

人工智能

五子棋 AI 设计

计算机应用技术

贾锋

2017532062

一. 五子棋 AI 的发展现状

既然要编写五子棋的 AI,那就应该先了解一下大家都是怎么做的.和围棋比起来,大家肯定认为五子棋是一种比较简单的游戏,前不久以 AlphaGo 为代表的一众深度学习博弈程序在围棋上可以说是基本打败了人类,那我们肯定会想当然的认为,复杂度较低的五子棋应该已经完全成为 AI 的领域了.然而实际情况却不是这样,

在国际上有一个五子棋比赛 Gomocup,这是目前规模最大的五子棋人工智能的国际比赛,而来自中国的“弈心”连续获得了 13,14,15,16,17,18 届 Gomocup 的冠军,而且在 2017 年,弈心成为首个在公开比赛中战胜人类顶尖棋手的人工智能程序.在四场比赛中,弈心并不是像 AlphaGo 那样绝杀零封人类棋手而是战的有来有回.弈心的成绩绝对可以称得上是最强的五子棋 AI,那么世界最强尚且如此,其他的 AI 我们有理由相信肯定是没办法在与顶级人类棋手的较量中占上风的.那么为什么会出现这种情况呢?这和五子棋的规则是有关联的.换句话说,我们之所以会觉得五子棋是一门简单的棋,是因为大多数人并不了解五子棋规则的演化和先行的规则,正是这些规则使得五子棋的 AI 设计变得扑朔迷离.

二. 五子棋的规则

五子棋最早是由中国发明的,但是对其规则演变起作用比较大的是日本,从五子棋的几种英文称呼例如"Gomoku,Renju"等来看,其中似乎带有一些日文的发音习惯,1988 年,日本、前苏联以及瑞典三个国家的连珠组织在瑞典成立国际连珠联盟 (Renju International Federation, 简称 RIF),其后很多国际比赛的举办和规则的制定都和这个组织有关系,同时瑞典也出现了许多高明的棋手.

1992 年,五子棋被证明在原始无禁手规则下先行必胜,2001 年被计算机证明原始有禁手规则下先行必胜,从此拉开了五子棋规则从简单到复杂到走火入魔的变革之路.

先走必胜的话,这个棋就完全没有下的必要了,最少是没有钻研发展的必要了.于是日本人发明了原始禁手规则,就是对先行的黑棋增加了三三禁手,一年以后,修改了禁手规则,三三禁手不但使黑棋少了一个获胜的手段,还让白棋多了一个进攻方法,"抓禁"这个词就诞生了,也就是说,白棋可以通过巧妙的布局逼黑棋在特定的点位防守让他形成禁手条件,从而使黑棋面对"不防守就输,防守算禁手也判输"的境地.

然而这还只是刚开始,随着博弈树等相关算法的逐步提出,计算机计算能力的进步,人们证明了三三禁手规则下仍然是黑棋必胜,所以在日本人的主导下,五子棋的规则继续修改,增加了四四禁手和长连禁手,至此黑棋取胜的棋形只剩下一个:三四,在三三禁手,四四禁手和长连禁手下,可以说我们身边绝大多数对五子棋的认识只停留在"连五个算赢"的水平的人来说就已经不会玩了,这个规则到底有多复杂会在后面给出图示,在一定程度上,这个规则限制了五子棋的推广普及.然后即使是这样,经过计算机推理证明,黑棋仍然必胜.

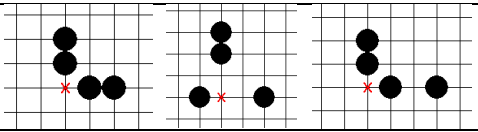
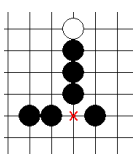
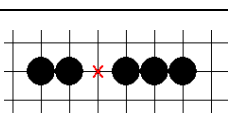
日本人从此走上了一条逐步使这个游戏变复杂的不归路,为了使这个游戏公平一点,他们考虑给黑棋增加更多的限制,首先引入的是"连打",就是在第五手的时候,黑棋要同时下两步棋,然后白棋在这两个棋子之间打掉一个,这就是"五手两打"规则,刚刚接触这些规则的话肯定觉得这没什么,实际上在高手之间的对决基本在第五六手就能锁定必胜路线,如果一个游戏完全变成了背谱的游戏,那么也就没什么可玩的了.

故事即将进入高潮.计算机上迅速的完成了证明:在三大禁手加五手两打规则下,黑棋必胜.为了国际比赛能继续办下去,"三步换手"规则被搞了出来,就是在第三步的时候,白棋可以选择交换棋子颜色,这样黑棋就不敢采用优势最大的开局,否则一旦被换手,将会难以收拾,这种换手规则在棋类游戏里可以说是绝无仅有,他使得五子棋不再是一个简

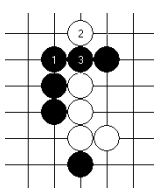
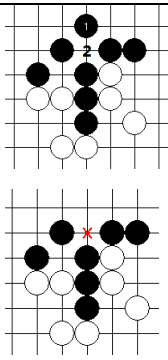
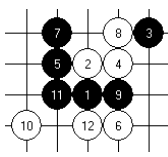
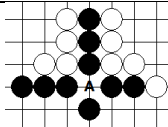
单的极大极小博弈过程,目前根本没有一个较好的算法能完美应对带换手规则的五子棋,然而这还没完,日本人的执着完美的体现在了这个问题上,为了以绝后患,更多的规则被发明出来,比如开局必须在 26 种标准开局中选择,黑棋第一手必须下在天元等等.

这个规则这么复杂想必很少有人喜欢,于是有什么山口规则,一手交换规则等等,制定规则就是一拍脑门的事,而针对规则编写算法可不能一蹴而就,这就是五子棋 AI 无法像围棋 AI 一样绝杀人类的一个原因,而且,极少有基于深度学习的五子棋 AI 出现,也是一个重要的原因.

先行禁手,执黑先行

①	双活三禁手 ：一颗棋落下，该子在棋盘上同时形成两个或者两个以上的活三，那么这个点就是双活三禁手点。	
②	双四禁手 ：一颗棋落下，该子在棋盘上同时形成两个或者两个以上的活四或者冲四，那么这个点就是双四禁手点。双四包括活四和眠四	
③	长连禁手 ：一颗棋落下，该子在棋盘上形成一个 6 颗或者超过 6 颗黑棋连在一起的模型，那么这个点就是长连禁手点。	

禁手规则

①	同时性，必须是同时形成。 当黑棋在 1 处活三，白棋 2 跳冲四逼迫黑棋在 3 处落子，然后指责黑棋形成双活三禁手，这个是不对的。因为竖线的活三和横线的活三并不是同时形成的，而是先后形成的。	
②	同子性，必须是刚落下的这颗棋形成。 当黑棋在 1 的位置冲四的时候，首先是斜线上形成了一个活三；其次，在同一时刻横线上原本由于在 2 位置的四四禁手而变成眠三的这个横线上的三却由于黑 1 的存在而变成了活三（因为这个时候可以在 2 的位置形成活四了，根据活三的定义，这个横线的三就是活三了）；那么这个时候白棋指责黑棋同时形成两个活三，是双活三禁手对么？ 明显是不对的。因为这个时候虽然没有违背同时性的原则，但是却违背了同子性原则，斜线的活三和横线的活三，并不是都由黑 1 构成。	
③	及时性，禁手出现，必须及时指出，过期无效。 如图 3-10，当黑棋 11 落下后，本来是在棋盘上出现禁手了，但是由于当时执白的同学并没有发现，而落下了白 12，当落下后，他突然发现了刚刚黑棋落了禁手，这个时候他指责黑棋刚刚 11 双活三禁手了。这样是不对的，因为当你走完 12 的时候已经默认为允许黑棋走双活三禁手了，所以禁手失效，这个时候黑棋可以继续将活三变成活四。	
④	连五最大 在棋盘上，如果连五和禁手同时出现，那么禁手失效，连五为胜。	

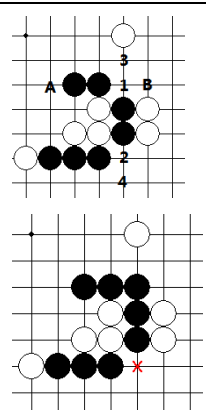
⑤ 多重性,后禁解前禁.

黑棋在 1 处落下, 形成了横线和竖线的两个三, 要判断这 1 点是不是双活三禁手点, 这里最重要的就是判断 1 处落子后, 横线和竖线的这两个三是不是活三。

对于横线的三, 由于可以在 A 点或者 B 点形成活四棋型, 根据活三的定义 (能够形成活四的三称为活三), 我们可以判断, 横线的三是活三。

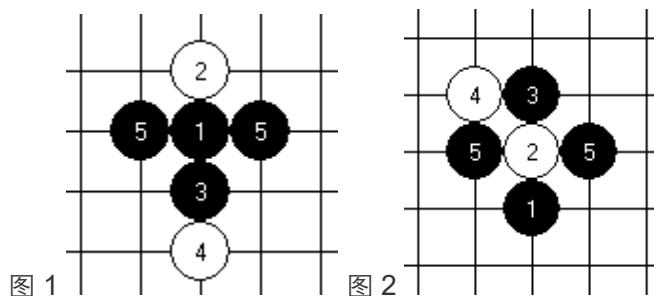
对于竖线的三, 在 2 的位置上, 我们原本是可以形成活四棋型的, 但是如果 1 点已经落黑棋, 在 2 点落黑棋将形成双活四禁手, 所以 2 点不能落子, 那么在 1 点落的黑棋就不能形成两个活三, 所以 1 点不是禁手。

这种情况称为“四四解三三”



三手交换五手两打

假先方 (即此时执黑的人) 摆完开局 (前三手) 后, 假后方 (即此时执白的人) 有提出交换的权利。如果他选择换手, 接着由执白方走第四手棋。(注意此时的情况是刚下完第二个黑子, 也就是棋盘上第三个子的那个人接过了白棋, 然后又下了一步白棋) 然后黑方在下黑 5 手时, 必须在盘面上的两个空白交叉点上各放一个黑子, 由白方在此两子中拿走其中一个黑子。但此两黑子放在棋盘的位置不能同形, 如图 1 所示, 图中的两个黑 5 就是属于对称的同形, 是不允许的。而图 2 所示的两个黑 5 并不同形, 因此是允许的。



三. 人机博弈五子棋程序的设计

接下来说我自己的程序, 考虑到我的 AI 水平并不能达到“黑棋必胜”的程度, 所以我打算在第一个版本中不考虑禁手规则和交换规则。

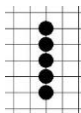
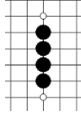
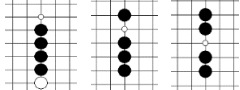
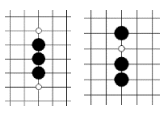
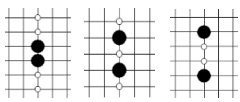
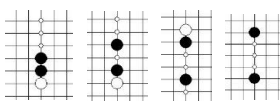
核心的算法按照课程要求采用 α - β 剪枝算法, 但是在实际操作中做了一些优化, 例如按照标准的算法, 每层的决策按照什么顺序依次探索并没有说明, 我们可以想象一下, 如果能对这些节点进行排序, 使得我们能提前发现必胜或者必败节点, 那么就可以大幅的减少搜索空间, 这是棋类游戏的一个特点, 一个比较好用的启发信息就是“历史表”, 当一个节点曾经以 β 节点的形式出现过的时候, 我们在历史表中给他打分为层数的平方, 然后以后的搜索中, 优先探查历史分比较大的节点, 我们有理由认为这个节点容易产生 β 截断, 类似的, 如果我们已经在“叫杀”状态了, 其实没必要进行深度搜索, 只需要发现叫杀点就行了, 所以原始的每次搜索相同的深度的剪枝方法并不适用, 这里采用的是“迭代加深”搜索方法, 即第一次搜索 2 层, 第二次搜索四层..., 直到发现必胜或必败状态或者达到最大深度。

在可行解生成方面, 如果我们选择所有的空位置作为可行解, 那么搜索空间就比较大, 而五子棋有着“聚堆抱团”的现象, 所以我们考虑只在有棋子的位置的附近搜索, 至于有多近, 我认为三格以外的地方就不用考虑了, 离得那么远就算是比较无聊的招法了。

四. 评价函数:局势分析

最重要的是估值函数,我采用的估值函数是根据五子棋贴吧的"基本棋形"来编写的,下面的表格给出了基本棋形和评分方法.

表一:基本棋形

序	解释	图示
①	连五：顾名思义，五颗同色棋子连在一起	
②	活四：有两个连五点（即有两个点可以形成五），图中白点即为连五点。	
③	冲四：有一个连五点，如下面三图，均为冲四棋型。图中白点为连五点。	
④	活三：可以形成活四的三在自己没有更好的进攻手段的情况下，需要对其进行防守，以防止其形成可怕的活四棋型。中间跳着一格的活三，也可以叫做跳活三。	
⑤	眠三：只能够形成冲四的三	
⑥	活二：能够形成活三的三	
⑦	眠二：能够形成眠三的三	

估值函数打分表

	情况	TAG	得分
	长连或 5 连	Long/Five	100000
	活四,双冲四,冲四活三	Alive4/DoubleRush4/ Rush4Alive3	10000
	双活 3	DoubleAlive3	5000
	活 3 眠 3	Alive3Sleep3	1000
	眠 4	Sleep4	500
	活 3	Alive3	200
	双活 2	DoubleAlive2	100
	眠 3	Sleep3	50
	活 2 眠 2	Alive2Sleep2	10
	活 2	Alive2	5
	眠 2	Sleep2	3
	死 4	Dead4	-5

	死 3	Dead3	-5
	死 2	Dead2	-5
	死 1	Dead1	-5

其中的死四表示无法形成连五的四,其他的死棋以此类推,这套估值函数的问题在于,进攻和防守主次不分,而且有的时候会错失叫杀的机会而转入防守.

五. 实验结果

程序经过了几次更新,目前版本的界面是这样的:



能显示棋子是第几手落下的,能标出上一步的位置,还有历史记录,可以选择人机对战和人人对战,人机对战中玩家可以选择使用黑子先行还是白子后手,AI 的等级有五级可以调整.界面采用 Qt 制作,编程语言采用 C++ 目前的代码量在 1700~2000 行之间.

代码可以在我的 Github 上找到: <https://github.com/AngelaViVi/Gomoku>

仓库中含有一个 Qt 版本,一个 unity3D 版本和一个 C# 版本,主要参考的程序是象棋巫师,他的源代码的修改版我也放在了仓库中,是一个 win32 版本.

目前的情况是 AI 执白的胜率一般,经常会放弃进攻转入防守,AI 执黑先行的胜率还可以.

六. 关于棋类 AI 的一些见解

我认为一个和人类对弈的棋类的 AI 应该是这样的:

- 1.拥有正常数量的开局存储.这个数量应该是一个优秀的棋手通过正常的训练能达到的.
- 2.采用正规比赛的规则
- 3.它必须要有正常的计算资源,如果他是用的是诸如天河一号之类的计算机或者有上万块显卡的 GPU 集群,那就根本没必要和人类比赛了.
- 4.它必须和人类有同样的时限,例如一个程序所拥有的经验是一个人类棋手一生都不达

不到的数量,那么比赛是完全不公平的.

如果一个 AI 不满足上述条件,我觉得他的诞生根本就不是为了打败人类,而是为了打败其他算法.所以说即便是人类可能再也不能在围棋这个游戏上战胜深度学习,但是这个游戏压根就不公平,AlphaGo 所拥有的计算资源和经验根本不是一个人类棋手所能比的,输给 AI 只是个迟早的事,退一步讲,未来我们可能会研发出足够大的存储器和足够快的 CPU,能够在极短的时间内对围棋进行穷举,然后证明围棋是先行必胜的,那我们也不必修改围棋的规则,因为这种计算根本是对人脑不友好不公平的.

就像弈心不是为了和人下棋而设计的,它是为了和其他 AI 玩 gomocup 才弄出来的,他能轻松的打败其他算法,但是和人打还是五五开,这就说明五子棋这个战场是用来检验这种空间搜索算法的而不是用来打败人拿出来显摆的,这也能解释为什么很少有人拿深度学习的程序上去比赛,同时这也是五子棋 AI 发展现状的一种解释.