ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.5,No.25, September 2009, pp.7198-7200

E-mail: eduf@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

六子棋智能博弈系统的研究与实现

黄继平,张栋,苗华

(重庆理工大学 计算机科学与技术学院,重庆 400054)

摘要:六子棋作为一个新兴的游戏,已在棋类计算机智能博弈领域占有重要的位置,其为保证竞赛公平性而采取的一步两子的规则对计算机博弈的效率是很大的考验。为此,在实现六子棋博弈系统状态表示、搜索策略和评估函数几大核心框架的基础上,提出"路"的概念,简化评估函数类型,有效提高博弈性能。

关键词:人工智能;博弈树;六子棋;

中图分类号:TP181 **文献标识码**:A **文章编号**:1009-3044(2009)25-7198-03

The Research and Implementation of Connect6 Intelligent Chess Game System

HUANG Ji-Ping, ZHANG Dong, MIAO Hua

(School of Computer Science and Engineering, Chongqing University of Technology, Chongqing 4000504, China)

Abstract: As a new game, Connect6 has held an important position in the field of Computer chess game intelligence. It´s a big problem to the efficiency of Computer Game, because Connect6 puts two chessman in one step to ensure that the game is fair. So, based on the implementation of main components of Connect6 game: description of state, search engine and evaluation function, this paper makes a new conception: "road", which simplifies the evaluation functions and effectively improve game performance.

Key words: artificial intelligence; game tree; connect6; genetic algorithms

在人工智能领域始终将棋类的机器博弈作为常用的研究平台之一[1]。以棋类游戏为研究和验证平台,各种搜索算法、智能方法在计算机博弈中都可以得到广泛的应用。1997 年,"深蓝"小组研究开发出"更深的蓝",以 3.5 比 2.5 击败了国际级大师卡斯帕罗夫,成为人工智能发展的一个里程碑。至今为止,计算机博弈在五子棋、象棋、国际象棋甚至围棋上都取得了卓越的成就。

1 六子棋简介

六子棋(Connect6)是由台湾交通大学吴毅成教授所发明的一种新游戏,由五子棋改良而来,相比较而言,它具有规则简单、变化复杂、游戏公平三个很好的特性。在六子棋里,除了持黑的第一手下一子外,黑白双方轮流各下两子,最后连成六子者胜。由于各方每次下完一手后,盘面都比对方多一子,因此赛局可自然达成平衡的状态,使得公平性大为提升。而五子棋、象棋等,先手常常具有一些优势。同时由于一次两子,组合变化莫测,其复杂度已被评估为仅次于围棋和日本象棋,远高于五子棋,与象棋相当或略高。

六子棋计算机博弈可以分解为四个核心部分:状态表示、走法生成、搜索引擎、评估函数。其中状态表示和走法生成是所有棋类游戏的基础,而搜索引擎很评估函数是使六子棋具有"思考"能力的关键。

2 状态表示

要实现计算机模拟六子棋的博弈过程,就需要对之的博弈状态进行分解描述。本文将六子棋机器博弈进程分别表示为三种离散数据结构:棋局状态,棋子状态和棋形描述。由于六子棋源于五子棋,在棋局和棋子状态的描述上和五子棋及围棋类似,所以下文主要针对六子棋的棋形进行阐述。

2.1 棋形描述

棋形就是棋局的一种状态描述,是描述棋盘上不同棋子的分布状态。采用状态矩阵可以比较清楚地描述某个时刻,机器博弈进程中棋局状态和棋子状态的变化,如何有意识地控制这个变化发展方向,并使得朝有利于己方赢棋的方向发展,是机器博弈进程中,计算机需要解决的核心问题。

本文采用了其中 15 种棋形:连六(获胜)、长连(获胜)、活五、眠五、死五、活四、眠四、死四、活三、眠三、朦胧三、死三、活二、眠二、死二,并将这些棋形的描述放入决策支持系统的知识库,形成了计算机博弈决策系统推理的基础,下面对其中重要的改进解释如下:

① 六连,C6:在棋盘的纵向、横向或斜向的任意一条线上,形成的连续相连的 6 颗同色棋子。
② 活五,C5:在同一直线上的 5 颗同色棋子,符合"对方必须用两手棋才能"的条件。
③ 活四,C4:在同一直线上的 4 颗同色棋子,符合"对方必须用两手棋才能挡住"的条件。
④ 眠四,S4:在同一直线上的 4 颗同色棋子,符合"对方用一手棋就能挡住"的条件。
⑤ 死四,D4:在同一直线上的 4 颗同色棋子,它们已无法形成六连或长连。
⑥ 朦三,S3:在同一直线上的 3 颗同色棋子,符合"再下一手棋只能形成眠四,但如果再下两手棋的话就能形成活五"的条件。

收稿日期:2009-05-25
基金项目:人工生命体行为与行为选择的进化研究(CSTC,2007BB2415)
作者简介:黄继平(1979-),男,重庆人,助教,硕士,主要从事管理信息系统、人工智能领域的研究。

2.2 “路”的思想

事实上，六子棋中每一种棋形都有很多的表现形式。同时如果有几种棋形交叉时判断起来就更加复杂。如何判断棋形就成为计算机系统难点，如果模板设置错误或棋形统计不完全，将严重的影响博弈系统的棋力。因此本文提出“路”的概念。所谓“路”就是在棋盘上连续 6 个点能够连成一条直线，则称为 1“路”。六子棋棋盘是 19×19 的棋盘，计算可得一共有 924“路”。这样每条“路”上就只有 6 个点，这样就使得棋形的判断和统计非常简单了。例如，在某“路”里已经存在 4 颗棋子，此时就无需去判定它到底是活四、眠四，还是眠四棋形，从而极大减少了计算量。

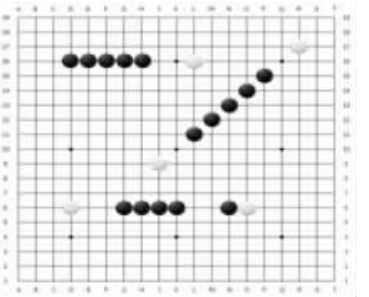

图 1 活五

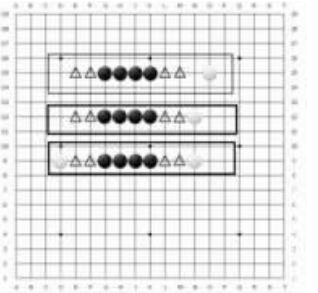

图 2 活四

当博弈开始并下棋子时，用两个哈稀表来分别存储棋盘上已下棋子和待搜索点的信息，对于有棋子的用哈稀表中的键值 key，分别表示棋盘上棋子的坐标和棋子对象，这样特定的 key 就对应一个 value 值。对于待搜索点(Evealutepoint)，将最外层的已下棋子的点(UsedPoint)向外扩展 5 个点，从而所得到未下棋子点的集合，并存放到一个哈希表中。这样的表示就可以比较方便地找到棋盘上面的棋子状态，实际上只要遍历哈稀表，而不是遍历整个棋盘，从而极大地节约时间。

3 搜索策略

在二人博弈问题中，为了从众多可供选择的行动方案中选出一个对自己最为有利的行动方案，需要对当前的情况以及将要发生的情况进行分析，通过搜索算法从中选出最优的着法。在博弈问题中，每个棋局可供选择的行动方案有很多。因此，将生成十分庞大的博弈树，如果试图通过直到终局的与或树搜索而得到最好的一步棋，这是不可能实现的。例如，30 步的六子棋完整的博弈树，可以计算出大约有 10140 个节点，假设每个博弈树枝的长度为 30，大约有 335 个判断分支点，那么从起点到终点大约有 2335 条路径，即使按照每条路径耗时 1/10000 秒，也大约需要耗时 2335/214≌2321(秒) 2298(年)。显然，要遍历所有分枝、接点，目前的任何计算机都无法完成这个搜索任务。因此，必须寻求合适搜索算法，以完成该项搜索任务。下面将介绍一些这样的搜索算法。

3.1 极大极小值搜索算法

极小极大搜索法是最常使用的搜索方法，其基本思想是：

1) 设博弈的双方中一方为 MAX，另一方为 MIN。然后为其中的一方(例如 MAX)寻找一个最优行动方案。

2) 为了找到当前的最优行动方案，需要对各个可能的方案所产生的后果进行比较，具体地说，就是要考虑每一方案实施后对方可能采取的所有行动，并计算可能的得分。

3) 为计算得分，需要根据问题的特性信息定义一个估价函数，用来估算当前博弈树端节点的得分。此时估算出来的得分称为静态估值。

4) 当端节点的估值计算出来后，再推算出父节点的得分，推算的方法是：对“或”节点，选其子节点中一个最大的得分作为父节点的得分，这是为了使自己在可供选择的方案中选一个对自己最有利的方案；对“与”节点，选其子节点中一个最小的得分作为父节点的得分，这是为了立足于最坏的情况。这样计算出的父节点的得分称为倒推值。

5) 如果一个行动方案能获得较大的倒推值，则它就是当前最好的行动方案。

在博弈问题中，面对庞大的博弈树，试图利用完整的博弈树来进行极小极大分析是困难的。可行的办法是只生成一定深度的博弈树，然后进行极小极大搜索，找出当前最好的行动方案。在此之后，再在已选定的分支上扩展一定深度，再选最好的行动方案。如此进行下去，直到取得胜败结果为止，至于每次生成博弈树的深度，当然是越大越好，但由于受到计算机存储空间的限制，只好根据实际情况而定。

3.2 Alpha—Beta 算法

Alpha-Beta 剪枝搜索是一种基于剪枝(Alpha-Beta cut-off)的深度优先搜索(depth-first search)。将走棋方定为 MAX 方，因为它选择着法时总是对其子节点的评估值取极大值，即选择对自己最为有利的着法；将应对方定为 MIN 方，因为它走棋时需要对其子节点的评估值取极小值，即选择对走棋方最为不利的、最有钳制作用的着法。

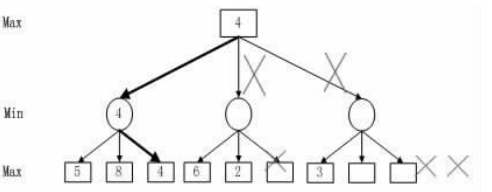

图 3 Alpha 剪枝演示

Alpha 剪枝：在对博弈树采取深度优先的搜索策略时，从左路分枝的叶节点倒推得到某一层 MAX 节点的值，可表示到此为止得以“落实”的着法最佳值，记为 Alpha。显然此 Alpha 值可作为 MAX 方着法指标的下界。在搜索此 MAX 节点的其它子节点，即探讨另一着法时，如果发现一个回合(2 步棋)之后评估值变差，即孙节点评估值低于下界 Alpha 值，则便可以剪掉此枝(以该子节点为根的子树)，即不再考虑此“软着”的延伸。此类剪枝称为 Alpha 剪枝。下图给出了搜索和剪枝过程，最后得到如粗箭头所示的最佳路径片断。

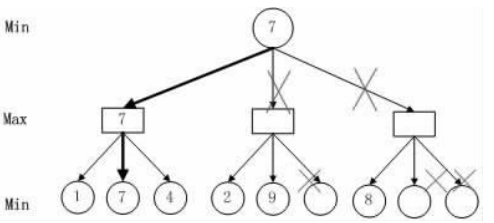

图 4 beta 剪枝演示

Beta 剪枝：同理，由左路分枝的叶节点倒推得到某一层 MIN 节点的值，可表示到此为止对方着法的钳制值，记为 Beta。显然此 Beta 值可作为 MAX 方可能实现着法指标的上界。在搜索该 MIN 节点的其它子节点，即探讨另外着法时，如果发现一个回合之后钳制局面减弱，即孙节点评估值高于上界 Beta 值，则便可以剪掉此枝，即不再考虑此“软着”的延伸。此类剪枝称为 Beta 剪枝。下图给出了搜索和剪枝过程，最后得到如粗箭头所示的最佳路径片断。

Alpha-Beta 剪枝算法的效率与子节点扩展的先后顺序相关。为了得到最好的节点扩展顺序,许多搜索算法在着法(节点扩展的分枝)排序上给予特别的关注。比如在着法生成(节点扩展)时,先生成吃子着法,尤其先生成吃分值高的"大子"着法,因为由此产生着法更有可能是最佳的.围绕着法排序,已经出现许多优秀的搜索算法与举措。如:同形表法、吃子走法的 SEE 排序、杀手走法、未吃子走法的历史启发排序、类比法等。当然这些只适用于象棋类型的多种棋子的游戏。

Alpha-Beta 剪枝算法伪代码描述如下:

```
int AlphaBeta(int depth, int alpha, int beta) {
  if (depth == 0) {
    return Evaluate(); }
  GenerateLegalMoves(); //产生所有的可能走法
  while (MovesLeft()) {
    MakeNextMove();
    val = -AlphaBeta(depth - 1, -beta, -alpha);
    UnmakeMove();
    if (val >= beta)
     {return beta; }
    if (val > alpha)
{ alpha = val; }}
  return alpha; }
```

4 确定评估值

六子棋的棋形估值涉及的因素多,是一个非线性规划问题。例如,活三、活四、眠三、眠四等,通过统计方法、经验值来比较,尽管符合大多数棋类游戏的估值特点,但是将会产生新问题,由于棋形的变化多种多样,很多时候因为棋形统计不全而导致估值有错,进而输掉了比赛。对棋局局面判断、估值主要就是通过统计方法和搜索棋形来完成的,是预先给每种棋形设定一个经验值,经过统计得到如果在某个位置行棋落子后,棋局局面的估值将产生什么变化来决定的。因此,此时的估值与搜索是相互影响和相互促进的,好的估值能够促进有效的搜索剪枝,这样能大大减少需要搜索的点,节约搜索时间。而且,搜索时能够有好的搜索顺序,结合估值的准确性,能最大程度的减少搜索点,在有效的时间范围内,最后确定的要走的点就更加准确。但是,估值与搜索是非常难于协调配合的,实际效果并不如意。

针对六子棋机器博弈的特点,本文采用"路"和"迫着"思想,将 19×19 的棋盘划分为 924"路",根据"路"来估值,计算出棋盘的状态值。这样就只需要计算 924"路"中的棋子颗数即可,根据棋子的颗数给出估值,从而使得估值范围大大缩小,就能明显地提高计算速度。新系统的估值不再进行棋形的判定,仅仅计算"路"中已有的同色棋子的颗数,然后找到"路"相对应的值。假设在某点走子后,计算棋盘局面的状态变化值,通过对比搜索出最优值所对应的行棋走子组合——"着法"。在本文博弈系统的估值中,需要注意两点:一是如果同一"路"中存在不同色的棋子,则该"路"就失去了利用价值或威胁,需要抛弃;二是在估值前,假设在某点走子时,某条"路"出现了 6 个同色的棋子,则胜败已分,系统将自动结束比赛。

5 结束语

本文介绍了实现六子棋智能博弈系统的几个核心部分:状态表示、搜索算法和估值函数。Alpha-Beta 剪枝算法提升了系统搜索的效率,结合针对六子棋特点设计的估值函数,使机器具有较强的"思考"能力。而本文提出的"路"的思想又使得估值函数类型大大减少,进一步提高搜索效率,对提升六子棋的棋力具有较好的作用。

参考文献:

[1] 徐心和,王骄.中国象棋计算机博弈关键技术分析[J].小型微型计算机系统,2006,27(6):961-969.
[2] 王小春.PC 游戏编程[M].重庆:重庆大学出版社, 2002:1-27.
[3] 蔡自兴,徐光.人工智能及其应用[M].北京:清华大学出版社,2005.
[4] 王永庆.人工智能原理与方法[M].西安:西安交通大学出版社, 2000.
[5] 瞿锡泉,白振兴,包建平.棋类博弈算法的改进[J].现代电子技术,2005(01):96-99.
[6] 张海峰,白振兴,张登福.五子棋中的博弈智能设计[J],现代电子技术,2004,27(7):25-27.
[7] 王镌.博弈树搜索的算法改进[J].福建电脑,2004(2):27-28.
[8] 林尧瑞,马少平.人工智能导论[M].北京:清华大学出版社,2002.