的策略型棋类游戏）

五子棋是世界智力运动会竞技项目之一，是一种两人对弈的纯策略型棋类游戏，是世界智力运动会竞技项目之一，通常双方分别使用黑白两色的棋子，下在棋盘直线与横线的交叉点上，先形成5子连线者获胜。

棋具与围棋通用，起源于中国上古时代的传统黑白棋种之一。主要流行于华人和汉字文化圈的国家以及欧美一些地区，是世界上最古老的棋。

容易上手，老少皆宜，而且趣味横生，引人入胜；不仅能增强思维能力，提高智力，而且富含哲理，有助于修身养性。已在各个游戏平台有应用。

规则

- (1) 对局双方各执一色棋子。
- (2) 空棋盘开局。
- (3) 黑先、白后，交替下子，每次只能下一子。
- (4) 棋子下在棋盘的空白点上，棋子下定后，不得向其它点移动，不得从棋盘上拿掉或拿起另落别处。
- (5) 黑方的第一枚棋子可下在棋盘任意交叉点上。
- (6) 轮流下子是双方的权利，但允许任何一方放弃下子权（即：PASS权）

1.3 术语

对局语：

〔黑方〕执黑棋一方的简称。

〔白方〕执白棋一方的简称。

〔胜局〕有一方获胜的对局。

〔和局〕分不出胜负的对局。

〔终局〕对局结束。

〔复盘〕对局双方将本盘对局全过程的再现。

行棋语：

〔阳线〕即：直线，棋盘上可见的横纵直线。

〔交叉点〕阳线垂直相交的点，简称“点”。

〔阴线〕即：斜线，由交叉点构成的与阳线成 45° 夹角的隐形斜线。

〔落子〕棋子直接落于棋盘的空白交叉点上。

〔轮走方〕即“行棋方”，有权利落子的黑方或白方。

〔着〕在对局过程中，行棋方把棋子落在棋盘无子的点上，不论落子的手是否脱离棋子，均被视为一着。

〔回合〕双方各走一着，称为一个回合。

〔开局〕在对局开始阶段形成的布局。

〔连〕同色棋子在一条阳线或阴线上相邻成一排。

〔长连〕五枚以上同色棋子在一条阳线或阴线上相邻成一排。

〔五连〕只有五枚同色棋子在一条阳线或阴线上相邻成一排。

〔成五〕含有五枚同色棋子所形成的连，包括五连和长连。

〔四〕在一条阳线或阴线上连续相邻的5个点上只有四枚同色棋子的棋型。

〔活四〕有两个点可以成五的四。

〔冲四〕只有一个点可以成五的四。

〔死四〕不能成五的四。

〔三〕在一条阳线或阴线上连续相邻的5个点上只有三枚同色棋子的棋型。

〔活三〕再走一着可以形成活四的三。

〔连活三〕即：连的活三（同色棋子在一条阳线或阴线上相邻成一排的活三）。简称“连三”。

〔跳活三〕中间隔有一个空点的活三。简称“跳三”。

【眠三】再走一着可以形成冲四的三。

【死三】不能成五的三。

【二】在一条阳线或阴线上连续相邻的5个点上只有两枚同色棋子的棋型。

【活二】再走一着可以形成活三的二。

【连活二】即：连的活二（同色棋子在一条阳线或阴线上相邻成一排的活二）。简称“连二”。

【跳活二】中间隔有一个空点的活二。简称“跳二”。

【大跳活二】中间隔有两个空点的活二。简称“大跳二”。

【眠二】再走一着可以形成眠三的二。

【死二】不能成五的二。

【先手】对方必须应答的着法，相对于先手而言，冲四称为“绝对先手”。

【三三】一子落下同时形成两个活三。也称“双三”。

【四四】一子落下同时形成两个冲四。也称“双四”。

【四三】一子落下同时形成一个冲四和一个活三。

1. 4 输入输出

输入：棋盘状态，轮走方

输出：最佳落子位置

* 算法之外的图形界面再次不再赘述，已有的图形界面加以修改

二、 程序设计与算法分析（python实现）

（本节介绍程序设计的分析过程，如数据结构定义，算法描述，算法框图等）

2.1 数据结构定义

a. 黑白方表示

1=黑方（玩家）

2=白方（AI）

0=空白

b. 棋盘状态

board: 15*15 的二维 list

```
self.__board = [[EMPTY for n in range(15)] for m in range(15)]
```

下图中第一维度为 i，第二维度为 j

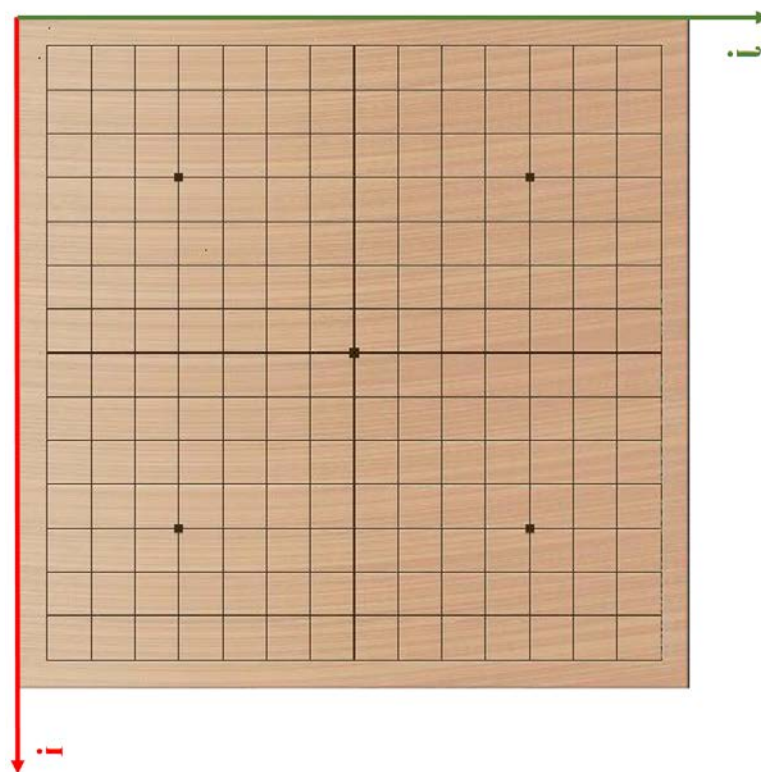


图 1 坐标示意图

c. 输出坐标 (i,j)

2.2 算法描述;

2.2.1主程序搜索算法:

a) $\text{Alpha} = -\infty$, $\text{beta} = \infty$, $\text{turn} = 2(\text{white})$, $\text{depth}=0, 1, 2, 3, \dots$

b) `serch (turn, depth, alpha, beta)`

1) if $\text{depth}=0$, return `evaluate(board, turn)`

2) $\text{score} = \text{evaluate}(\text{board}, \text{turn})$

3) if $\text{score} \geq 9999$, return score , game over

4) $\text{moves}[] = \text{generateNextMove}()$, $\text{bestMove} = \text{None}$

5) for move in moves:

- i) move on board, turn reversed
- ii) `score = - self.__search(nturn, depth - 1, -beta, -alpha)`
- iii) undo move on board
- iv) if `score > alpha`:
 - `alpha = score`
 - `bestMove = move`
 - if `alpha >= beta`: break
- 6) If `depth=maxdepth`, return `alpha`, `bestMove`

2.2.2 评估函数

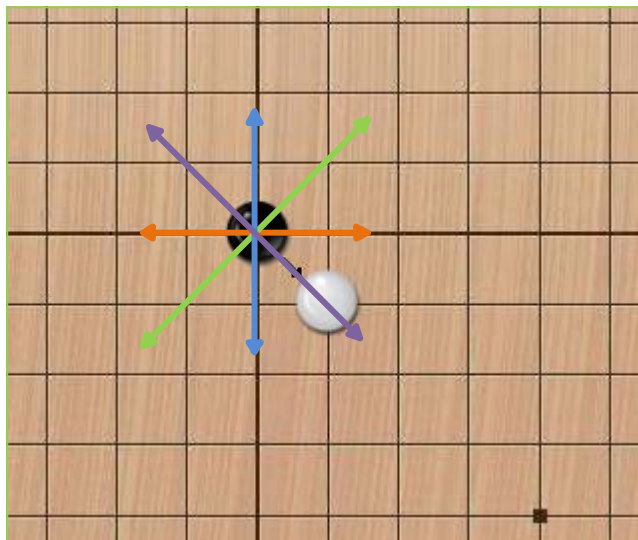


图2 方向示意图

Evaluate算法:

输入: 棋局状态 (board), 轮走方 (turn)

1) for each piece in board:

a) 分析水平方向

b)分析垂直方向

c)分析左斜方向

d)分析右斜方向

累计双方每种棋局出现次数（如冲四，冲三等）

2)得到每种棋局出现次数，对其进行评分（AI为白方）

黑白方分数归零：wscore=bscore=0

五连：立马返回分数9999

两个冲四相当于一个活四

白活四：9999

白冲四：9980

黑活四：-9970

黑冲四或活三：-9960

白活三或黑冲四为0:9950

黑活三>1 且 白冲四、活三、冲四=0： -9940

白活三>1 wscore+=2000, 白活三=1 wscore+=200

黑活三>1 bscore+=2000, 黑活三=1 bscore+=200

冲三一个10分，活二一个4分， 冲二一个1分，分别累加至wscore, bscore

棋子越靠近棋盘中心权值越高，中间为7，边界棋子为0

遍历所有棋子，累加至双方分数wscore, bscore上

return wvalue - bvalue

2.3说明

2.3.1 难度系数说明

难度1: naïve, 搜索深度为1

难度2: hard, 搜索深度为2

难度1: hell, 搜索深度为3

2.3.2 alpha, beta剪枝

利用alpha, beta剪枝减少搜索范围

```
58     if score > alpha:
59         alpha = score
60         bestmove = (row, col)
61     if alpha >= beta:
62         break
```

三、 实验结果

(本节列出实验的结果, 必要时加入一些自己的分析)

1、 棋局示例

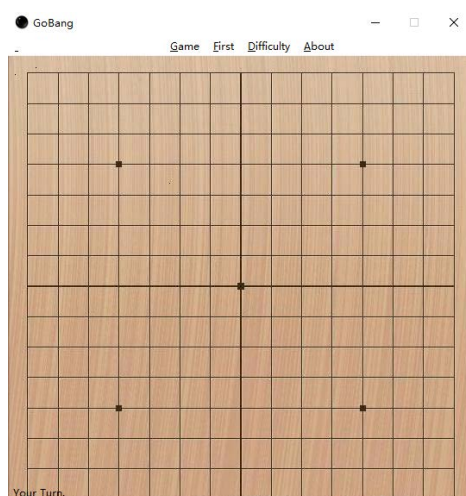


图3 界面示意图

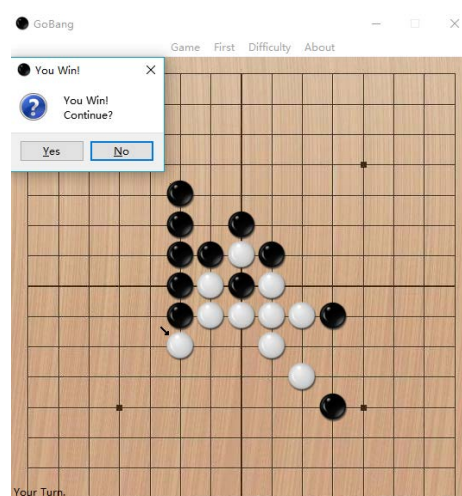


图4 难度1

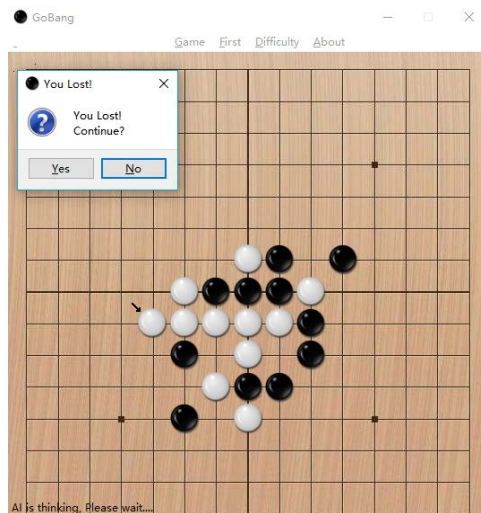


图5 难度2

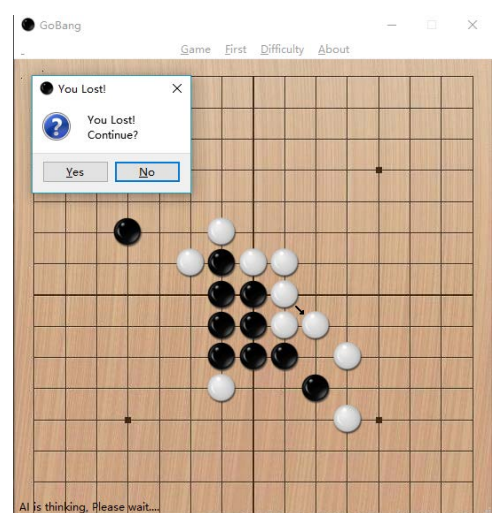
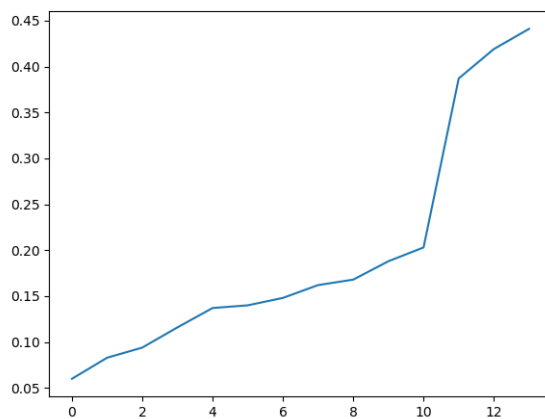


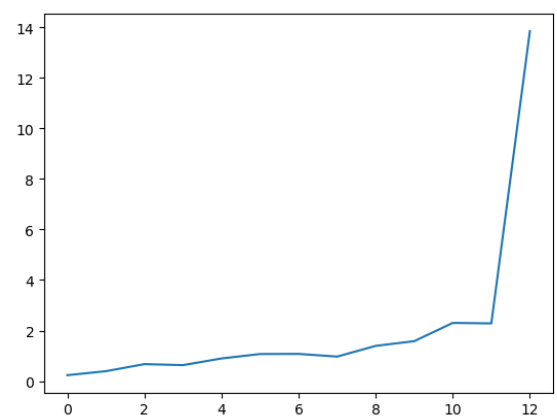
图4 难度3

2、分析

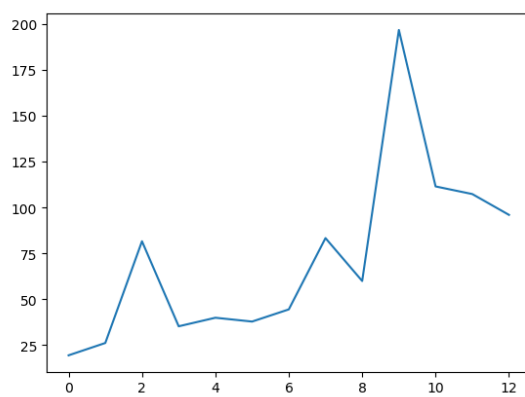
2.1 时间评估；（注意纵坐标时间刻度范围的不同）



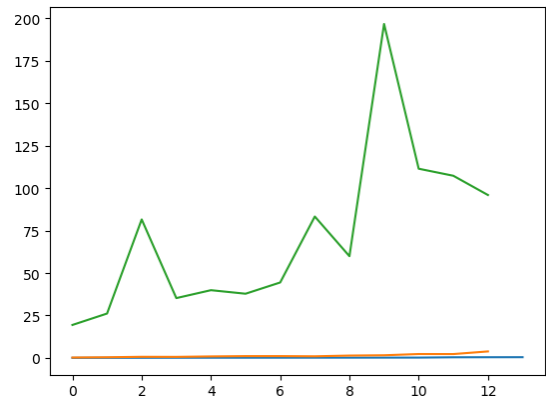
难度1



难度2



难度3



对比图

横坐标为第几步，纵坐标为时间s

随着步数的增加，时间增加明显。

时间随搜索深度的增加呈指数级增长。

2.2 alpha, beta 剪枝的影响

在搜索深度过小的情况下，alpha, beta剪枝的效果不明显，但是在深度加深的同时，剪枝的效果会逐渐显现，但是由于电脑性能受限，深度超过3的测试无法进行，而搜索深度在3以下的情况有无剪枝对搜索时间的影响不是很大。

四、 附件

（本节非必须的，可以列出源代码等，但是要把格式组织好）

pycharm工程文件夹	AI_GoBang_py
boardMap.py	棋盘类
AI.py	AI类
Evaluate.py	评估方法
Start_ui	启动图形界面