

亚马逊棋中评估函数的研究

陈莹华¹, 杨玲²

(1. 武警海警学院, 浙江 宁波 315801; 2. 武警特警学院, 北京 102211)

摘要: 机器博弈是人工智能的重要领域, 国内外普遍采用棋类作为研究机器博弈技术的载体。以亚马逊棋为载体, 总结了实现亚马逊棋机器博弈的几个基本部分, 并着重对亚马逊棋对局中用于评定招法优劣的评估函数做了初步研究。在处理评估函数时用到了领地评估特征 territory、位置特征 position、灵活度特征 mobility 三个评估特征, 并提出使用关于回合数的一次函数加权的计算模型, 最后通过实验进行了调参和检验。

关键词: 亚马逊棋; 评估函数; territory; position; mobility

中图分类号: TP312 文献标识码: A

文章编号: 1009-3044(2019)08-0224-03

DOI: 10.14004/j.cnki.ckt.2019.0806

Research on Evaluation Function for the Game of Amazons

CHEN Xuan-hua¹, YANG Ling²

(1. China Coast Guard Academy, Ningbo 315801, China; 2. Special Police College of the Chinese People's Armed Police Force, Beijing 102211, China)

Abstract: Machine game is an important field of artificial intelligence. Chess is widely used as the carrier of machine game technology at home and abroad. Taking the computer-game of Amazons as the carrier, this paper summarizes several basic parts of the realization of Amazons' computer-game, and focuses on the preliminary study of the evaluation function of Amazons' game for evaluating the merits and demerits of tactics. In dealing with the evaluation function, three evaluation features which include territory, position and mobility are used. A weighted calculation model of the first-order function about the number of rounds is proposed. Finally, the parameters are adjusted and tested by experiments.

Key words: Amazons; evaluation function; territory; position; mobility



开放科学(资源服务)标识码(OSID):

1 背景

亚马逊棋是由阿根廷沃尔特 Zamkauskas 于 1988 年发明的双人抽象策略游戏, 由国际象棋中的皇后走法衍生而来, 策略和思路又类似中国围棋中的圈地, 其中的机器下棋部分用到计算机博弈技术来实现, 且属于完全信息动态博弈。

本文总结了实现亚马逊棋机器博弈的几个基本部分, 并着重对亚马逊棋对局中用于评定招法优劣的评估函数做了初步研究。

2 亚马逊棋的基本内容

2.1 游戏规则

棋盘是 $n \times n$ (比赛时通常采用 10×10) 的方格, 分为黑白两方, 双方各有四子, 黑色在上、白色在下; 游戏开始后双方轮流行棋, 黑方先行, 每一轮中双方行棋都必须完成走棋、设置障碍(放箭)两个步骤。走棋原则是双方的八个棋子都类似国际象棋中的皇后, 可以向横、竖、斜对角八个方向移动若干个格子,

但不能穿过棋子和障碍, 也不能吃子。一轮中一方只能移动一个棋子, 完成走棋后, 由移动的棋子在落子的棋位上开始设置障碍, 障碍又称为“箭”, 障碍的释放方式和走棋方式相同。当某一方完成某次移动后, 对方的四个棋子均不能移动时, 系统判定对方输掉该局比赛。

2.2 实现亚马逊棋机器博弈的基本部分

亚马逊棋机器博弈的基本部分包括: 棋局表示、招法生成、招法搜索、评估函数。其中对博弈树的招法搜索以及通过评估函数计算博弈树节点的估值是机器博弈的核心。

棋局表示包括局面上的双方棋子走动、设置障碍等, 可以用一个 $n \times n$ (10×10) 的二维数组表示, 将黑方棋子、白方棋子、障碍、空格设定不同的表示记录下来即可, 程序通过它来获取当前博弈的状态。

招法生成是产生在当前局面下某一方落子、设置障碍的可行方法, “可行”通过下棋规则来进行描述, 确保双方博弈公平公正地进行, 当行棋不符合游戏规则时系统将判决“非法棋步”并发出提醒或直接判输, 基于游戏规则, 一个完整的招法应该

收稿日期: 2019-01-29

作者简介: 陈莹华(1969-), 女, 福建尤溪人, 武警海警学院电子技术系教授, 主要从事计算机应用方面的教学与研究工作。

至少包括三个数据:起点、终点、障碍放置点。

招法搜索用于在生成的所有可行招法中利用评估函数找到最优招法,是实现亚马逊棋中机器博弈的重要部分,体现了人类思考的模式,可以使用极大极小算法、alpha-beta剪枝等各种剪枝算法,并可尝试搜两层、搜三层统计搜索准确性和搜索时间^[1]。

评估函数体现了执棋方在模拟某个招法后棋局的优劣,使得程序不需要一直深度模拟至游戏结束,用于在搜索到的所有可行招法中找到最优招法,因此,评估函数是亚马逊棋机器博弈中非常重要的部分,评估函数的好坏很大程度上决定了搜索的准确性,影响到棋局的胜负。

3 评估函数

亚马逊棋的规则涉及四个子、皇后的行棋方式、走棋、设置障碍(放箭)等,这使得亚马逊棋不同局面下每一步的可行走法数量十分庞大,平均在1000多种左右,第一步有2176种可行走法,大大提高了游戏的复杂程度。因此,我们需要用评估函数对搜索到的所有可行招法进行估值,据此选出最优招法。

在亚马逊棋博弈中,因为最终目的是使得对方无棋可走,所以在局面对峙过程中要尽可能的较对方圈得更大的地,一般针对己方和对方考虑以下两种策略:针对己方采取策略圈的策略,圈占领地,通过障碍设置等使己方棋子拥有较大的领地而对方无法进入该领地,即扩大己方领地;针对对方采取策略堵的策略,将对方的棋子堵在空格数很小的范围内,即限制对方领地。当棋局进入一定的时期(比如18个回合之后),双方棋子都已经被限制在各自的领地中了,此时双方的行棋将互相无法对对方产生影响,双方棋子互不干扰地各自在己方领地中行棋至游戏结束。采用上述两种策略后,圈得领地较少的一方会先走完(放置障碍)所有的空格,输掉比赛。这两种策略在评估函数中具体体现为领地评估特征 territory、位置特征 position 和灵活度特征 mobility。

3.1 Queen走法和King走法

Queen走法是按照国际象棋中的皇后走法,可以向横竖斜对角八个方向走直线,只要路径上没有障碍就可以沿直线一直走;king走法则是按照国际象棋中的国王走法,即只能向八个方向走一格。棋子的走法和放箭都采用Queen走法。^[2]

3.2 Territory值

territory值分为tq和tk两个。tq代表用queen走法圈的领地数量,相应的,tk代表用king走法圈的领地数量,结合考虑king走法的意义,tk其实就是某一方对某个格子周围一圈(相邻8个格子)的控制权大小。

计算某一方的territory值时,需要分两步进行,第一步:得到棋局中双方对每个棋子的控制权;第二步:通过控制权大小比较分析双方领地并按照公式得到territory值。此处只对基于queen走法的tq进行分析,tk同理。

针对棋盘上的任意一个空格,用queenmove来表示某一方通过queen走法走到这个格子需要的最小步数,即某一方四个棋子走到该格需要的四个最小步数中的最小值。计算双方的queenmove值,queenmove值小的一方获得对该格的控制权,此时该方较另一方需要更少的步数即可到达该格。

某一格对territory值的贡献分为五种情况:(1)双方到达该格的步数相同(包含都走不到的情况),该格贡献值记为equal,

equal为一个定值,因为这种情况下先行方占据主动权,所以equal取正值,笔者取为+0.5,可根据实际情况进行调整;(2)双方都可以到达,且执棋方的queenmove值小(到达所需步数少),执棋方对该格控制权更大,贡献值记为1;(3)双方都可以到达,且执棋方的queenmove值大(到达所需步数多),对方对该格控制权更大,贡献值记为-1;(4)执棋方可以到达,而对方无法到达,该格直接归属执棋方,贡献值记为2;(5)对方可以到达,而执棋方无法到达,该格直接归属对方,贡献值记为-2。

3.3 Position值

position值是在territory的基础上进一步计算对某个格子双方控制权的差值,代表通过控制权差异反映出的双方棋子的地理位置优劣,同样分为基于queen走法的p1和基于king走法的p2。此处不进行详细分析,直接给出参考文献^[3]中公式实现的具体代码。

```
for (int i = 0; i < n; ++i)
for (int j = 0; j < n; ++j)
//myQueenmove表示行棋方的, yourQueenmove表示对方的
if (myQueenmove[i][j] == yourQueenmove[i][j]) continue;
else if (myQueenmove[i][j] == inf) p1 += - pow(2.0, - yourQueenmove[i][j]);
else if (yourQueenmove[i][j] == inf) p1 += pow(2.0, - myQueenmove[i][j]);
else p1 += pow(2.0, - myQueenmove[i][j]) - pow(2.0, - yourQueenmove[i][j]);
p1 = 2 * p1;
for (i = 0; i < n; ++i)
for (j = 0; j < n; ++j)
p2 += min(1.0, max(-1.0, (double)(yourQueenmove[i][j] - myQueenmove[i][j]) / 6.0));
```

3.4 mobility值

mobility,意为灵活度,棋子的灵活度反映了棋子在八个方向上移动的灵活性大小。棋盘上的每个空格都有自己的灵活度值,而棋子的灵活度值取决于它能到达的空格的灵活度值。

文献^[4-5]中空格的灵活度的计算方式如下,针对某一空格,该空格通过queen走法走一步能到达的空格数记为它的灵活度。空格的灵活度值除以该棋子按照king走法走到该空格所需的最小步数(kingmove)得到该空格对该棋子灵活度值的贡献值,将棋子通过queen走法能走到的所有空格的灵活度贡献值相加,就得到了该棋子的灵活度值。一方四个棋子的灵活度值之和即该方的灵活度值,将双方灵活度值作差就得到了棋局评估所需的灵活度值。

由于在实际操作过程中,在每一局面下都要计算一遍棋盘所有空格灵活度和kingmove,花费的时间太多,于是我们将棋子灵活度的计算方式改为棋子按照queen走法走一步的所有可行招法数目,而评估函数中所需的灵活度值改为双方灵活度值的比值,其中对方(除数方)的灵活度值加上0.00001,用意在于防止对方棋子出现被堵死的情况导致除数为零,同时,使用比值而非差值,可以防止开局阶段灵活度值过大的情况,更能在出现对方被堵死的情况下使灵活度值很大,使行棋方直接选择这一招法将对方堵死。

4 评估函数的计算公式

评估函数的计算公式为 $value = k1(turnID) * tq + k2(turnID) * tk + k3(turnID) * p1 + k4(turnID) * p2 + k5(turnID) * mobility$, 其中 $k1 \sim k5$ 都是关于 $turnID$ 的一次函数, 用于给要素加权, 需要通过对局进行调参。

4.1 要素占比变化情况

$turnID$ 代表当前进行到的回合数, 随着对局的进行, 每个要素对棋局的影响会发生不同的变化。

开局阶段棋盘上主要都是空格, 障碍很少, 走法几乎不受限制, 此时双方对空格的控制权相差不大, 所以 tq 占比不是很大; 而此时需要开始圈地并尽可能防止对方棋子攻占我方棋子附近的格子, 因此, 评估我方棋子附近圈地能力的 tk 占比较大。但随着棋局向中局发展, 局面上的障碍越来越多, 双方对空格的控制权大小差异逐渐增大, tq 在整体估值中的重要性不断增加, 占比越来越大。

当棋局行进至残局阶段, 棋盘基本上已经被完全分割为多个独立的区域, 而棋盘上的空格都直接归属某一方, tq 的占比非常大, 由于此时领地已经不可能被对方进犯, tk 的占比降到最低。

$p1$ 、 $p2$ 反映的位置优劣在开局的时候占比较大, 因为这个阶段需要将棋下得分布均匀, 而当棋局进行到中局、残局, 棋盘已经被分割开, 不再需要考虑位置的优劣来使棋子分布均匀了。

$mobility$ 同样也是在开局占比大, 随着棋局进行, 局面分割, 所占比越来越小。

因此, 在棋局进行到一定程度 (通过 $turnID$ 衡量) 后, 为了计算方便, 我们直接将 tq 值当作整体的 $value$ 值。用于给棋局程度分界的 $turnID$ 值也可以通过不断尝试、观察对局并统计来得到。

4.2 具体实现

鉴于这些要素不同的变化情况, 考虑使用关于 $turnID$ (回合数) 的一次函数为要素进行加权, 具体数值通过统计对局结果调参可得。笔者在棋局上调参得到的具体公式如下:

```
if (turnID < 17) { // 在第1~16个回合内
    k1 = 2 * (32 + 2.0 * turnID/2);
    k2 = 1 * (32 - 1.8 * turnID/2);
```

```
    k3 = 1 * (32 - 1.8 * turnID/2);
    k4 = 2 * (32 - 2.0 * turnID/2);
    k5 = 0.5 * (32 - 1.8 * turnID/2);
}
else { // 在经过了16个回合之后
    k1 = 5;
    k2 = k3 = k4 = k5 = 0;
}
value = k1 * tq + k2 * tk + k3 * p1 + k4 * p2 + k5 * mobility;
```

而在具体实现过程中, 由于对局中出现了棋子在开局初期就走到边界的情况, 使得局面变得不利, 因此, 在开局、中局和残局等不同的阶段最好设置不同的参数。笔者通过在程序中强制设定开局几步中棋子不得走到边界处来解决这个问题。此外, 在设计界面上还可以自行添加各种功能, 如悔棋等。

5 结束语

本文完成了实现亚马逊棋的基本部分, 并较为深入地分析了其中非常重要的评估函数, 通过策略圈和策略堵两方面提出 $territory$ 、 $position$ 、 $mobility$ 几个要素, 更进一步地探讨了这些要素随着棋局进行的不同变化, 由此得出使用关于回合数的一次函数进行加权。同时, 所有给出的参数都只是来源于一小部分实验, 需要优化; 几个评估特征的具体确定也可以进一步完善, 可以考虑加入新的评估特征。

参考文献:

- [1] 张柳. 基于极大极小搜索算法的亚马逊棋博弈系统的研究[D]. 沈阳: 东北大学, 2010.
- [2] 乔治, 黄鸿. 亚马逊棋博弈技术研究[C]. 北京: 中国人工智能学会, 2013.
- [3] Jens Lieberum. An evaluation function for the game of amazons[J]. Theoretical Computer Science 349, 2005: 232-234.
- [4] 郭琴琴, 李淑琴, 包华. 亚马逊棋机器博弈系统中评估函数的研究[J]. 计算机工程与应用, 2012, 48(34): 51-53.
- [5] 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.

【通联编辑: 谢媛媛】