六子棋博弈系统设计与实现

周新林,李淑琴,张晨光,赵学智,薛炜明

(北京信息科技大学 计算机学院,北京 100085)

摘 要:六子棋作为计算机博弈比赛项目越来越受到重视。从棋局表示、估值函数设计、搜索算法、走法生成器以及开局库几个方面介绍了六子棋博弈系统的设计与实现。该系统在“成理杯”2014 届全国大学生计算机博弈大赛六子棋项目比赛中,获得了一等奖的好成绩。大量模拟实验证明,该算法具有一定的有效性和实用性。

关键词:六子棋;估值函数设计;走法生成器;开局库

DOI:10.11907/rjdk.143983

中图分类号:TP319 **文献标识码**:A **文章编号**:1672-7800(2015)003-0092-03

0 引言

人工智能是研究智能的理论、方法、技术及应用系统的科学,其最关心的是知识表示与搜索,也是计算机博弈要解决的问题。计算机博弈也称为机器博弈,最早来源于博弈论思想。目前计算机博弈研究主要针对人机对战的棋盘类游戏,因而人工智能相关技术在计算机博弈游戏中被广泛应用。

六子棋由五子棋演变而来,2003 年由台湾新竹交通大学吴毅成教授提出,其规则简单、游戏公平、玩法复杂,受到玩家欢迎,逐渐得到推广。

六子棋标准棋盘为 19×19,黑白双方轮流下棋,除了第一次由黑方先下一颗子外,其后各方每轮下两颗棋子,连成六子(或以上)者获胜,如图 1 所示。六子棋没有禁手,长连(连成六子以上)即算赢棋,若全部棋盘填满仍未分出胜负则算和棋。

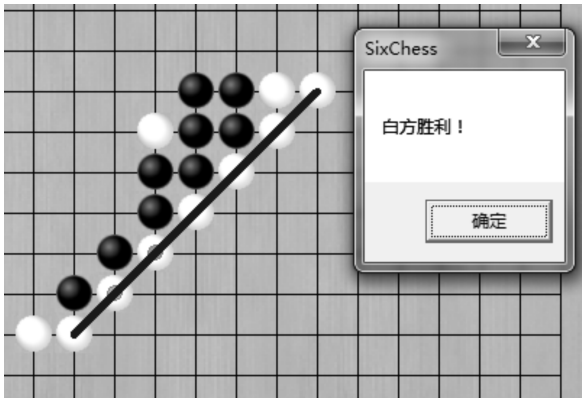

图 1 六子棋棋盘

本文主要从棋盘表示、走法生成器、估值函数设计、搜索算法实现以及开局库几个方面介绍六子棋博弈系统的设计与实现。

1 棋盘表示

棋盘的表示一般有数组表示法和比特表示法,比特表示法又称为位棋盘。六子棋棋盘一般为 19×19,若用二维数组表示棋盘即为 Board[19][19]。

采用位棋盘的作用就是记录棋盘上的某些布尔条件,根据布尔值放入对应的棋子。位棋盘如今广泛应用在六子棋中,原因是效率高、占用空间小。本文程序采用 64 位长度的变量,与数组表示法相比较,位棋盘每 19 个点需要 38 位。用 64 位长度的变量表示,19 行就需 19×8B,即为 152B,而数组表示法需要 361B。

2 走法生成器

六子棋中所有的位置都是合法位置,但如果将其都加入到走法生成器中,对后续的估值会造成极大的负担,进而导致搜索深度受到限制。因此对棋盘采用合适的方法选择有效的位置再进行估值,将有利于搜索的深度。走法生成器需考虑 3 个要素:①考虑能让我方形成六连局面的位置;②考虑限制对方形成六连的位置;③考虑其它合法位置。

3 估值函数设计

六子棋常用的估值方法有两种:一种是采用棋型分析方法,另一种采用“路”的分析方法。本程序为了提高估

基金项目:北京市教委教育教学一本科生科研训练项目(PXM2014_014224_000079)

作者简介:周新林(1993—),男,四川成都人,北京信息科技大学计算机学院学生,研究方向为计算机博弈。

值价值、减小搜索量，采用棋型分析和“路”分析相结合的方法。

3.1 基于棋型的估值函数设计

在估值分析中采用棋型分析方法。由于六子棋每次落两子的缘故，在估值分析中便会分析两层，棋型分析也分为两类，一类棋型是将局势向己方有利的局面发展，另一类是下棋时会直接产生胜负。我们在程序中采用常见棋型估值分析方法计算局面价值，减少搜索量，判断合理落子顺序。常用的位置信息有眠五、眠四、活五、活四、活三等，具体如下：

眠五：在同一直线上有 5 颗同色棋子，符合“对方用一手棋就能挡住或长连”的棋型；

活四：在同一直线上有 4 颗同色棋子，符合“对方必须用两手棋才能挡住或长连”的棋型，如图 2 所示；

眠四：在同一直线上有 4 颗同色棋子，符合“对方用一手棋就能挡住或长连”的棋型，如图 3 所示；

活三：在同一直线上有 3 颗同色棋子，符合“在下一手就能形成活四”的棋型，如图 4 所示；

活五：在同一直线上(包括对角斜线，以下同)有 5 颗同色棋子，符合“对方必须用两手棋才能挡住或长连”的棋型，如图 5 所示；

朦胧三：在同一直线上有 3 颗同色棋子，符合“在下一手棋只能形成眠四，而如果在下两手棋的话就能形成活五”的棋型；

眠三：在同一直线上有 3 颗同色棋子，符合“在下两手也只能形成眠五”的棋型；

活二：在同一直线上有 2 颗同色棋子，符合“在下两手就能形成活四”的棋型；

眠二：在同一直线上有 2 颗同色棋子，符合“在下两手也只能形成眠四”的棋型。

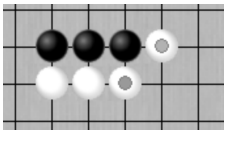

图 2　活四示意图

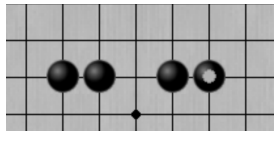

图 3　眠四示意图

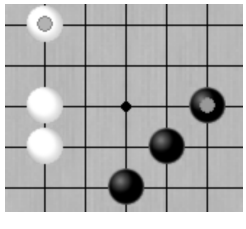

图 4　活三示意图

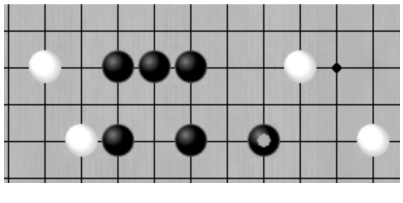

图 5　眠三示意图

3.2 基于路的估值函数设计

路也是估值分析中常用的方法。所谓“路”是指在棋盘上存在连续 6 个可能连成一线的点位，由于每条“路”上有 6 个连续点位，这样，判断棋型就变得更为简单。比如：某“路”有 5 颗同色棋子，不必关心它是活五还是眠五，一律统称为五路。路的总数减少，可以有效降低棋型判断的复杂度，方便计算棋盘的状态值。

基于路的函数设计需要从横向、纵向、左斜和右斜 4 个方向根据同色棋子的分布情况进行统计，然后再根据不同的路进行价值计算就可获得当前局面的价值。

Score 的分值决定了当前的局面情况，ScoreofRoad[i]是形成不同路的价值，且不同路的价值随长度增加而增加。例如基本设置可设为 ScoreofRoad[i]={0，20，80，150，800，1000，10000}，ScoreofRoad[0]代表路中没有棋子，ScoreofRoad[6]代表路中有 6 颗同色棋子。

双方每种路的条数计算如下：

纵向和横向分别为：19 行×(19－6＋1)路/行＝266 路；左斜向和右斜向分别为：14 列×(19－6＋1)路/路＝196 路。

因此在 19×19 的棋盘上共有：266×2＋196×2＝924 路，只记录双方合法的路数。合法的路数是指在该路上没有对方的棋子(只有相同颜色的棋子或空格的路)，最终将计算得到的路数(包括 NumberOFMyRoad 和 NumberOfEnemyRoad)与 ScoreOfRoad 一起代入公式就可得到 Score，这样就可对一个局面进行评估。

4 开局库的使用

在比赛过程中通常有时间限制，设计良好的开局库不仅可节省时间，还可弥补估值的不足。传统开局库少、估值不足，可以将丰富的开局库放入相应数据库，在开局依据搜索估值函数的调用选择合适的开局库。

目前常用的开局库大致有 70 余种，针对比赛使用的开局库有时可达到 5～7 手开局库。将理想的开局库放入相应的数据库中，开局库通常只需要添加、删除和修改等基本功能。开局库通常需要与搜索相结合，在开局阶段，首先调用开局库，查找是否有合适的开局，若有则采用开局库中的相关下法，若没有则通过搜索和估值获得最佳位置。

本程序设计中，当比赛开始时，如果对方先手落在棋盘中心，程序往往会随机调用开局库获得相应位置，否则会通过搜索和估值获得最佳位置。图 6～图 12 为几种常见的开局库。

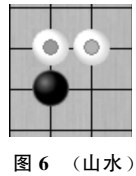

图 6　(山水)

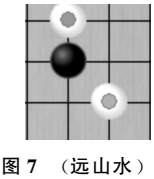

图 7　(远山水)

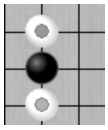

图 8 （远山山）

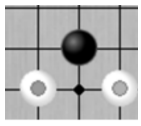

图 9 （水水）

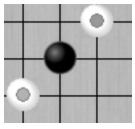

图 10 （远水水）

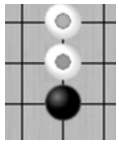

图 11 （山心）

5 搜索算法实现

目前常用的博弈搜索算法有极大极小算法、AlphaBeta 减枝算法、爬山法、遗传算法。本程序采用 AlphaBeta 减枝算法，下面介绍其原理。

AlphaBeta 减枝算法：其来源于极大极小算法。极大极小算法思想是：开始时两方总和为零，一方要在可选项中选择优势最大化，一方选择令对手优势最小化。但极大极小法由于搜索深度和可能的情况很多，且不是每个节点都有搜索的必要，导致搜索效率受到影响，故采用演化得到的 AlphaBeta 减枝算法。

Alpha 剪枝：在对博弈树进行深度优先的搜索策略时，从左路分枝的叶节点倒退，得到某一层 MAX 节点值，记为 Alpha，将此值记为 MAX 方指标的下界。在搜索此 MAX 节点的其它子节点时，如果发现评估值小于下届 Alpha 值，则可以减掉此枝。

Beta 剪枝：在对博弈树进行深度优先的搜索策略时，从左路分枝的叶节点倒退得到某一层 MIN 节点值，记为 Beta，将此值记为 MAX 方指标的上界。在搜索此 MIN 节点的其它子节点时，如果发现评估值高于上届 Beta 值，则可以减掉此枝。

一个 MAX 节点的 Alpha 值等于其后继节点当前的最终倒推值，一个 MIN 节点的 Beta 值等于其后继节点当前最小的最终倒推值，如图 12 所示。

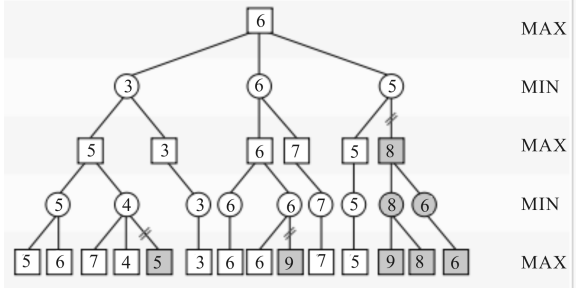

图 12 AlphaBeta 减枝算法

AlphaBeta 减枝算法伪代码如下：

```
intalphaBeta(ChessBoard board, int depth, int alpha, int beta)
{
int value;
if( depth == 0 || board.isEnded())
{
value = evaluate(board);
return value;
}
board.getOrderedMoves();
int best = -MATE-1;
int move; ChessBoardnextBoard;
while (board.hasMoreMoves())
{
move = board.getNextMove();
nextBoard = board.makeMove(move);
value = -alphaBeta(nextBoard, depth-1, -beta, -alpha);
if(value > best)
best = value;
if(best > alpha)
alpha = best;
if(best >= beta)
break;
}
return best;
}
```

6 结语

本文从 4 个方面介绍了六子棋构成，基于路的估值函数设计、走法生成器、开局库的使用，以及 AlphaBeta 减枝搜索算法。优秀的搜索算法搭配上良好的开局库，不仅可以节省比赛时间，还能提高搜索效率。经过大量模拟实验，该策略在和随机策略的比赛试验中，具有明显优势。将该策略应用于“成理杯”2014 届全国大学生计算机博弈大赛六子棋项目比赛中，获得了一等奖的好成绩，证实了这种策略具有一定的优越性和实用性。

参考文献：

[1] 王静文，吴晓艺．全国大学生计算机博弈大赛培训教程[M]．北京：清华大学出版社，2013．

[2] 黄继平，张栋，苗华．六子棋智能博弈系统的研究与实现[J]．电脑知识与技术，2009(5)：165-169．

[3] 王永庆．人工智能原理与方法[M]．北京：清华大学出版社，2005．

[4] 王小春．PC 游戏编程(人机博弈)[M]．重庆：重庆大学出版社，2002．

[5] GILLOGLY J．Performance analysis of the technology chess program [D]．Pittsburgh：Carnegie-Mellon Univ．，1978．

[6] PLAAT A，SCHAEFFER J，PIJLS W，et al．SSS * =-β+ TT[R]．Canada：Technical Report TR-CS-94-17，Department of Computer Science，University of Alberta，Edmonton，AB，1994．

（责任编辑：杜能钢）