文章编号：2095-2163(2021)02-0129-03 中图分类号：TP312 文献标志码：A

基于 UCT 搜索算法的点格棋博弈系统研究

朱良双，王静文，李 媛
（沈阳工业大学 理学院，沈阳 110870）

摘 要：蒙特卡罗树搜索（MCTS）在许多完备的信息双人游戏中获得成功。本文给出了 UCT（ Upper Confidence Bound Apply to Tree）算法结合了 UCB 公式和蒙特卡洛树搜索算法，同时与局面评估相结合，根据点格棋长链和环的特点对算法进行了优化。有利于更快更准地找到当前局面的最优解。

关键词：UCT 算法；估值函数；点格棋

Dots and Boxes game base on UCT algorithm

ZHU Liangshuang, WANG Jingwen, LI Yuan
(School of Science, Shenyang University of Technology, Shenyang 110870, China)

【Abstract】Monte Carlo tree search (MCTS) has been successful in many perfect information games. In this paper, UCT(Upper Confidence Bound Apply to Tree) algorithm is proposed, which combines UCB formula and Monte Carlo tree search algorithm. Meanwhile combined with situation assessment, the selection of nodes for evaluation by UCB algorithm is conducive to find the optimal solution of the current situation faster and more accurately.

【Key words】UCT algorithm; evaluation function; Dots and Boxes

0 引 言

众所周知，点格棋是由数学家爱德华·卢卡斯提出的一种只需要在纸上就可以进行的游戏。与一般的传统游戏不同，点格棋的玩法是通过点与点之间的边来占领格子，再根据所占领区域大小来判定胜负，是一种将图论、数学等知识结合在一起的游戏。

在国内，自 2011 年起在全国大学生计算机博弈竞赛已将该项目作为竞赛项目之一，随着国内外各类机器博弈赛事的陆续举办，对于点格棋搜索算法的研究受到了越来越多的爱好者关注，在 2020 年的全国大学生计算机大赛中，点格棋的参赛队伍已达到 27 支，为历年最多。

1 点格棋简介

点格棋的棋盘大小可以根据情况进行设置，典型的点格棋棋盘由 6×6 的点围成的 5×5 的格子，如图 1 所示。

图 1 中，棋盘中共 60 条边，格子数为 25，由于比赛胜负是按照双方所占格子数来确定，格子数为奇数时可以避免出现平局情况。

点格棋的基本规则如下：

（1）双方轮流在水平或竖直方向的相邻两点之间下棋。

（2）当每个格子四条边被占满时，这个格子被最后一条边的捕获者所占领。

（3）当一方占有格子后，必须在下一条边，直至不再占有格子。

（4）当格子全部围成后，根据格子归属统计得分，捕获格子多的一方获胜。

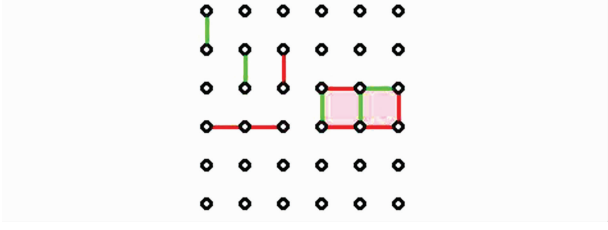

图 1 点格棋棋盘

Fig. 1 The board of Dots and Boxes

2 搜索算法

点格棋博弈系统的构成要素主要由搜索算法和估值函数两个部分组成[1]。传统的搜索算法是以极大极小算法为基础并加以改进而来，UCT 搜索算法是近几年在计算机博弈游戏设计被逐步采用的一种算法，具有时间可控、搜索效率高的特点。

作者简介：朱良双(1999-)，男，本科生，主要研究方向：计算机博弈；王静文(1965-)，男，工程师，主要研究方向：人工智能和信息安全；李 媛(1976-)，女，博士后，教授，主要研究方向：人工智能和随机过程。

收稿日期：2020-11-19

UCT 算法(Upper Confidence Bound Apply to Tree),即上限置信区间算法,是将蒙特卡洛树搜索(Monte-Carlo Tree Search,MCTS)方法与 UCB 公式结合,在超大规模博弈树的搜索过程中相对于传统的搜索算法有着时间和空间方面的优势,UCT 的工作模式是时间可控的[2-3]。在算法执行过程中的任何时间突然终止算法,UCT 算法可以返回一个理想的结果。当然如果允许更为充分的执行时间的话,算法结果会非常逼近实际的最优值。UCT 算法的计算方法如下:

$$r_i = v_i + c \times \sqrt{\frac{2\ln(\sum_i T_i)}{T_i}}\ , \qquad (1)$$

其中,v_i表示以节点 n_i(非叶节点)为根节点的所有仿真结果的平均值,反映了根据目前仿真结果观察到的节点 n_i能提供的回报值的期望;T_i表示节点 n_i的访问次数,也是被树内选择策略选中的次数;c 表示一个参数,用来平衡深度优先还是宽度优先。

UCT 算法的执行过程如图 2 所示。

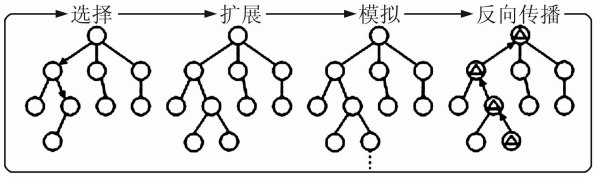

图 2 UCT 算法的基本过程

Fig. 2 The process of UCT

这里,对 UCT 算法的 4 个过程的设计功能可分述为:

(1)选择:从根节点出发,用递归的方法对于每一个孩子节点进行选择,选择 r_i最大的节点作为下一次选择的开始,直到叶子节点。

(2)扩展:通过选择的节点,将所有合法的下法作为新的叶子节点加入到搜索树中,并正确初始化 v 值和 T 值。

(3)仿真:简单的仿真策略是对所有合法的下法,均匀地随机选择下一步,叶子节点的评估策略可以比较简单,例如:当己方获胜时为 1,对方获胜时为 0。通过若干次仿真后获得新的 v 值。

(4)反向传播:当叶子节点通过仿真获得新的 v 和 T 时,UCT 算法通过结果回传更新搜索路径上的所有节点 v 和 T 的值,其计算公式为:

$$T = \sum_i T_i, \qquad (2)$$

$$v = \frac{\sum_i v_i T_i}{T}\ . \qquad (3)$$

由式(2)、式(3)可知,父节点的访问次数 T 是所有叶子节点访问次数 T_i之和,v 是当前节点下所有叶子节点 v_i的加权平均值,权值为子节点的访问比例 T_i/T。结果回传从叶子节点开始,沿搜索路径逐级向上更新,直到根节点。

点格棋是通过占边从而达到捕获格子的目的,而边的总数只有 60 条,数量相对较少,比较适合使用 UTS 搜索算法进行搜索。

将 UCT 算法结合到点格棋博弈系统中的伪码如下:

```
FunctionUCT(Node rootNode)
currentNode = rootNode
while(currentNode ∈ T)
    lastNode = currentNode
    currentNode = select(currentNode)// 选择
lastNode = Expand(lastNode)// 扩展
R = playSimulatedGame(lastNode)// 模拟
while(currentNode ∈ T)// 反向传播
    currentNode = backPropagate(R)
    currentNode. visitCount ++
    currentNode = currentNode. parent
return bestMove
```

在搜索过程中,若仅采用基本的 UCT 搜索算法进行搜索,搜索的效率就比较低,特别是在搜索的初期,会产生大量的无效模拟,从而降低搜索效率。在采用 UCT 算法进行搜索过程中,引入估值,对节点的评估采用优化估值的方法进行,用以提高搜索效率和搜索的准确度。

3 优化估值方法

点格棋的下法与一般棋类游戏的下法略有所不同,在完成一步下棋后,需要对是否捕获格子来做出判断,若捕获格子则要再继续下一步走棋,直到不再捕获格子为止,这个过程会导致一方连续捕获格子的情况。在博弈中后期,需要直接捕获局面,并且做出让格(中断捕获格子)、还是不让格的判断,所以在进行搜索前就要对当前局面进行评估和预处理,预处理由 2 部分组成,对此可阐释如下。

(1)当棋局出现死格时,直接捕获该死格。

(2)棋局中出现死树时,则对局面进行评估,根据连和环的定理判断是否留下最后两格。死格和死树的状态如图 3 所示。

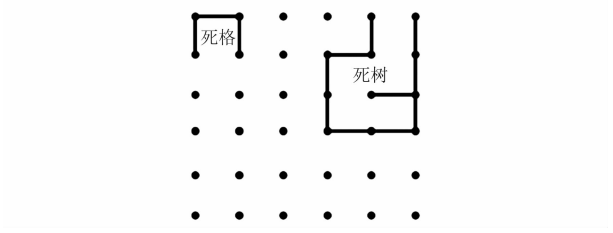

图 3　死格和死树

Fig. 3　Dead grid and dead tree

对局面进行估值主要分为 2 方面。一方面是局面的控制估值，用于判断对棋局是否保持控制[4-5]。公式如下：

$$c(G) = size(G) - 4 * long\_chain - 8 * loops + tb(G), \tag{4}$$

其中，G 表示当前棋局局面[6]；$c(G)$ 表示当前局面的控制值，即 $control\_value(G)$；$size(G)$ 表示 G 中未被占领格子数；$long\_chain$ 表示长链数；$loops$ 表示环的数目。$tb(G)$ 取值如下：

$$tb(G) = \begin{cases} 0, & G \text{ has no chains or loops;} \\ 8, & G \text{ has one or more loops;} \\ 6, & G \text{ has one or more loops and 3-chains;} \\ 4, & \text{otherwise.} \end{cases}$$

另一方面，如果对手下边，当打开一个环，除去该环后的 $control\_value(G)>2$；或者打开一条链，除去该链后的 $control\_value(G)>4$，则己方应保持控制，否则放弃控制。

点格棋的局面估值的伪码如下：

Functionvalue( )

if $control\_value \geqslant 2$ *then value* = *control_value.* ,

if $control\_value = 0$ and

$G = 4l + (anything\ except\ 3 + 3)$,

then *value* = 0.

if *number of* 3 − *chains* = 0, *or* if *number of* 3 − *chains* = 1

and $size(G) \equiv 3 \bmod 4$, then

if $control\_value + number\ of\ 4 - loops \geqslant 2$ then *value* = 0,

1,2,3,4 *according to whethercontrol_value* $\equiv 0, \pm 1, \pm 2, \pm 3, 4\ mod\ 8$.

if $control\_value + 4f < 2$ then

if *G is odd* then *v* = 1 *resp.* 3 if *f is odd resp. even.*

if $size(G) \equiv 2 \bmod 4$ then *value* = 2.

if $size(G) \equiv 0 \bmod 4$ then *value* = 0 *resp.* 4 if *number of*

4 − *loops is odd*

resp. even.

in all other cases, *value* = 1, 2 *as* $size(G)$ *is odd*,*even.*

return *value*

4　实验与结果

在对比实验中，采用的博弈系统为相同的 UCT 搜索算法，节点的估值则分别采用 UCB 计算公式和优化估值进行计算。UCT 搜索算法的时间限制分别设为：5.00 s，10.00 s，15.00 s 和 20.00 s。并分别设定先后手来进行对弈，得到的结果见表 1。

表 1　对比结果表

Tab. 1　Table of comparative results

搜索时间/s	UCT 搜索胜率/%	UCT+最佳估值搜索胜率/%
5.00	21.02	79.98
10.00	16.15	83.85
15.00	12.13	87.87
20.00	8.21	91.79

通过表 1 分析发现，在采用相同的 UCT 搜索算法的情况下，运用估值函数的胜率要显著高于运用 UCB 值搜索的胜率。对于点格棋，各种局面的长链和环一般的估值并不适用。因此，该估值函数有显著优势，同时，随着每步搜索时间的增加，使用优化后的估值函数的搜索算法胜率更高。

5　结束语

本文通过先后手多轮对弈比较，计算出运用 UCB 估值和最佳估值的不同胜率。通过胜率对比结果表可以看出，采用优化估值的方法对于局面的评估提供了一定的帮助，并能有效提高博弈胜率。

参考文献

[1] ALLCOCK D. Best play in dots and boxes endgames[EB/OL]. [2021-02-04]. https://doi.org/10.1007/s00182-020-00730-4.

[2] 张宜放，孟坤. 基于点格棋的 UCT 算法研究与分析[J]. 智能计算机与应用，2020，10(4)：27-31.

[3] 刘洋. 点格棋博弈中 UCT 算法的研究与实现[D]. 合肥：安徽大学，2016.

[4] BERLEKAMP E. Forcing your opponent to stay in control[J]. Plos One, 2001, 11(1).

[5] BREWKA G, HABEL C, NEBEL B. An artificial intelligence approach to dots - and - boxes [C]// German Conference on Artificial Intelligence: Advances in Artificial Intelligence. Berlin/Heidelberg: Springer-Verlag, 1997.

[6] 张雪峰，连莲，徐心和. 基于有限自动机的“点点连格”机器博弈系统的建模与分析[J]. 沈阳建筑大学学报(自然科学版)，2009，25(4)：796-801.