

六子棋

——人工智能系统的设计与研究

周菁菁

(兰州资源环境职业技术学院, 甘肃 兰州 730021)

[内容提要] 本系统是对六子棋计算机博弈技术的一次尝试。此系统主要通过当前局面的所有的棋型进行了静态估值, 然后再采用优化 Alpha-Bate 搜索技术实现了 AI。最后, 本文探讨一些提高电脑智能的方法。并相信随着科技的发展, 人工智能对我们的帮助将会越来越显著, 越来越大。

[关键词] 计算机博弈; 六子棋; Java; Alpha-Bate 搜索技术; 人工智能

[中图分类号] TP311.1 **[文献标识码]** A **[文章编号]** 1008-7427 (2011) 10-0159-01

一、引言

人工智能, 英文缩写: AI。是研究、开发、用于模拟、延伸以及扩展人的智能的理论、方法、技术、应用系统地一门新兴技术科学。

计算机博弈则被认为是被认为在人工智能领域最具挑战性的研究方向之一, 其涉及了人工智能中的推理技术、搜索方法以及决策规划等等, 计算机博弈的研究为人工智能带来了许多重要的方法与理论, 产生了广泛的社会影响和学术影响以及大量的研究成果。六子棋作为最近几年才兴起的棋类游戏, 对其计算机博弈技术和算法的研究相对比较少。所以基于 Java 的智能六子棋系统的实现是对计算机博弈问题的一次探索研究。

二、系统的总体设计

1. 在界面方面: 应该简单化, 做到实用化、方便化, 这样以来可以满足不同层次的使用者的需要了。
2. 在实现方面: 对六子棋其数据结构, 棋子触发, 搜索算法, 评估函数等进行了设计和实现的过程。
3. 在系统规范方面: 删除了不必要的冗余, 实现了程序代码的标准化, 以及软件的统一化, 确保此软件的可扩展性, 可维护性以及其实用性。

三、主要技术

第一, 是界面设计及事件的响应方面, 系统界面包括了棋盘部分, 用户面板部分与游戏信息及选项部分, 当前状态部分和系统选项这六个部分。其中为了系统的实现的方便, 棋盘的背景与棋盘的触发面板是分开发设计地, 也就是在主框架中增加了背景, 并且放置了用户面板部分, 当前状态部分以及系统选项的部分, 再就是单独设计了棋盘触发面板, 其覆盖于棋盘的背景之上。至于棋子信息的部分, 就隐藏在了棋盘触发面板上了。同时, 为了方便触发, 当前状态部分、系统选项部分、游戏选项部分都是直接得嵌入到了主框架中了。

第二, 就是评估函数的设计, 从理论上分析, 评估函数就是对棋局的综合评估, 换句话说, 是给当前局面黑白棋分别的评分, 从而可以确定当前的形势, 为搜索引擎提供了评判标准。该函数的好坏与否直接决定了解题能力的强与弱, 所以只有好的评估函数才能对结果予以保证, 否则, 搜索所得到的结果就有可能与期望相距很大。就像是一个好的棋手通常都能够对当前棋局做出判断, 以最好的方式进行进攻, 并控制整盘棋局的发展方向。由此可见, 仅凭几个简单的评估函数是不可能很好的抽象这整套评价模式的, 因为太多的因素要考虑到。

第三, 是走法生成(告诉其他部分下一步往哪里走的模块)方面, 每种棋类的规则都不同, 走法生成的复杂程度也是大相径庭。举个例子, 在一种棋类游戏中, 棋子种类越多, 棋子的走法与规则也就越多, 要是在程序中实现的话就越复杂。

在六子棋地对弈程序当中, 黑白双方交换下子, 其中, 没有吃子、提子、等规则的存在, 哪一方在棋盘的水平、垂直、斜线方向上首先形成了连续的六颗棋子, 就获胜了。因此, 六子棋规则的方法生成来说, 棋盘上的空白位置就是合法的位置。不过在现实中, 有些位置是不该首先考虑的, 比

如说对局时是不会把棋子直接放在下边界上。在这里, 有一点需要注意的, 那就是我们当下的棋子的位置与对方下的棋子的位置是有很大关联的, 也就是说我们当下的棋子有可能是用来限制对方的棋子, 并阻止其连成六颗, 亦或是躲开对方的棋子, 让自己的棋子连成六颗, 这样, 较优的走法生成的位置就应该考虑在上一次对方下子的周围。

第四是搜索技术方面, 先来了解一下博弈树。

基本上所有的棋类问题, 都可以用博弈树来描述的。博弈树, 是把计算机和用户所有可能的走法和局面一一罗列出来的一颗树。黑白双方交替的按相对合理走法把树展开, 树的每个节点都表示一个局面。根节点是当前需要进行计算的局面, 中间节点是对弈过程中的某个局面, 最下面的节点是推导出的局面。整个博弈树是从当前局面出发的, 包含所有可能的对弈过程的搜索树。所以六子棋博弈问题也就转化成了寻求最佳路径的问题了。假设博弈树是很有限的, 这样, 就不会碰到永无止境的棋局或一步有无限种算法的棋局了。

举个例子, 如果有两个玩家对弈, 他们分别为 max 和 min。让 max 先走, 然后两人交替走, 一直到游戏结束。我们用产生式系统来进行描述。由于不可能对一个完整解图进行搜索, 我们需要定义一个静态估计函数 g, 以便对游戏状态进行估值。在此我们假设有利于 max 的状态取 $g(a) > 0$, 并且有利于 min 的状态取 $g(a) < 0$ 。这样得出以下搜索的五个步骤。

1. 首先生成了整个博弈树(扩展树的每个节点)。
2. 用函数 g 对每个子节点进行估值, 得终节点的评价值。
3. 使用终节点的估值 g, 得到其搜索树上一层节点的估值。
4. 重复三过程在 max 层取其分支的最大值, min 层取其分支的最小值, 一直回溯到根结点。
5. 根结点选择分支值最大的走走走。

四、设计进的方案

目前有以下四种方法可以提高电脑的棋力:

第一, 动态估值。主要在于攻守的转换和急手的处理, 这个要综合很多方面的因素, 并且要从以往的经验为基础多加尝试, 但是工作量很大并且不易拿捏。

第二, 增加细致的特定棋形的判断。增加棋型判断, 程序中的判断不是很全面, 与实际存在一些差异, 不过, 这是个数学问题。

第三, 将局部最优和全局最优结合起来。这种方法是更容易实现的, 就是增加搜索深度, 然后考虑整体以及局部的估值, 用递归可以实现。

第四, 完善开局库和增加学习功能。在对局中十分有用, 带有完善开局库和有学习功能的六子棋将是难以战胜的, 不过, 这是一个十分复杂的过程, 其中, 完善开局库的学习功能比较容易实现。

[参考文献]

- [1] 李果. 六子棋计算机博弈及其系统的研究与实现[D]. 重庆大学, 2007.
- [2] 王小春. PC 游戏编程(人机博弈)[M]. 重庆大学出版社, 2003.