

棋类游戏中的先行权

黄 晨

(复旦大学 化学系, 上海 200433)

摘 要:通过对一些典型的棋类和球类游戏规则考察,指出先行权问题在游戏规则中的重要作用,由先行权引发的规则问题甚至导致了诸如六子棋这样全新的棋类游戏的产生.还结合奇偶层效应对先行权作了量化,指出空着裁剪的本质同其他基于 Alpha-Beta 窗口的诸多裁剪算法是一致的,并阐述了诸多裁剪算法的作用被空着裁剪算法所掩盖的原因.

关键词:先行权;六子棋;计算机博弈;奇偶层效应;空着裁剪

中图分类号:TP18 **文献标识码:**A **文章编号:**1673-4785(2007)03-0091-04

The first-move advantage in board games

HUANG Chen

(Department of Chemistry, Fudan University, Shanghai 200433, China)

Abstract: The importance of the first-move advantage in rules of board and ball games is presented in this article. Consideration of the first-move advantage in Connect-Five lead to the invention of a new board game, Connect-Six. The value of the first-move advantage can be calculated after studying the even-odd effect in computer gaming programs. Since the first-move advantage can be evaluated, the principle of null-move pruning is identical to other pruning algorithms based on an Alpha-Beta window. This is the reason the functions of other pruning algorithms are concealed by null-move pruning in chess programs.

Keywords: first-move advantage; Connect-Six; computer games; even-odd effect; null-move pruning

象棋和围棋中有个术语叫“先手”,文中称为“先行权”.不管是棋类游戏、球类比赛还是在战场上,先手、球权、先发制人等术语都和主动权、优势联系起来.因此,对先行权的研究也就成了博弈、竞技和军事研究中的一个重要课题.

在棋界,先行权一直是棋类规则的重点讨论对象,由此也引发出很多争论,诸如围棋先手应该贴几目的问题.要解决由先行权引发的规则问题,首先要对先行权的本质有所了解,再对先行权作具体的量化,才能制定出一套合理的方案.通过对棋类规则的改进,全新的棋类游戏就会孕育而生,“六子棋”就是一个最典型的例子.人类自从20世纪中期计算机诞生以来就希望它会下棋,可是计算机的博弈水平直到90年代“空着裁剪”提出以后才有大幅度的提高.到现在为止,空着裁剪仍旧是大多数中国象棋和国际象棋博弈程序所采用的唯一的裁剪算法.为何空

着裁剪如此有效,而诸如 Multi-ProbCut 等著名算法在它面前却不值得一提,本文试图从先行权问题中寻找答案.

1 先行权及其相关规则

1.1 棋类游戏

棋类游戏的一大特点就是双方轮流下子,这样就会有先走方保持优势的问题.以中国象棋为例,对大量棋局的统计表明,先胜、先和与先负的比例是37:37:26,这说明先行方是占明显优势的,即先行方胜率为56%(换算成Elo等级分,先行方要高出40分)^[1].

和中国象棋类似,国际象棋、五子棋、围棋等,都存在明显的先行权,其中五子棋在没有禁手的规则下是先行必胜的.五子棋的禁手规则、围棋的贴目规则等,就是用来消除先行权的,这类规则通常会给予先后双方略微不同的限制.至于中国象棋和国际象棋,就棋类规则本身而言,先行权是无法消除的,然

而棋类以外的规则却有很大的灵活性,有学者建议给后方更多些时间,也不失为一种可以尝试的做法.

还有些棋类的规则看似怪异,实际上也是用来消除先行权的,只是它对先后手给予同样的限制,使一方的先行权有所削弱.一个最典型的例子就是古代围棋的“座子”,在 4 个星位分别放 2 个黑子和 2 个白子(对角交错),然后再开始对弈.由于先行方采用平行型布局要比交错型布局更有优势,因此开局前硬是给定交错型布局,会很大程度上削弱先行方优势.五子棋也可以对先后手方做同样的限制来削弱先行权,在日本有一种“提二子”的玩法,如果一步棋能夹住对方(当且仅当)2 个子,就可以把这 2 个子提走.这样,很多先行方扩大优势的手段就被限制了(如图 1 所示).

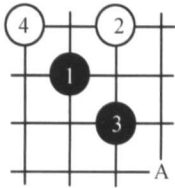


图 1 提二子规则

ig.1 Ninuki-Renju with a rule of sandwich Capturing

按照一般的五子棋规则(即便有禁手的情况下)走到白 4 都是先行方(黑方)必胜的,但在提二子规则下,黑只能走 A 位(不是必胜点),否则被白走到 A 位,就可以提掉 1、3 二黑子了.

1.2 从球类比赛中得到启发

其实很多球类游戏也有先行权的问题,比如网球、乒乓球是发球方占优势,排球、羽毛球则是接发球方占优势.换发球规则尽管形式不同(网球是每人发一局,乒乓球是每人发 2 球,都是轮流发球制;排球和羽毛球都是由赢球方发球的每球得分制,早期都是发球得分制),但目的就是让双方具有同等的先行权.

尽管如此,某些心理上的因素仍然会导致公平性的失衡,一个典型的例子就是足球比赛中的互射点球.在“一旦射失即将输掉比赛”的情况下,球员会承受巨大的心理压力,导致罚球命中率下降,所以一般来说先罚球的一方在心理上稍占上风.

规则上比较完善的是网球的抢七局,为了避免打成平局后总是由一方先发球,就采用先发一个球,此后一人连发 2 球的规则,这样就使得“一旦发球丢分就会输掉整盘”的情况轮流发生了.

1.3 六子棋

六子棋在规则上只和五子棋上有 2 点差异,一是“连成六子得胜”,二是“每次连下 2 子”(黑方第一步只下一子).尽管规则非常简明,但直到 2005 年这个规则才有完整的官方记录(首先由新竹清华大学的吴毅成教授发表在 2005 年的 ICGA 会议上).

不管六子棋的诞生是否受了网球抢七局发球制度的启发,但是“每次连下两子”的规则初衷显然是消除双方的先行权优势——每方在落子前盘面上总是比对方少一子,落子后总是多一个子^[2-3].

六子棋尽管规则简单,但“每次连下 2 子”给棋局带来无穷变数,这两个子既可以都用来进攻(制造“活四”),又可以都用来防守(围堵“活四”),也可以一个子进攻、一个子防守(由“活三”制造“活四”,再去拦截对手的“活三”).因此六子棋可能会取代围棋,成为世界上规则最简单,但最难以驾驭的一种棋.

2 先行权问题在博弈程序中的体现

2.1 奇偶层效应

象棋程序通常会用到“迭代加深”(iterative deepening)这个算法,即从 $n=1$ 开始做深度为 n 的 Alpha-Beta 搜索,逐渐增加搜索深度 n ,直到程序用足计划好的时间为止.采用迭代加深算法有时会出现一种很奇特的现象,例如使用象棋程序 Elephant-Eye 的“棋子-位置数组”^[4]做简单的局面评价,在起始局面会得到如表 1 所示的结果.

表 1 奇偶层效应
Table 1 Even-odd effect

深度	分值	主要变例	累积节点数	分枝因子
1	9	马八进七	97	
2	0	马八进七 马 2 进 3	673	6.9
3	9	马八进七 马 2 进 3 马二进三	2029	3.0
4	0	马八进七 马 2 进 3 马二进三 马 8 进 7	8594	4.2
5	6	马八进七 马 2 进 3 马二进三 马 8 进 7 车九进一	23711	2.8
6	1	马八进七 马 2 进 3 兵七进一 马 8 进 9 马七进六 炮 8 平 5	73628	3.1
7	7	马八进七 马 2 进 3 车九进一 炮 2 平 1 车九平四 车 1 平 2 马二进一	290124	3.9

续表 1

深度	分值	主要变例										累积节点数	分枝因子
8	1	马八进七	马 2 进 3	马二进三	马 8 进 7	兵七进一	车 9 进 1	马七进六	车 9 平 4			999417	3.4
9	6	马八进七	马 2 进 3	马二进三	马 8 进 7	兵七进一	车 9 进 1	车九进一	车 9 平 4	车九平四		2751341	2.8
10	1	马八进七	马 8 进 9	马二进三	车 9 进 1	车九进一	马 2 进 3	车九平四	车 9 平 4	兵七进一	车 1 进 1	11986746	4.4

当程序搜索奇数层时, 先行方总比后行方多走一步, 而当程序搜索偶数层时, 先行方走的着数和后行方一样多, 因此奇数层的搜索分值总是比偶数层略大. 这种现象称为奇偶层效应.

如果博弈程序运用了窗口搜索技术(如期望窗口、PVS、MTD(f)等), 那么搜索效率就会因奇偶层效应而降低. MTD(f)算法的创始人 Plaat 甚至建议, 为克服奇偶层效应的影响, 在迭代加深时把深度步长改成 $2^{[5]}$.

2.2 先行权分值

既然奇偶层效应是由于先行权引起的, 那么可以考虑在局面评价函数上作先行权的调整, 让奇偶层的评价尽可能地接近. 例如, 表 1 的搜索结果, 采用的局面评价函数是:

```
int Evaluate(void) {
```

```
    if (bWhiteTurn) {
        return nWhiteValue-nBlackValue;
    } else {
        return nBlackValue-nWhiteValue;
    }
}
```

为了消除奇偶层效应, 可以在局面评价函数中增加了“先行权分值”:

```
#define ADVANCED 3
int EvaluateAdjust(void) {
    return Evaluate() + ADVANCED;
}
```

使用了调整后的局面评价函数, 各层的分值、分枝因子都会比较接近, 而搜索的节点数则有明显的下降, 可见搜索效率有了充分的提高, 见表 2.

表 2 削弱的奇偶层效应
Table 2 Weakened even-odd effect

深度	分值	主要变例										累积节点数	分枝因子
1	6	马八进七										97	
2	3	马八进七	马 2 进 3									563	5.8
3	6	马八进七	马 2 进 3	马二进三								2109	3.7
4	3	马八进七	马 2 进 3	马二进三	马 8 进 7							7812	3.7
5	6	马八进七	马 2 进 3	车九进一	马 8 进 7	车九平四						18284	2.3
6	3	马八进七	马 2 进 3	车九进一	车 1 进 1	车九平四	车 1 平 4					59725	3.3
7	6	马八进七	马 2 进 3	车九进一	车 1 进 1	车九平四	车 1 平 4	马二进三				190052	3.2
8	3	马八进七	马 2 进 3	车九进一	车 1 进 1	车九平四	车 1 平 4	马二进三	马 8 进 7			596068	3.1
9	3	马八进七	马 2 进 3	车九进一	车 1 进 1	车九平四	车 1 平 4	马二进三	马 8 进 7	车一进一		2099191	3.5
10	3	马八进七	马 2 进 3	车九进一	车 1 进 1	车九平四	车 1 平 4	马二进三	马 8 进 7	车一进一		5782475	2.8

每一着棋都会有一个价值, 使局势朝有利于自己的方向发展, 这个价值的一半就是先行权分值. 然而, 先行权分值并不是始终不变的, 一般情况下, 开局的先行权分值最大, 随着棋局渐渐进入残局, 先行权分值会减小甚至丧失. 中国象棋和国际象棋中还有“无等着”局面(Zugzwang), 此时走任何一着棋都会让自己的局势更糟, 在这种情况下, 先行权分值就相当于负值.

围棋的局势最容易被量化, 布局和中盘阶段可认为先行权分值是 5.5 目(因为先手方贴目 5.5 目), 由此可认定平均每一着棋的价值在 11 目左

右^[6]. 随着棋局进入收官阶段, 先行权分值锐减, 收到单官就是零了.

2.3 空着裁剪

空着裁剪(null-move pruning)的思想是, 假设对手连走 2 着, 如果本方在浅层的搜索中仍旧具有良好表现(产生 Beta 截断), 那么就不需要进行完全深度的搜索了. 写成伪代码, 就是:

```
if (can_do_null()) {
    do_null();
    value=-search(-beta, 1-beta, depth-R-1);
    undo_null();
    if (value >= beta) {
```

```

return value;
}
}

空着裁剪由 Donninger 在 1993 年首次发表以后[7],就一直被绝大多数的国际象棋和中国象棋程序所采用,但也因此带来了其他问题——有些基于 Alpha-Beta 窗口的传统裁剪算法,如无效裁剪(futility pruning)、剃刀裁剪(razoring)、multi-probe-cut 等算法都失效了.看来观察以下的裁剪算法:

#define MARGIN 6
if (can_do_razor()) {
    value = search(beta + MARGIN - 1, beta + MARGIN,
depth-R)-MARGIN;
    if (value >= beta) {
        return value;
    } // Beta 型裁剪,高过边界则直接返回
}

```

这是一个 Beta 型的剃刀裁剪算法,它和前面空着裁剪的伪代码有些相似.根据先前介绍的先行权分值的概念,用了“do_null()”函数以后,就相当对手争取到了 2 倍的先行权分值,因此以上 2 段代码中加粗的部分,有异曲同工之妙.

当然,在搜索到的大多数局面下,先行权分值都会非常大(如果对手正在捉吃一个车,那么先行权分值就会高达一个车的分数,比如说 200 分),所以空着裁剪在大多数情况下都是安全的.

而对于无效裁剪和剃刀裁剪来说,2 倍于先行权分值的边界实在太不安全了,因此实际使用时不得不把边界大幅度提高(比如说从代码中的 6 提高到 100,即一个马或炮的价值).因此无效裁剪和剃刀裁剪跟空着裁剪同时使用时,效果就会被空着裁剪完全淹没.

国际象棋程序 DarkThought 的作者 Heinz 曾经把这 3 种裁剪混合在一起应用,称为 AEL 算法.在这种算法中,无效裁剪和剃刀裁剪都是 Alpha 型的,这才使得它们发挥在空着裁剪所不能管辖的 Alpha 结点上(到目前为止,空着裁剪都是 Beta 型的),在程序中起到非常有限的效果^[8].

```

#define MARGIN 50
if (can_do_razor() && depth == R) {

```

```

value = eval() + MARGIN;
if (value <= alpha) {
    continue;
} // Alpha 型裁剪,低过边界则跳过搜索
}

```

3 结束语

面对由先行权引出的诸多问题,仅仅就事论事是远远不够的.就以六子棋来说,尽管它的初衷是消除五子棋中明显的先行权优势,然而它自诞生之日起就成为一种全新的棋类游戏,其丰富的战略战术思想跟其他棋类迥然不同,也不是一个先行权问题所能解决的.

“先手”在棋类游戏中占有重要的地位,但先手不是目的而是手段,只有擒王(对于象棋来说)和争夺地盘(对于围棋来说)才是目的.所以,本文对先行权的研究只是一个开端,对于博弈理论来说,如何把先行权化为真正的优势,才是需要进一步研究的课题.

参考文献:

[1]黄 晨. ELO 等级分计算公式详解[W]. 2004.
 [2]台湾交通大学资讯工程系. 六子棋首页[W]. 2005.
 [3]Wu IC. Connect 6[J]. ICGA J, 2009, 29(4): 234-241.
 [4]黄 晨. 中国象棋对弈程序 ElephantEye[W]. 2006.
 [5]PLATT A. MTD(f)-A Minimax Algorithm faster than NegaScout[W]. 1997.
 [6]陈志行. 电脑围棋小洞天[M]. 广州:中山大学出版社, 2001.
 [7]DONNINGER C. Null move and deep search: selective search heuristics for obtuse chess programs[J]. ICGA J, 1993, 16(3): 137-143.
 [8]HEINZ E A. AEL pruning[J]. ICGA J, 2000, 23(1): 21-32.

作者简介:



黄 晨,男,1981 年生,硕士研究生,主要研究方向为量子化学计算理论和应用的研究,业余从事中国象棋的计算机博弈理论研究.

E-mail: webmaster @ elephantbase.net