南 京 理 工 大 学

智能计算技术实验二

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名:** | 蒋旭钊 | **学 号:** | 918106840727 |
| **学院(系):** | 计算机科学与工程学院 | | |
| **专 业:** | 计算机科学与技术 | | |
| **课 程:** | 智能计算技术 | | |

2021 年 11 月

# 一．问题重述

实现章节《5.6博弈树的启发式搜索》中，5\*5格子的一字棋问题，要求MAX方和MIN方都用博弈树来决策，运用极大极小分析法，同时加入a-β剪枝策略。

# 二．算法介绍

**极大极小分析法：**

（1）设博弈的双方中一方为A，另一方为B。极大极小分析法是为其中的一方（例如A方）寻找一个最优行动方案的方法。

（2）为了找到当前的最优行动方案，需要对各个方案可能产生的后果进行比较。

（3）为了计算得分，需要根据问题的特性信息定义一个估价函数，用来估算当前博弈树端节点的得分，称为静态估值。

（4）当端节点的估值计算出来后，再推算出父节点的得分。

（5）如果一个行动方案能获得较大的倒推值，则它就是当前最好的行动方案。

**博弈树的启发式搜索算法：**

（1）k=1，初始棋局Sk= S1;

（2）如果棋局Sk是终止节点棋局，则算法成功终止；否则，由棋局Sk生成A方所有可能的或关系子节点Si(i=1,2,…,n)。

（3）对每一个或关系子节点Si，生成其B方所有可能的与关系子节点Sj (i=1,2,…,m)。生成节点数为n×m+1的部分博弈树。

（4）计算每个与关系子节点Si的启发函数值。

（5）分别由m个与关系子节点倒推计算其父节点（与节点）的启发函数值:



1. 由n个或关系子节点倒推计算其父节点Sk（或节点）的启发函数值：



（7）A方从Sk的n个或关系子节点中选择节点i作为最优行动方案，获得棋局Si。

（8）B方从Si的m个与关系子节点中选择节点j作为最优行动方案，获得棋局Sj。

（9）若节点j是端节点，则算法终止；否则令k=j，转步骤（2）。

**a-β剪枝策略：**

（1）对于一个与节点MIN，若能估计出其倒推值的上界β，并且这个β值不大于MIN的父节点（一定是或节点）的估计倒推值的下界α，即α≥β，则就不必再扩展该MIN节点的其余子节点了。这一过程称为α剪枝。

（2）对于一个或节点MAX，若能估计出其倒推值的下界α，并且这个α值不小于MAX的父节点（一定是与节点）的估计倒推值的上界β，即α≥β，则不必再扩展该MAX节点的其余子节点了。这一过程称为β剪枝。

# 三．实现思路

主要有三个java类来实现：

* Class AIGobang

定义了棋盘大小BOARD\_SIZE，以及无穷大MAX\_VAL、无穷小MIN\_VAL、平局DRAW。

此外实现了Human和AI双方下棋位置的展示以及每个棋盘的显示，最终给出双方的终局结果。

* Class BoardState

定义了博弈树中每个节点的状态，包括生成该博弈树的角色mainPlayerTurn、要落子的角色playerTurn、结点的alpha、beta值，以及博弈树搜索最大深度MAX\_DEPTH(=2)。

棋盘上alpha、beta值以及状态估值的更新，通过getScore（）函数递归更新，剪枝通过：

if (this.alpha>=this.beta){  
 break;  
}进行判断。

* Class Utils

定义了要用到的各种函数，包括根据棋盘落子算出静态估值的函数getCost、生成新棋盘的函数generateNewBoard（）等等。

四．实验结果

本次实验不需要输入，输出展示了下棋的棋盘状态以及走子情况，最后判断双方的终局结果。

输出样例：

