维普资讯 http://www.cqvip.com

2005 年 4 月
第 22 卷 第 2 期
枣庄学院学报
JOURNAL OF ZAOZHUANG UNIVERSITY
Apr.2005
Vol.22 NO.2

智能五子棋博弈程序的核心算法

董红安[1]，蒋秀英[2]

（1.山东师范大学 信息管理学院，山东 济南 250014；2.枣庄学院 计算机科学系，山东 枣庄 277160）

[摘 要]人工智能是一门正在迅速发展的新兴的综合性很强的边缘科学，而博弈是人工智能的主要研究领域之一，本文通过一个五子棋博奕程序的设计，介绍了博弈程序设计的核心内容：包括博奕树搜索和估值函数两个方面.

[关键词]人工智能；博弈；五子棋；博奕树；估值

[中图分类号]TP319；G891.9 [文献标识码]B [文章编号]1004－7077(2005)02－0061－05

0 引言

人工智能是近年来很活跃的研究领域之一，计算机博奕是人工智能研究的重要分支.人工智能中大多以下棋（如象棋、围棋、五子棋等）为例来研究计算机博奕规律.五子棋是一种深受大众广泛喜爱的游戏，其规则简单，变化多端，非常富有趣味性和消遣性.这里设计和实现了一个人机博奕的五子棋程序，采用了博弈树的方法，应用了剪枝和极大极小树原理进行搜索发现最好的下子位置.介绍五子棋程序的数据结构、评分规则、胜负判断方法和搜索算法过程.

1 计算机博奕程序的主要内容

要想实现一个让计算机能够下棋的程序，至少应具备如下五个部分[1]：

1.1 状态表示：某种在机器中表示棋局的方法，让程序知道博弈的状态.

1.2 走法产生：产生合法走法的规则，以使对弈公正的进行.

1.3 搜索技术：从所有合法的走法中选择最佳的走法的技术.

1.4 估值函数：一种评估局面优劣的方法，用以同搜索技术配合做出智能的选择.

1.5 对弈界面：有了界面，对弈才能进行.

下面分别介绍以上五个部分的设计.

2 棋局状态表示及相关的数据结构

五子棋程序的棋盘状态及主要的数据表示如下：

2.1 棋盘的状态用一个 15×15 的二维数组表示.

2.2 用数字 0 和 1 来表示不同的棋子，其中黑色棋子用"0"表示，白色棋子用"1 表示.

2.3 没有棋子的格子用 0xFF 表示.

为了在使用数据时能够避免数据表示出差错，我们将棋盘状态定义成一系列便于使用的宏.下面是棋盘状态的数据表示的宏定义.

```
# define   GRID-NUM   19   //每一行列的棋盘交点数
# define   GRID-COUNT   361   //棋盘上交点总数
# define   BLACK   0   // 黑棋用 0 表示
# define   WHITE   1   // 白棋用 1 表示
```

[收稿日期]2005－03－09

[作者简介]董红安(1969－)，男，山东滨州人，山东师范大学信息管理学院在读硕士研究生，主要从事软件理论与技术研究.

维普资讯 http://www.cqvip.com


```
#define  NOSTONE  0xFF
typedef  struct-stoneposition//用以表示棋子位值的结构
{
BYTE x;
BYTE y;
}STONEPOS;
typedef  struct -stonemove
{
STONEPOS  StonePos;  //棋子位置
int  Score  //走法的分数
}STONEMOVE;  //这个结构用以表示走法
BYTE m-RenjuBoard[GRID-NUM][GRID-NUM]//棋盘状态的数组
```

3 走法产生

五子棋的走法产生相对简单一些,对于五子棋盘来说所有空白的交点位置都是合法的落子点(本走法是针对业余五子棋而言,而职业五子棋对奕有三·三、四·四等禁手的规则),走法产生的算法如下:

```
int CreatePossibleMove(BYTE positon[][GRID-NUM],int depth)
// depth 表示搜索的深度
{
int i,j;
m-nMoveCount = 0; //合法走法的个数
for(i = 0;i < GRID-NUM;i + + )
for(j = 0;j <  GRID-NUM;j + + )
{
if(position[i][j] = = (BYTE) NOSTONE) // NOSTONE表示空白
{
AddMove(j,i,depth) //加入当前走法;
nMoveCount + + ;
}
}
return nMoveCount;
}
```

4 估值函数

在博弈程序的几大主要部分里,估值核心(估值函数)是与具体的棋类知识紧密结合的一部分,可以说估值函数在很大程度上决定了博弈程序的棋力高低.对于下子的重要性评分,可以从四个方向来考虑当前棋局的情况,分别为:水平、垂直、左斜、右斜.实际上需要考虑在这四个方向上某一方所形成的子的布局的情况,对于在还没有子的地方落子以后的当前局面的评分,主要是为了说明在这个地方下子的重要性程度,在此,设定了一个简单的规则来表示当前棋面对机器方和人方的分数.

在对当前棋盘的局面进行估值之前,首先介绍一下五子棋中的常用术语:

活三:由于一方走一步在无子交叉点上所形成的一个"活三"的局面,也就是3子无间隔的相连,并且此3子两端延长线上各有一个无子的交叉点与此3子紧密相连,死四:由于一方走一子后所形成的4子无间隔紧密相连,但是此4子两端延长线上只有一个无子的交叉点,而另

维普资讯 http://www.cqvip.com

一端的位置是对方的棋子.

基本的规则如下:棋形所对应的得分

	单独一子		二子相连		三子相连		四子相连		五子相连
棋形	活一	死一	活二	死二	活三	死三	活四	死四	五连
	20	4	400	90	6000	800	20000	10000	50000

搜索整个棋盘上得出双方共有多少个活一,死一,活二,…最后把双方各部分分别求和,得到当前局势的评价值.注意这里的规则是根据一般的下棋规律的一个总结,在实际运行的时候,用户可以添加规则和对评分机制加以修正.

最后,根据当前最后一个落子的情况来判断胜负.需要从四个位置判断,以该子为出发点的水平,竖直和两条分别为 45 度角和 135 度角的线,目的是看在这四个方向是否最后落子的一方构成连续五个的棋子,如果是的话,就表示该盘棋局已经分出胜负.

5 搜索算法

在五子棋对弈过程中,我们将对奕双方其中一位叫做甲,另一位叫做乙,假定现在该甲走棋,甲可以有 40 种走法(无论好坏),而对甲的任一走法,乙也可以有与之相对的若干种走法,然后又轮到甲走棋,对乙的走法家又有若干种方法应对,如此反复.我们可以依次构建一棵博奕树,将所有的走法罗列出来.博奕树是从初始局面(根部)向下递归产生的一棵包含所有可能的对奕过程的搜索树,成为完全搜索树.除了极少数非常简单的棋类游戏,大多数棋类游戏,我们都没有建立完全搜索树的可能.

在一棵博奕树中,如果我们令甲胜的局面值为 50000,乙胜的局面值为 - 50000,而和局的值为 0,当轮到甲走时,甲定会选择子节点值最大的走法;而轮到乙时,乙则会选择子节点值最小的走法.所以,对于中间节点的值可以有如下计算方法:如果该节点所对应的局面轮到甲走棋,则该节点的值是其所有子节点中值最大的一个的值.而如果该节点所对应的局面轮到乙走起,则该节点的值是其所有子节点值中最小的一个的值.我们将上述在一棵博奕树中搜索一个好的走法的方法成为极大极小搜索[2].我们可以利用上面的估值函数对博奕树的每一个局面进行评分,然后我们就可以通过极大极小搜索在博奕树中为机器寻找最佳的合法走法了.

用伪代码将深度优先搜索极大极小树算法描述如下

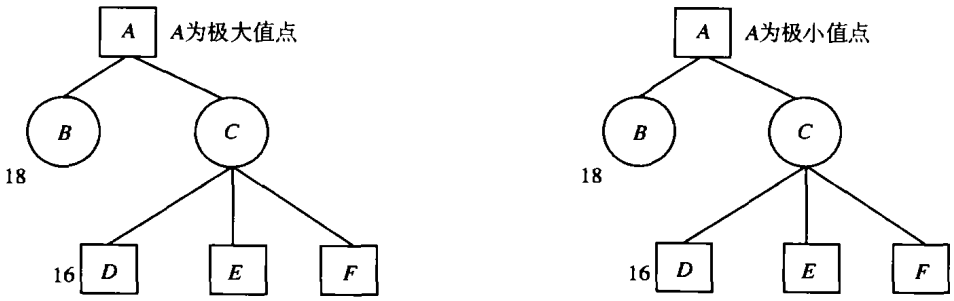
```
int miniMax(position p,int d)
{
int bestvalue,value; //
if(Game Over) //检查棋局是否结束
return evaluation(p); //棋局结束,返回棋局
if(d < =0) //是否叶子节点
return evaluation(p); //叶子节点,返回估值
if(p.color = = BLACK) //是否轮到黑方走棋
bestvalue = - INFINITY; //是,令初始最大值为极小
else
bestvalue = INFINITY; //否,令初始最大值为最大
for( each possibly move m) //对每一可能的走法 m
{
MakeMove(m); //产生第 i 个局面(子节点)
Value = MiniMax(p,d - 1); // 递归调用MiniMax向下搜索子节点
UnMakeMove(m); //恢复当前局面
```

维普资讯 http://www.cqvip.com


```
If(p.color = = BLACK)
bestvalue = Max(value,bestvalue); //取最大值
Else
Bsetvalue = Min(value,bestvalue); } //取最小值
return bestvalue; //返回最大/最小值
}
```

在极大极小搜索的过程中，存在着一定程度的数据冗余.如图1所示左半部的一棵极大极小树的片断，节点下面为该节点的值，设A为极大值节点，节点B的值为18，节点D的值为16，由此可以判断节点C的值将小于等于16(取极小值)；而节点A的值为节点Max(B,C)，是18，也就是说不再需要估节点C的其他子节点如E、F的值就可以得出父节点A的值了.这样将节点D的后继兄弟节点减去称为Alpha剪枝(alpha cutoff).

同样在图1右半部一棵极大极小树的片段中，设A为极小值节点，节点B的估值为8，节点D的估值为18，由此可以判断节点C的值将大于等于18(取极大值)；而节点A的值为Min(B,C)，是8，也就是说不再需要求节点C的其他子节点如E、F值就可以得出父节点A的值了.这样将节点D的后继兄弟节点剪去称为Beta剪枝(beta cutoff).



将Alpha剪枝和Beta剪枝加入MiniMax搜索，就得到Alpha－beta搜索算法，将这个算法用类C的伪代码描述如下：

```
int AlphaBeta(nPly,nAlpha,nBeta)
{
if(game over)
return Eveluation; //胜负已分，返回估值
if(nPly = = 0)
return Eveluation; // 叶子节点返回估值
if(Is Min Node) //此句用于判断当前节点是何种节点
{//是取极小值的节点
for(each possible move m) //对每一可能的走法 m
{
make move m; //生成新节点
score = AlphaBeta(nPly - 1,nAlpha,nBeta);// 递归搜索子节点
unmake move m; //撤销搜索过的节点
if(score < nBeta)
{
nBeta = score; //取极小值
if(nAlpha > = nBeta)
return nAlpha; //剪枝，抛弃后继节点
}
}
}
return nBeta; //返回最小值
}
else
{//取极大值的节点
```

维普资讯 http://www.cqvip.com


```
for(each possible move m) //对每一可能的走法 m
{
make move m; //生成新节点
score = AlphaBeta(nPly - 1,nAlpha,nBeta); //递归搜索子节点
unmake move ; //撤销搜索过的节点
if(score > nAlpha)
{
nAlpha = score; //取极大值
if(nAlpha > = nBeta)
return nBeta ; //剪枝,抛弃后继节点
}
}
return nAlpha; //返回最大值
}
}
```

6 小结

在 Windows 操作系统下,用 VC ++ 实现了这个人机对战的五子棋程序.和国内许多只是采用规则或者只是采用简单递归而没有剪枝的那些程序相比,在智力上和时间有效性上都要好于这些程序.同时所讨论的方法和设计过程为用户设计其他的游戏(如象棋和围棋等)提供了一个参考.

参考文献

[1]王小春.游戏编程(人机博译)[M].重庆:重庆大学出版社,2002.
[2]蔡自兴.人工智能及其应用[M].北京:清华大学出版社,1999.

The Core Algorithm Ways of Renju Game - Playing Program of Intelligence

DONG Hong - an[1], JIANG Xiu - ying[2]
(1.Shandong Normal University, Ji'nan 250014, China;
2.Zaozhuang University, Zaozhuang 277160, China)

Abstract: The artificial intelligence is a just very strong edge science of newly arisen comprehensive that develop quickly, and the game - playing is one of the main research realms of the artificial intelligence. This text passes one program designs of the Renju of game - playing, introducing the core contents of the program design of game - playing: Include the game - playing tree search and the function of the eveluation two aspects.

Key words: artificial intelligence; game - playing; renju; the tree of game - playing; the eveluation