ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.15, No.33, November. 2019
E-mail：eduf@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

机器博弈主要技术分析——以六子棋为例

何轩,洪迎伟,王开译,彭耶萍

(吉首大学,湖南 吉首 42700)

摘要:该文针对机器博弈中常见的技术和各种优化方法以六子棋为例进行分析和讨论,从棋盘表示、走法生成、博弈树与搜索算法这三个方面进行展开,从各种技术的优缺点出发,为机器博弈新思路提供了参考。

关键词:机器博弈;六子棋;博弈树;搜索算法;蒙特卡罗树;剪枝

中图分类号:TP391 **文献标识码**:A

文章编号:1009-3044(2019)33-0172-02
DOI:10.14004/j.cnki.ckt.2019.3964

开放科学(资源服务)标识码(OSID):

机器博弈是人工智能领域最富挑战性的项目之一,而六子棋作为一种典型的博弈类竞技游戏,相比五子棋黑棋先手必胜的单调不公平性,其公平性到目前为止还不能被证伪,其状态空间大小(约为10^{172})为五子棋(约为10^{105})的10^{72}倍,搜索结点数大大增加,极具挑战性。因此,以六子棋作为研究机器博弈的切入点既能促进六子棋的发展,同时也可推动机器博弈乃至人工智能领域的进步。

六子棋的棋盘大小为19行19列,其规则为:黑棋先手一子,以后白黑轮流落二子,在纵、横、斜任意一条直线上先连成六子(或六子以上)者获胜。如下满棋盘仍未分出输赢则判为平局。

1 棋盘数据结构

为使用计算机分析六子棋问题,首要任务是将棋盘存储到计算机中。主要有三种常见的表示方式:一维数组,二维数组以及位棋盘。

1.1 二维数组

一个19×19的棋盘可以用一个二维数组board[19][19]来表示,二维数组中的每一个元素都映射棋盘中的一个位置。例如board[0][0]表示棋盘中的第0行第0列的交叉点,优点:表示清晰,容易实现。

1.2 一维数组

将二维的棋盘存放在一维数组中需要建立一个二维与一维之间的相互映射关系(pos(x,y):棋盘中第x行第y列的交叉点,$board_1[i]$:一维数组)

$$pos(x,y) \rightarrow board_1[19x+y] \quad ①$$

由①可得②:

棋盘中坐标(x,y)与数组下标i的映射关系:

$$\begin{cases} x = 19 / i \\ y = 19 \% i \end{cases} \quad ②$$

占用空间同二维数组,但如能利用某种方法提取待搜索的特征信息作为模式串,便可借助模式匹配算法,把递归搜索转换成线性匹配,提高效率。

1.3 位棋盘(Bit Boards)

为提高运算速度、减少存储空间,常采用位棋盘的方式来表示数据,在六子棋中,棋盘上只可能出现:空、黑子、白子这三种情况,使用二进制表示至少需要2位。棋盘上每行共19个点,需要至少38位,考虑使用一个长整型(占8个字节共64位)则整个棋盘需要19*8B,共152B。利用位进行存储,可以使用逻辑运算来处理问题,其效率远高于其他运算。

2 走法生成

走法生成即将下一步落子的所有合法位置枚举出来。枚举走法的过程往往依赖搜索,因此一个好的走法生成策略是博弈系统效率提升的关键。常见的走法生成策略有:预置表、棋盘扫描、Null-Move启发(空着启发)以及它们的结合等。预置表:存储所有可行的走法,生成走法时直接查表,优点即速度快,缺点即局限于规则较多且走法有限的棋类。棋盘扫描:即按照规则对棋盘区域进行遍历,确定落子位置。Null-Move启发(空着启发):假设一方先不动,让另一方多落子一轮,然后搜索动方获胜的迫著序列,此时对于不动方来说只需重点防范这个序列,产生的防范区域称为R-Zone,不动方便只需在R-Zone中生成走法,大大减少了搜索空间,提高了搜索效率,缺点即工于防范而疏于进攻,难以获胜乃至防不胜防而落败,因此空着启发常常与其他算法结合使用。

3 博弈树与搜索算法

博弈树即对博弈局面及其未来可能性的抽象。就六子棋而言,博弈双方轮流交替回合,这种情况能够抽象成一颗与或树。“与”指我方需要考虑所有的走法,走法之间是与的关系,或表示对方可能选择众多走法中的任意一种。其中交替表现在树中偶数层结点为我方(黑子),奇数层为对方。

收稿日期:2019-09-25

作者简介:何轩(1999—),男,吉首大学软件工程专业本科在读;洪迎伟(2001—),男,吉首大学软件工程专业本科在读;王开译(1999—),男,吉首大学软件工程专业本科在读;彭耶萍(1981—),女,主要研究方向为数据挖掘及算法。

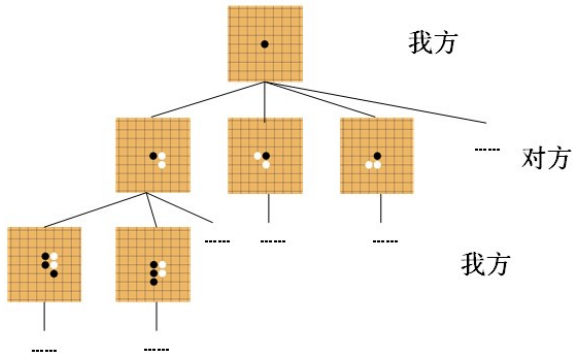

图1　六子棋博弈树

在建立博弈树的过程中评估函数需要对每个结点评估。所谓评分就是给当前棋局打分，对我方越有利分数越高，反之分数越低。搜索则是遍历博弈树的过程，目前来说，几乎所有的搜索算法都基于极大极小值思想。

3.1 极大极小值算法

在与或博弈树的基础上，假设一方为我方，则在搜索的过程中我方回合时总是选择评分最高的结点，而轮到对方时，总是选择评分最低的结点。

3.2 Alpha-Beta 剪枝

极大极小值算法在搜索的过程需要完整遍历博弈树的每个结点，然而有很多结点不会对局面产生贡献，如果能够去掉对这些无用结点的遍历，算法的效率就能够得到极大的提升，这也是一系列改进算法的着手点。对于Alpha-Beta算法来说，它通过改变搜索的上下界来缩小搜索空间，即所谓的剪枝。

3.3 MCTS树搜索

大名鼎鼎的AlphaGo Zero也是基于该算法，MCTS即蒙特卡罗树搜索。虽然同样通过剪枝来缩小搜索空间，与Alpha-Beta不同的是，它对结点的评判依据不是人为构造的评估函数，而是蒙特卡罗模拟，所以随着搜索深度增加，越接近最优解，收敛较快。尤其适合应用于分支较多的搜索问题。该搜索算法主要有四个部分：选择、扩展、模拟以及回溯更新。

选择阶段，从起点开始递归地对评价最高的结点进行搜索，评价一般采用UCT策略，该策略在搜索时采用UCB算法(Upper Confidence Bounds 置信区间上界)其思想是：先对起点所有的子结点都搜索一遍，按照公式③计算每个结点的分数，然后选择分数高的继续搜索。

$$Score_s = \frac{Q(s)}{N(s)} + c\sqrt{\frac{2\ln N(p)}{N(s)}} \quad ③$$

s指当前结点，p表示父结点$Score_s$表示当前结点的分数，$Q(s)$表示当前结点的累计分数，N表示结点的访问次数，c是一个自定常数。

扩展阶段，选中一个结点对其进行扩展子节点。

模拟阶段，根据蒙特卡罗模拟方法对拓展出的子节点进行模拟，即随机选择一个可落子位置作为其子节点，然后子节点继续模拟，直到博弈结束。

更新阶段，将子结点的得分累加到其父节点，不断从下向上累加更新。

4 结束语

限于版面以及评估函数的差异性，本文没有详细讨论评估函数部分。机器博弈所涉及的领域较多，作者仅以所了解进行讨论，希望能给读者一点收获。

参考文献：

[1] 林云川.基于深度学习和蒙特卡洛树搜索的围棋博弈研究[D].哈尔滨工业大学,2018.
[2] Justesen N,Mahlmann T,Risi S,et al.Playing Multi-Action Adversarial Games: Online Evolutionary Planning versus Tree Search[J].IEEE,2017.
[3] 曹一鸣.基于蒙特卡罗树搜索的计算机扑克程序[D].北京邮电大学,2014.
[4] 刘雅靖.计算机博弈之六子棋的主要技术分析[J].电脑知识与技术,2011,7(10):2310-2312.
[5] 闵文杰.六子棋计算机博弈关键技术研究[D].重庆交通大学,2010.
[6] 安涌.六子棋机器博弈研究与开发[D].沈阳航空工业学院,2008.

【通联编辑：朱宝贵】