

# 一种五子棋博弈算法的分析

周洋,邓莉,谢煜

(武汉科技大学计算机学院,武汉 430065)

## 摘要:

博弈是用来解决一组决策者之间冲突或合作问题的数学方法。在实现玩家和电脑之间的五子棋对弈时,常常使用博弈方法来确定电脑的走法步骤。经过对五子棋的一种博弈算法设计和实现的分析,总结出五子棋问题求解的算法思路,并分析出算法的性能瓶颈及相应的解决方案。

## 关键词:

极大极小搜索算法; Alpha-beta 剪枝; 博弈; 五子棋

## 0 引言

五子棋是一种深受大家喜爱的棋类游戏,其规则简单,变化多样,为其增加了更多的趣味性。五子棋起源于中国古代棋类,而对其发展起重大作用的是日本。18 世纪五子棋在日本流行起来,后来发现五子棋有执黑先行必胜定律,为了让其能更公平,在比赛中添加了许多禁手规则,而在业余人士的游戏中则一般不设置这些规则。五子棋发展至今禁手规则已逐渐削弱了执黑先行的优势。在 1997 年计算机就下赢了当时的国际象棋冠军。而对五子棋一直以来都没有发展出一种可以绝对取胜算法。原因就在于五子棋(15×15)的搜索空间比国际象棋(8×8)更大。想在有限的时间找出一种必胜的策略显然是件非常困难的事。博弈是在一定条件下,遵守一定的规则,一个或几个拥有绝对理性思维的人或团队,从各自允许选择的行为或策略进行选择并加以实施,并从中各自取得相应结果或收益的过程。博弈最终是要取得一种均衡解,即博弈的参与者对其收益达到一种较为满意的程度。所以将博弈应用于五子棋算法的研究可以有效地提高算法的效率。

五子棋的对弈双方只能是黑子或者白子,而执黑先行者一般都会选择在棋盘中心,所以五子棋的开局往往比较相似,常见的有 26 种开局。如下:

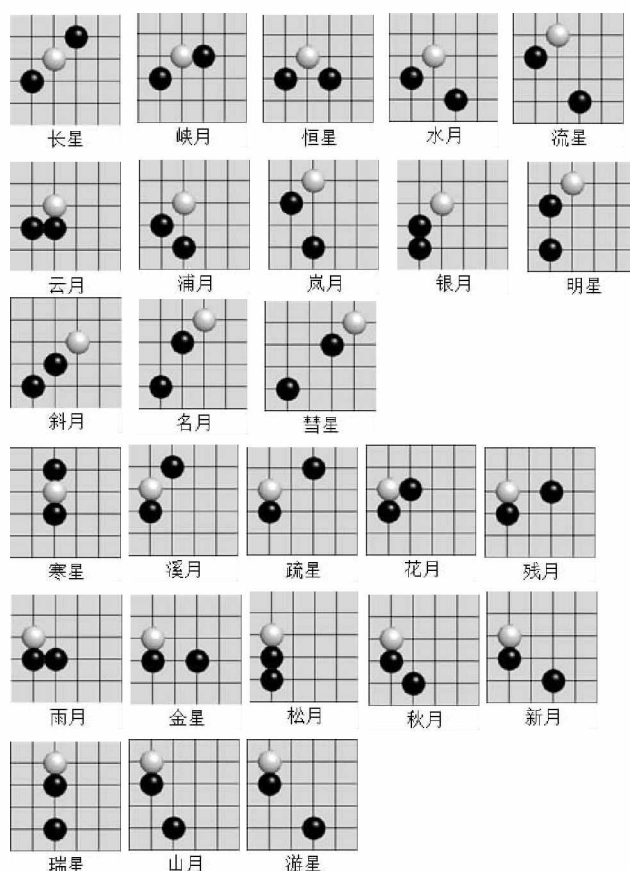


图 1

而在这次将博弈的思想应用于五子棋用到了极大极小的搜索算法和 $\alpha$ - $\beta$ 剪枝算法。运用这两个算法和已经确定的分数表来产生出当前节点的评估分值,计算机选择一个对自己最有利的点落子。

## 1 主要算法

### 1.1 极大极小搜索算法

一般在一个五子棋对弈中如果采用暴力搜索得到最终结果,则会使搜索树的深度非常大,尽管计算机的计算性能在不断加快,但还是不能满足需要,主要是因为耗时太长。所以一般是规定一个搜索深度,然后在这个深度里进行深度优先搜索。而此处设计的极大极小搜索算法就是一个在有限深度上找出一个当前搜索深度对自己最有利的节点,从而做出一个最优的选择。由此也可以看出搜索深度越深评估出的值更接近真实值。

举个例子来说明这个算法:假设A和B对弈,当前到A落子,此时我们会遍历A的每一个落子点,而对于A的每一个具体的落子点,遍历B的每一个落子点,然后再遍历A的每一个落子点(如图2),如此搜索下去,直到达到搜索深度。到此就遍历了A的所有节点。

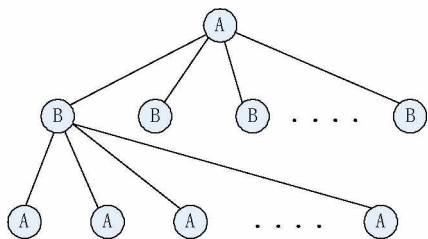


图2

但是并没有结束,我们得选出一个最有利的点落子,在刚刚的遍历中我们已经评估出每一个节点分数值,如图2所示。在搜索树中,A落子的点为极大值点,B落子的点为极小值点。这是因为A落子时肯定会选出对自己最有利的点下棋,而B会选择一个当前局面评分最小的点来落子。这样做的目的就是为了在有限深度的搜索里找出最佳的走法。

### 1.2 Alpha-Beta 剪枝

使用alpha-beta剪枝算法可以剪去一些不必要的分支,从而提高搜索效率,是一种优化博弈树的搜索算法。

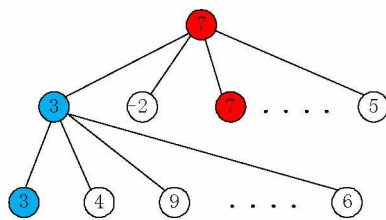


图3

将alpha-beta剪枝算法与极大极小搜索算法结合起来,可以有效改善决策搜索的性能。假设A和B对弈,当前已通过极大极小搜索算法遍历出部分节点的值,假设如图3所示,当前对第二行第二个节点的深度搜索中出现了-2这个值,这就表明在其之后的节点中即使有比-2大的值,第二行第二个节点的值都不可能超过-2,由于第二行以一个节点的值已经为7,故可将-2这个节点这条路剪去,从而有效的节约了计算机资源,提高效率。对于 $\alpha$ - $\beta$ 剪枝算法来说最主要的影响是排列顺序,理想排列顺序下的剪枝效率和最差情况下的剪枝效率相差极大。

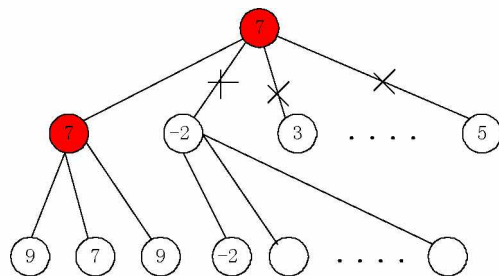


图4

## 2 性能分析

在这个五子棋游戏的设计中,除了以上两个算法外最重要的就是评分表的设定,一个好的评分表对结果的影响也是显而易见的,在这里由于很多人已经做过相应的研究,有很多比较好的评分表,所以这里可以借鉴那些经过时间检验过的评分表。不同搜索深度和搜索广度对算法的效率影响较大,在这里以搜索深度为三(由于实际并不用遍历所有节点,此处设置了搜索的广度为七)和搜索深度为五的比较为例进行算法的性能分析。理论上,深度为五和深度为三的算法的复杂度分别为 $O(n^3)$ 和 $O(n^5)$ ,但是经过alpha-beta剪枝,算法的执行时间要远远低于这个值。我们分别实测了两

种不同深度下算法的执行时间, 每组分别重复执行了五次, 测试结果如表 1 和表 2 所示。

以上数据表明,  $\text{depth} = 5$  的耗时大概为  $\text{depth} = 3$  的 4 到 9 倍, 比理论上预估的倍数 49 要小很多, 由此可以说明 alpha-beta 剪枝算法有效缩小求解搜索空间的大小, 大幅度降低计算复杂度, 从而缩短了执行时间。

### 3 结语

一个好的算法可以有效地提高计算效率, 节约时

间。将博弈的思想应用于五子棋问题的求解也正是出于此目的, 由以上的分析可知五子棋博弈对搜索算法的效率有很高的要求。在算法中融入博弈的思想, 力求在更快的时间里找出一个均衡解, 本文证实这里使用的极大极小搜索算法与 alpha-beta 剪枝算法结合是一种高效的评估算法, 由此可以更快地找出相对较好的解决方案。

表 1 深度为 3 时五子棋每走一步的时间

测试次数	每步的时间 (ms)									
	1	2	3	4	5	6	7	8	9	10
1	52	34	36	23	31	27	27	36	35	0
2	50	43	36	24	27	29	47	36	51	56
3	54	29	31	19	37	35	33	37	0	47
4	52	33	28	28	28	22	0	0	33	0
5	50	38	47	32	31	38	53	48	69	50

表 2 深度为 5 时五子棋每走一步的时间

测试次数	每步的时间 (ms)									
	1	2	3	4	5	6	7	8	9	10
1	265	319	260	246	298	226	275	328	235	108
2	284	241	166	146	124	154	377	381	408	305
3	260	284	202	164	170	159	125	169	119	223
4	285	259	167	175	110	161	104	164	157	162
5	270	233	178	301	181	295	158	150	218	243

#### 参考文献:

- [1]严蔚敏, 吴伟民. 数据结构(C语言版)[M]. 北京: 清华大学出版社, 2011.
- [2]王晓东. 计算机算法设计与分析(第4版)[M]. 北京: 电子工业出版社, 2012.
- [3]A.Levitin 著. 算法设计与分析基础(第3版)[M]. 潘彦译. 北京: 清华大学出版社, 2015.
- [4]B.Eckel 著. Java 编程思想(第4版)[M]. 陈昊鹏译. 北京: 机械工业出版社, 2007.

#### 作者简介:

邓莉(1972-), 女, 湖北钟祥人, 博士, 研究方向为云计算、分布式计算

周洋(1994-), 男, 湖北黄冈人, 本科生

谢煜(1993-), 男, 湖北仙桃人, 本科生

收稿日期: 2017-01-12

修稿日期: 2017-03-28

## Analysis of a Game Playing Algorithm for Five-in-a-Row

ZHOU Yang, DENG Li, XIE Yu

(College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065)

#### Abstract:

Game playing is a mathematical method to tackle conflict or cooperation between several decision-makers. Game playing is often used to find an optimal solution to five-in-a-row. By analyzing the design and implementation of a game playing algorithm for five-in-a-row, gives thorough design thoughts, finds performance bottleneck, and also presents according solution.

#### Keywords:

Max-Min Search; Alpha-Beta Pruning; Game Theory; Five-in-a-Row