**2022年春季学期**

**数据结构课程设计第5-7次课赛道B**

**实验报告**

周泽焜1 刘浩尘2董致彤1毛梓鉴1

1计算机科学与技术学院2020级28班

2计算机科学与技术学院2020级29班

# 分工与合作

周泽焜（组长）：主要负责任务的分配与方案的制定，参与估值函数的优化与调试，参与博弈函数的设计，参与策略部分的实现。同时负责小组蒙特卡洛方案的实现。

刘浩尘：负责查取资料，选择合适的估值函数，并对估值函数进行实现以及优化。其所负责的估值函数是我们程序的核心

董致彤：负责博弈函数的实现，参与策略部分的实现，参与蒙特卡洛方案的实现

毛梓鉴：负责策略部分的实现，同于参与蒙特卡洛方案的实现

# 算法思想

## 总体思路

整个程序的框架是估值函数加上博弈树，并以策略为辅。估值函数采用以双距离为主，结合潜力值与进攻机动性对局面进行估值。博弈树设计为两层剪枝博弈函数，尽可能减少枝杈，提高精确性。策略点主要包括双桥判断与连接，无用点判断，边界判断等。

## 所用方法的特别、新颖或创新之处

我们采用两层博弈树与纯估值函数进行结合，以减少一步贪心策略带来的不足。具体实现上我们以第二层得出的最低值代替第一层得出的值。同时为了避免无法成功完成搜索，我们将第一层结点设置为八个，取第一层最大的八个结点。经过我们的统计，前八个点几乎一定能包含最终性价比最高的那个点。之后我们对此八个点求第二层结果，找出第二步对我方最优的点，将其变为敌方的棋子，重新对局面进行估值，并将估出的值传回第一层，最后取第一层中值最大点。如图1所示，倘若只有一层，便会返回点1，但经过第二次判断后便会返回实际价值更高的点3。

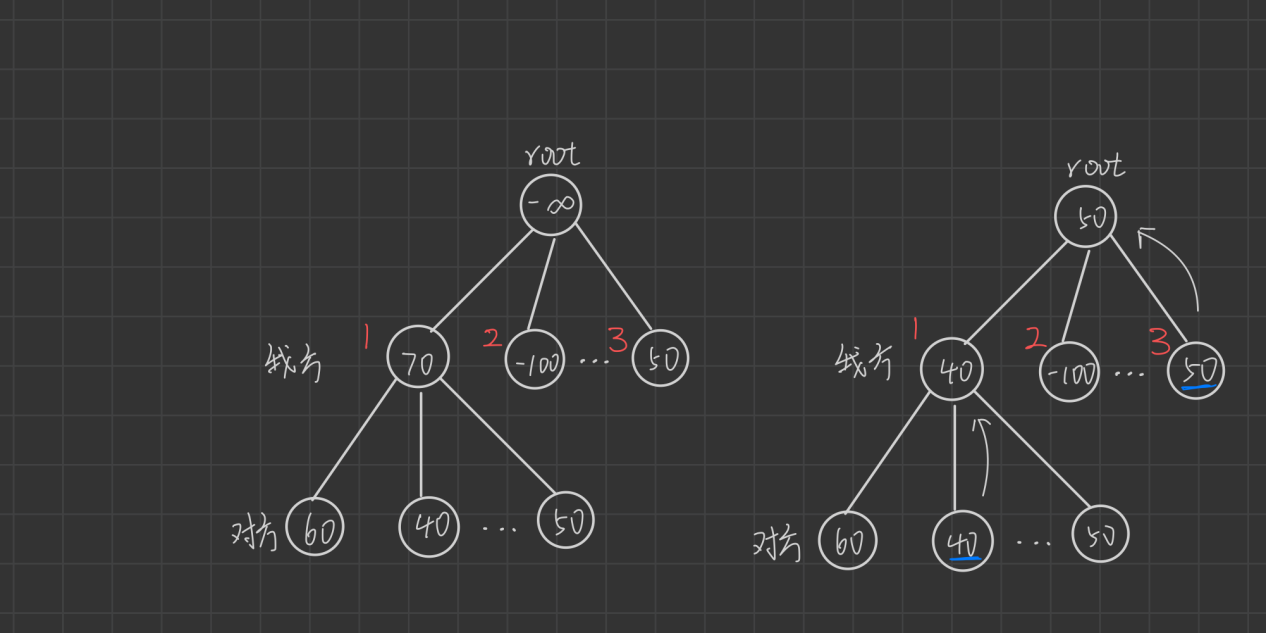


图1

在估值函数优化过程中，我们采用动态规划的思想对其进行了优化。在以双距离为依据的估值函数中，找邻居的过程最为耗费时间，我们针对这个找邻居的过程，进行了优化。我们对我们找邻居的过程分析后发现我们会对某些点重复查找邻居，尤其是某些实心点。因此，综合程序实现的复杂度，以及降低程序运行时间的效率，我们进行了以下优化。

1.按如图所示顺序对各个节点求邻居

2.对于空节点和边界节点，我们仍使用原有的求邻居函数

3.对于对方节点，我么不对其求邻居

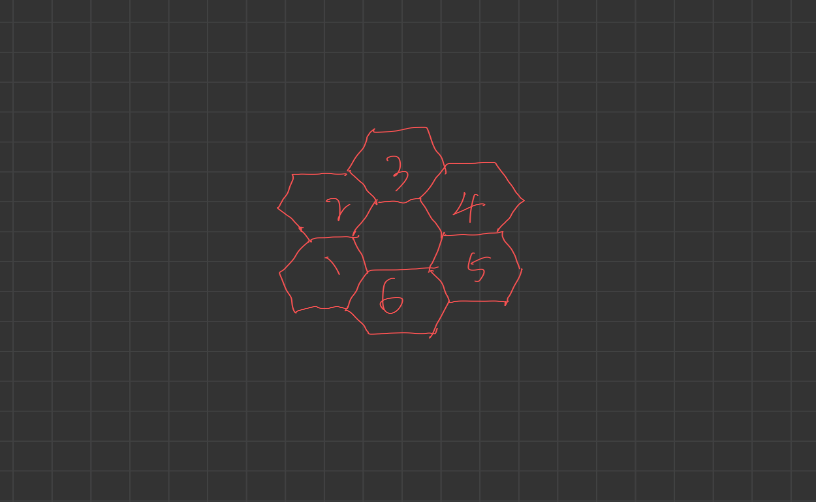
4.对于我方节点

(1) 节点的考虑顺序为1、2、3、4、5、6

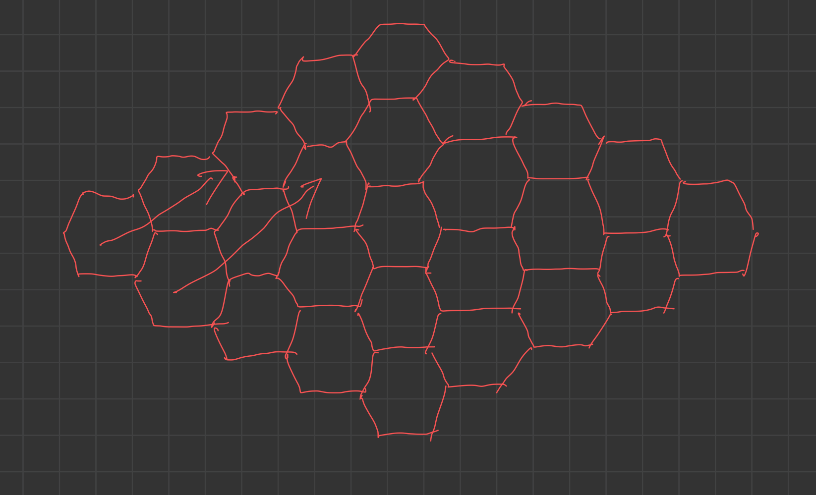
(2) 对于节点1、2、3，若其中有我们的节点，则将其邻居加入到当前链表中，找邻居工作完成。否则将空节点加入

(3) 对于节点4、5、6，空节点直接加入，对方节点忽视，若为己方节点，则调用原本的求邻居函数，将返回值加入邻居链表中

经过我们的优化与实际测试，我们将时间平均缩小了三倍。

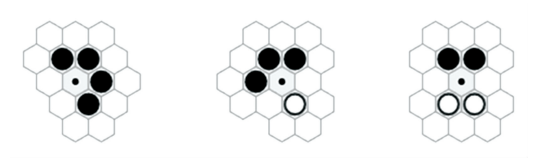


节点编码



查找顺序

在课设进行的初始阶段，纯策略的算法已经初具规模，它可以作为一个单独的游戏bot对局面进行判断，当双桥被拦截时，策略进行双桥连接，并对边界情况进行棋盘扩充，以实现边界双桥判断；当在前进方向的双桥桥心被对方棋子占领时，我们会将另一个桥心填充；并且纯策略算法作为一个单独bot，它在游戏开局时期会主动在前进方向搭建双桥，并在边界可能触碰时，优先调用边界连接函数；开局时一直沿用的weight数组在每次落子时更新，weight记录的是每个空间的权值，当以上所有情况均未出现时，采用weight优先落子，保证bot的可靠性；当先手后手不同时，策略遍历的方向不同；除此之外还有外三层和里层的无用点判断，减少非必要落子的情况。在最终阶段策略主要起到估值博弈的辅助作用，在奠定胜局后锁定胜利，避免失误。



# 总结

我们编写程序之初采用的方案是蒙特卡洛树加上估值函数。当我们完成后，发现由于估值函数的耗时性，1s时间内并不能模拟出足够的数据，导致我们的初代成品每次评估出的点并不是“好点”。对此，我们采取两步优化并行，一是估值函数的优化，减少其耗费时间；二是放弃蒙特卡洛，转而研究博弈树。

最终研究出我们成品的初代版本。我们此时的bot已具备不错的智能，但经过我们的测试发现我们在与对手对弈时，最终一旦局面变为我们即将获胜，我们的bot便会丧失智能。

对此，我们通过大量观察棋局，并测试，发现当我们的bot得出我们必胜后，由于其评判机制，它会认为我们的胜利会对局面造成极不利的影响。对此，我们在开始评测前对我们的代码进行分析，一旦我们进入必胜局面，我们便不会调用估值函数与博弈树，转而使用我们之前完成的纯策略函数，以此保证我们在必胜局面下依旧会赢，不会主动放弃。

其后，我们通过观察他人对弈棋局，摸索策略，找到了几个关键点，包括（7,3）（4,7），使我们在某些情况下，占据极大的先手优势，从而得到了大的增强。

我们的创意之处主要体现在对博弈树的改进，使其在只有两层的情况下得到增强。策略点中那几个关键点的查找也至关重要。

待改进之处主要在于我们的估值函数，如果我们的估值函数运算时间能够进一步缩短，我们可能能造出三层甚至四层博弈树，使我们的结果更精确。

# 参考文献

【1】《Are Bees Better Than Fruit flies Thesis》

【2】《基于UCT算法的Hex棋博弈系统的研究》

【3】《A-hierarchical-approach-to-computer-Hex\_2002\_Artificial-Intelligence》