

五子棋人机对弈实验报告

PB16111245王立峰

Overview

由于时间仓促，最近实验太多，时间没规划好，再加上五子棋的实验有点难，导致本实验没有做完，只做了一个大致框架而已。所以，就写一个实验报告来阐述一下设计思想。

五子棋人机对弈实验报告

Overview

实验目的

实验内容

主要函数及功能

实验总结

实验目的

建立博弈树，利用极大极小算法，进行 $\alpha\beta$ 剪枝，利用构造的评价函数来确定下一步棋的位置，从而实现五子棋的AI算法。

实验内容

1. 对每个点位给出两种颜色棋子的打分，分别存在两个15*15的数组里，数组下标代表点的位置。
2. 确定最大值所在数组之后，遍历该数组找出所有最大值对应的位置，然后对这些位置统计另一种颜色的棋子的分数，再选取一次最大值，从而确定要落点的位置。
3. 打分函数的设计：在四个方向分别统计然后相加。对于某一个方向的分数统计，则分为正反两个方向进行，统计的时候如果有连成5个则直接返回一个最大值（最高分）。其他情况则按不同情况设置不同的权重，触发结束某一个方向上的统计的事件如下：遇到异色棋子；空白格子超过两个；遇到棋盘边界。其中遇到异色棋子和棋盘边界均视为一边被堵死，相比空白来说适当减分，而1个空白相比于完全连续则应再适当减分，最后取10的次幂，以保证不同情况的优先级，即不至于出现因为下到位置A可以形成4个活2而放弃下可以形成1个活4的位置B。

主要函数及功能

```
int evaluate(position pos, state color, position (*pf)(position ,bool ));//给出落子位置和方向移动函数，返回该落子位置在该方向上的评分
```

```
int point(position pos, state color);//给出一个落子位置，返回该落子的得分
```

```
void whole_points(int points[][15], state color );//给定颜色，记录该颜色棋子下在每一处的得分
```

```
int best_posits(const int points[][15], position p_s[], int& count); //给出分数数组，找出最大值对应的位置（可能不止一个），返回分数最大值
```

```
position right_up(position pos,bool dir);//位置函数，用于右上左下移动棋子并判断是否越界
```

```
position left_up(position pos,bool dir);//位置函数，用于左上右下移动棋子并判断是否越界
```

```
position left(position pos,bool dir);//位置函数，用于左右移动棋子并判断是否越界
```

```
position up(position pos,bool dir);//位置函数，用于上下移动棋子并判断是否越界
```

```
position set_chess();//确定落子
```

整个程序的主体框架如上所示，不过由于时间有限，目前还没有详细实现，等后面有时间还会详细去做的。

实验总结

虽然实验没有做完，但通过本次五子棋的实验还是能领略到五子棋的魅力的。在github上找了一个python实现的五子棋，然后和其对局发现，其搜索深度为3的AI还是比较厉害的，前期的话比我这个搜索深度为1的人脑厉害不少，基本上都是它下我堵，后期逐渐变成我下它堵了，不过总体来说还是比较厉害的。但是搜索深度为3感觉还是时间成本太高了一点，一局下了一下午的时间，尤其到后期，AI走一步需要计算几十分钟的时间，感觉计算力还是跟不上的，有必要利用并行计算来提升一下计算力。甚至有的时候也不能完全遵循最好评分去下棋，而是可以折中的方式，可以随机选择一个较好的，在某一水平线上的方式去下棋。由于实验没有做完，导致只能交一份报告，我感到很抱歉，但心里还是想助教能酌情给分，万分感谢！