  人机版五子棋是很有挑战性的。至今好像没有保证可以取胜的算法，但已经有不少写的很专业的五子棋程序了。我在编写五子棋的过程中参考了不少资料，发现人机五子棋大致有两种策略。在这儿总结一下，与大家共享。先说两种算法，然后介绍一点五子棋的专业知识。有些地方表述不清，望见谅，如有表述不清的地方，可以回帖讨论。

此类算法有一点是相同的，就是要对整个棋局或其中的有效位置进行评价。往往会使用一个分表。而评分表却很难确定，也没有所谓最好的，有人根据经验和测试，总结了不错的评分表，我在程序中都是用的别人的评分表。

第一种方法是很简单的，但是效果也是很好的。从运行结果看，这么简单的方法达到如此的水平，着实让我吃了一惊。当然，这是跟第二种算法比较的感觉。大致思路如下：

评估当前棋局中，哪个位置的得分最高。五子棋要赢，必然要有五个棋子在一起成线，那么我们就可以计算棋盘中每一个五格相连的线，一下称之为五元组。一般情况（包括专业五子棋）下棋盘是15\*15的。那么应该是572个五元组。同时，针对五元组中黑子和白子的数量（可以不考虑相对位置）的不同，给该五元组评不同的分。然后每一个位置的得分就是包含这个位置的所有五元组的得分之和。 下面是我的程序中用的表示方法及其评分表。

说明：当五元组为空，分数为7，不为零的原因是，还有跟糟的情况：五元组中既有黑子又有白子，五元组也就是无效了，这时才给0分。这个评分表是我在一个老外的程序中找到的，这是我见到的几个评分表中最出色的。

思考：B代表电脑的棋？？？

**[c-sharp]** [view plaincopy](http://blog.csdn.net/onezeros/article/details/5542379)

1. // tuple is empty
2. Blank,
3. // tuple contains a black chess
4. B,
5. // tuple contains two black chesses
6. BB,
7. // tuple contains three black chesses
8. BBB,
9. // tuple contains four black chesses
10. BBBB,
11. // tuple contains a white chess
12. W,
13. // tuple contains two white chesses
14. WW,
15. // tuple contains three white chesses
16. WWW,
17. // tuple contains four white chesses
18. WWWW,
19. // tuple does not exist
20. Virtual,
21. // tuple contains at least one black and at least one white
22. Polluted

25. tupleScoreTable[0] = 7;
26. tupleScoreTable[1] = 35;
27. tupleScoreTable[2] = 800;
28. tupleScoreTable[3] = 15000;
29. tupleScoreTable[4] = 800000;
30. tupleScoreTable[5] = 15;
31. tupleScoreTable[6] = 400;
32. tupleScoreTable[7] = 1800;
33. tupleScoreTable[8] = 100000;
34. tupleScoreTable[9] = 0;
35. tupleScoreTable[10] = 0;

 至于如何有效统计这么多五元组的得分，就自己想吧，我在写的一个程序中第一次写为了便于自己和朋友理解，居然计算了两遍，但是条理清晰。这就是在可读性和效率之间来个折中了。

 就这种算法，我写了一个程序，供大家参考：<https://github.com/zhijie/Gobang>

===============更新=========================

同时还有个中国象棋的源码 <https://github.com/zhijie/ChineseChess> vc版

后来学习android时，写了个android版的中国象棋，源码如下：

<https://github.com/zhijie/ChineseChess4Android>

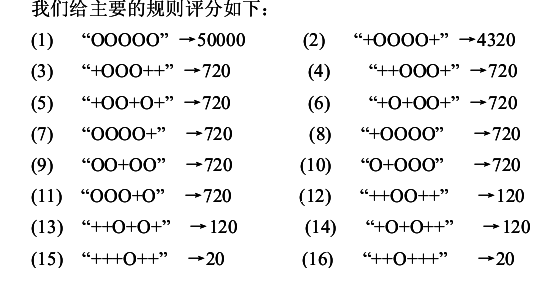
第二种算法是使用正规军中的博弈算法。在极大极小值搜索中应用alpha-beta剪枝。

使用这种算法，就是估计几步之内（步数越多，时间越长，自己根据实际情况决定）放在哪个位置最有利。根据当前局面，评估每一个可以落子的位置，看看在这儿落子后的得分怎样。极大极小值搜索就是在估计自己走的时候，选得分高的，估计别人走的时候，选得分低的，这儿的低是对自己而言的，也就是假设对手选了对他最有利的位置，淡然就是让我们得分最低的位置了。然后这样假设几步，选最好的。因为这样的搜索控件往往太大了，就是用剪枝，对没有必要的走步就不用分析了。

这种算法的评分方式跟第一种不同，第一种方法是评估当前棋局的每一个位置的得分，找最优的。而这里是对假设的每一种走步的局面进行评估，选能够到达最优局面的位置落子。

这种算法属于经典的而且应用很广的算法。很多书上都有介绍。我就不多次一举了。而且我也不一定能说清。

这里给出山东师范大学董红安在2005年的硕士毕业论文中使用的的评分表

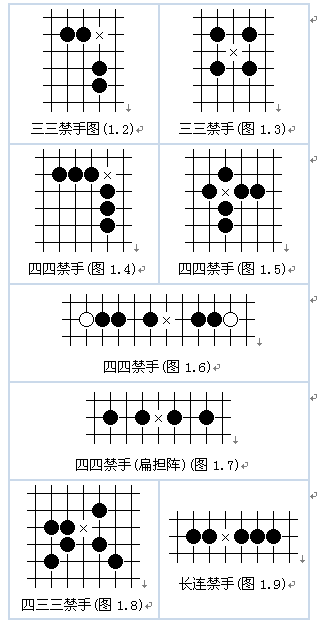


有时间我会把这种算法的五子棋写一下

下面提一下五子棋的专业知识

国内已有一些编程比赛是针对博弈的

那么就有了一些规定；在五子棋方面，比赛多使用专业五子棋比赛规则。主要是禁手，为了尽量公平，对先落子的一方（先手）下禁手，就是一些技术不能再用，用则为输。我找了一个禁手的图示



另外一点，为了使自己的程序更智能，先手一般会使用经典开局

因为根据经验，先手的某些开局赢的可能性更大。这种可能性的来源是，这种局面要有尽量多的两个子在一起的情况。这是26种经典开局，可以应用到自己的程序中

