

亚马逊棋机器博弈系统中评估函数的研究

郭琴琴, 李淑琴, 包 华

GUO Qinqin, LI Shuqin, BAO Hua

北京信息科技大学 研究生部计算机学院 北京 100192

Department of Graduate, Beijing Information Science & Technology University, Beijing 100192, China

GUO Qinqin, LI Shuqin, BAO Hua. Research on evaluation function computer game of Amazon. Computer Engineering and Applications, 2012, 48(34): 50-54.

Abstract: Computer Game is the research carrier of artificial intelligence field. Amazon, as a quite new game, it is very suitable for the research of Computer Game. This paper, using Amazon's game system as an experimental platform, does research on the evaluation function which is one of the key technologies of Computer Game. Based on territory, position, mobility, the three main features of Amazon's assessment, and according to the Amazon game's different characteristics in Opening, Middlegame, Endgame, the paper analyses the evaluation factors' importance level and weight value in the various stages, then gets an evaluation function in form of stages. The experimental results show the evaluation function is feasible and effective.

Key words: Computer Game; Amazon game; evaluation function; game stage

摘 要: 机器博弈是人工智能学科研究的载体, 亚马逊棋作为一个相对较新的博弈棋种, 走棋特点介于围棋和象棋之间, 非常适合用来进行机器博弈研究。以亚马逊棋博弈系统为实验平台, 对机器博弈中的关键技术之一——评估函数进行了研究。以 territory、position、mobility 三个主要评估特征为基础, 根据亚马逊棋在开局、中局以及残局三个不同阶段的棋局特点, 分析了不同阶段中各评估因子的重要程度以及权重取值, 最后得到一个分阶段的评估函数。通过实验结果可知, 提出的棋局评估函数是可行并且有效的。

关键词: 计算机博弈; 亚马逊棋; 评估函数; 比赛阶段

文献标识码: A **中图分类号:** TP18 **doi:** 10.3778/j.issn.1002-8331.1108-0241

1 引言

博弈游戏是一种人类复杂逻辑思维活动的反映。机器博弈研究的主要内容是让计算机拥有人的思维去进行博弈游戏, 它的研究既简单方便、经济适用, 又包含丰富的内涵, 变化无穷的逻辑思维^[1]。所以就像果蝇被遗传学视为最理想的遗传学研究载体一样, 机器博弈也被称为人工智能学科研究的载体。

一个完整的机器博弈系统主要包括棋局表示, 着法生成器, 搜索引擎以及评估函数。棋局表示是指对比赛过程中形成的棋局的描述, 主要内容包括棋子、空格、箭、走棋; 着法生成器的作用是在已形成

的棋局下生成可行的着法, 这一部分的内容包括对下棋规则的描述并根据规则生成所有可行着法, 着法生成器是搜索对象的产生器。搜索引擎的目标是在可行着法中搜索最优着法, 这是机器博弈的核心部分, 是模拟人类思维的最佳体现。评估函数是用来对棋局进行评估的, 是搜索算法的前提, 评估函数的好坏直接影响搜索结果的准确性。

为了提高亚马逊棋机器博弈系统的性能, 本文从改进评估函数入手, 通过对亚马逊棋中 territory、position、mobility 三个主要评估特征的计算分析, 改进了 mobility 评估特征的计算方法, 并且根据阶段特

基金项目: 北京信息科技大学2011年研究生优质课程建设项目; 北京市属市管高等学校人才强教计划资助项目(No.PHR201007131)。

作者简介: 郭琴琴(1987—), 女, 硕士研究生, 主要研究领域为人工智能, 机器博弈; 李淑琴(1963—), 女, 博士, 教授; 包华(1988—), 男, 硕士研究生。E-mail: guoqinqin10@163.com

收稿日期: 2011-08-17 **修回日期:** 2011-12-07 **文章编号:** 1002-8331(2012)34-0050-05

点分阶段地设计三个评估特征的权重,得到一个改进的分阶段的亚马逊棋评估函数,最后通过实验结果验证了本文改进的评估函数有效地提高了博弈系统的性能。

2 亚马逊棋简介以及研究现状

2.1 亚马逊棋简介

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,是圈地游戏家族中的一员,和国际象棋是远亲。亚马逊棋的棋盘由 $n \times n$ (比赛时 10×10) 的格子组成,对弈双方各有 4 个皇后,棋盘初始摆放位置如图 1 所示,横坐标由 10 个大写英文字母 A~J 表示,竖坐标由阿拉伯数字 1~10 表示。当轮到一方走棋时,该方只能且必须移动属于它的四个皇后中的一个,每一个着法分两步完成,第一步是移动皇后,与国际象棋中皇后的走法相同,可以在八个方向上任意行走,但不能穿过阻碍(包括皇后和箭);第二步是放箭设障碍,在第一步移动完成后,在移动棋子的当前位置释放一个箭,箭的释放方法与棋子的移动方法相同,参见图 2,白方棋子的着法: $(1, D) \rightarrow (7, D) \rightarrow (7, B)$, 从 $(1, D) \rightarrow (7, D)$, 是完

成第一步移动皇后;第二步 $(7, D) \rightarrow (7, B)$ 是放箭设障碍,箭用黑色方格表示。在比赛的整个过程中,双方棋子按照下棋规则可以移动,但是箭一旦被设置在某处就不能再移动。当某方四个皇后均不能再移动时,则表示该方已经输掉了比赛。

2.2 亚马逊棋棋局评估研究现状

在机器博弈系统中,搜索算法直接影响博弈系统的性能。评估函数又是搜索算法的基础,要进一步提高搜索结果的准确性,除了提高搜索算法的搜索能力之外,更要完善评估函数。本文的主要工作就是对亚马逊棋评估函数的改进。

通过亚马逊棋的下棋规则可以看到,在亚马逊棋博弈中,博弈一方的最终目的是用本方的棋子以及箭将对方的棋子堵死,使其不能移动,由此产生了两种走棋策略。思路一是堵策略,即将对方棋子堵死在有限的区域里面,让对方无棋可下,这种策略进攻性强;思路二是占领土策略,我方棋子自行为自己圈定一个较大的领地,在这个领地里,对方棋子无法入侵,而我方棋子却有很大的自由活动范围,当对方棋子在域外未能占领足够大的领地时,也会因无棋可下而输掉比赛,这种策略注重防守。

这两种走棋策略思想体现在评估函数中,可以用两个评估特征表示,一个是占领地的特征 territory,另外一个为灵活度特征 mobility。关于这两个特征在 Jens.Lieberum^[2]以及乔治、黄鸿的文章^[3]中有很详尽的介绍。Jens.Lieberum 在他的文章^[2]中还考虑了皇后及箭的位置 position 对局面的影响,并提出一个可以描述比赛进度的计算公式 w , 最后给出基于比赛进度 w 的评估函数,评估因子的权重系数是由一系列与 w 相关的权重函数组成。按照 Jens.Lieberum 的方法,必须设计 5 个恰当的函数来作为相应评估因子的权重系数。现实中要优化这 5 个权重函数是很困难的,而且描述比赛阶段的 w 并不是一个万能的公式,它只适用于某些博弈系统下的棋局特点,不能通用在所有的博弈系统中。

3 亚马逊棋机器博弈系统评估函数的设计

在分析亚马逊棋博弈整个过程后,作者认为可以将亚马逊棋比赛过程划分成三个阶段,开局、中局以及残局,划分根据及划分方法请详见参考文献[4]。然后根据在不同的阶段,领地 territory、位置 position、灵活性 mobility 这三个特征对棋局的评估所起的作用即重要程度不同,设计出一个分阶段的评估函数:

$$value = a \times t_1 + b \times t_2 + c \times p_1 + d \times p_2 + e \times m \quad (1)$$

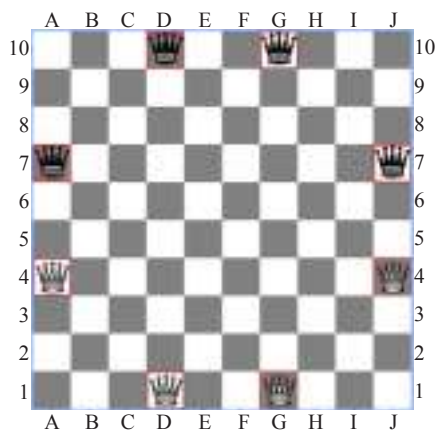


图1 亚马逊棋的初始棋盘

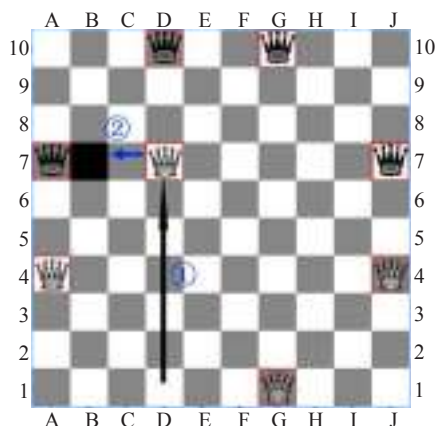


图2 亚马逊棋的走棋规则

其中 t_1, t_2 为 territory 特征值, 分别代表双方基于 Queen 以及 King 走法的对空格的控制权的评估值; p_1, p_2 为位置 position 特征值, 分别表示双方基于 Queen 以及 King 走法相对空格的位置优劣的评估值; m 是双方棋子的灵活性评估值; a, b, c, d, e 为相应的权重值。下面叙述这些值的计算方法。

3.1 territory 特征值的计算

参照文献[2], 对 territory 特征值的计算涉及两种走棋方式, 一个是 Queen 走法, 另外一个 King 走法。Queen 走法是指移动的八个方向上只要没有跨越障碍(障碍包括棋子和箭)就可以移动; 而 King 走法是指一次移动距离是 1, 也就是只能走步到相邻的空格。QueenMove 值代表对于一个空格(即没有棋子和箭), 某方的四个棋子通过 Queen 的走法走到此空格最少需要移动多少步; KingMove 值表示对于某一个空格, 某方的四个棋子通过 King 的走法走到此空格最少需要移动多少步。如图 3、图 4 所示, 图 4 左上方(10, A)空格内左上角的数值 2 表示白方通过 Queen 走法最少需要 2 步可以到达此处, 右下角的数值 4 表示黑方需要 4 步, 同样在图 4 中该空格中的数值 2 和 7 表示 King 走法时的情况。

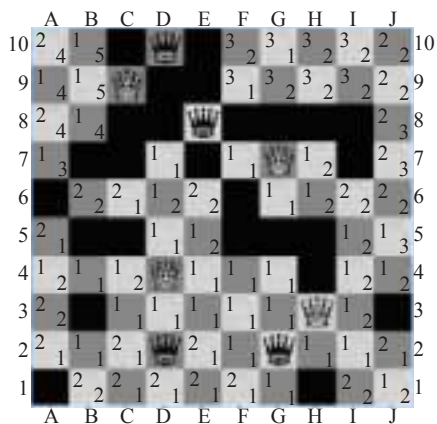


图3 KingMove 计算

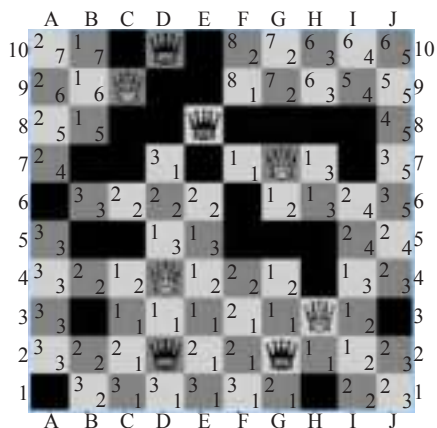


图4 QueenMove 计算

如上所述, 通过计算每一空格的 QueenMove 和 KingMove 值, 大体可以看出棋局的优劣。当白方棋子比黑方棋子用更少的步数到达某一空格时, 认为白方控制了该空格, 这样通过对全局每一个空格的计算, 可以算出每方控制的空格数目, 从而大体确定局势的优劣。下面给出计算公式:

$$t_i = \sum_a \Delta(D_i^1(A), D_i^2(A)) \quad (i=1 \text{ 或 } 2) \quad (2)$$

$$\Delta(D_i^1(A), D_i^2(A)) = \begin{cases} 0 & (D_i^1(A) = D_i^2(A) = \infty) \\ k & (D_i^1(A) = D_i^2(A) < \infty) \\ 1 & (D_i^1(A) < D_i^2(A)) \\ -1 & (D_i^1(A) > D_i^2(A)) \end{cases}$$

A 代表当前棋局中未被占领的空格, i 取 1 表示基于 Queen 走法, 取 2 表示基于 King 走法, $D_i^1(A), D_i^2(A)$ 的值分别表示白方、黑方到空格 A 的 QueenMove 及 KingMove 值。在第二种情形下, $D_i^1(A) = D_i^2(A) < \infty$ 时, 对该空格 A 的控制权由先走棋那一方决定, k 代表的是先行方的优势大小, 取值范围大于 -1 小于 1, 当先行方是白方时, k 取大于 0, 否则小于 0。计算结果 t_i 表示分别采用 QueenMove 或 KingMove 时, 白方和黑方对所有空格占有情况的综合评估得分。

3.2 position 特征值的计算

从 t_i 计算公式可以得知基于 territory 的评估仅仅反映了双方对空格的控制权归属, 但未能反映对空格的控制权差值大小以及控制权差值大小所反映的位置特征。实际上双方对一空格控制权的差值能反映出一方是否具有地理位置的优势。距离空格较近一方控制权大, 而距离空格较远的一方控制权小, 处于地理位置上的劣势。同样地, 这种地理位置优劣评估即对于 position 特征值的计算也涉及 Queen 和 King 两种走步方式, 可以利用 territory 特征评估中使用的 QueenMove 和 KingMove。参照文献[2], position 特征值 P_i 计算公式为:

$$p_1 = 2 \sum_A (2^{-D_1^1(A)} - 2^{-D_1^2(A)}) \quad (3)$$

$$p_2 = \sum_A \min(1, \max(-1, (D_2^2(A) - D_2^1(A))/6)) \quad (4)$$

在公式(3)中, P_1 值以 Queen 走步方式体现双方棋子在控制空格 A 上的位置优劣。当 $D_1^2(A) - D_1^1(A)$ 的值越大时, 相应的 $2^{-D_1^1(A)} - 2^{-D_1^2(A)}$ 值也相应大。 P_2 值以 king 走步方式体现双方棋子在控制空格 A 上的位置优劣。从公式(4)可知只有 $D_2^2(A) - D_2^1(A)$ 绝对值大于等于 6 时, 对该空格的控制权的位置优势值才能取得 1。

3.3 mobility 灵活度特征值的计算

在文献[2]中灵活度是用来衡量棋子向相邻八个方向移动的灵活性大小的。棋盘中的每个空格和棋子都有一个灵活度值,棋子灵活度的计算依赖于棋子可达空格的灵活度。图5中空格内的值代表该空格的灵活度值,而棋子皇后左上角的数字则表示该棋子在此棋局下的灵活度数值。空格的灵活度值为与该空格相邻的空格数。例如:图5中左上角(10,A)空格因为有3个相邻空格,所以该空格的灵活度值为3。棋子的灵活度值则与棋子采用 Queen 走法一步之内能到达的空格的灵活度值有关。下面是某个棋子 a 灵活度值的计算方法:

步骤1 计算棋盘上所有空格的灵活度值。

步骤2 记录棋子 a 采用 Queen 走法时一步之内能到达的空格。

步骤3 对步骤2记录的每个空格,计算空格的灵活度值除以棋子 a 采用 King 走法到达该空格的步数的值,然后将这些值累加,即得到当前棋子 a 的灵活度。

例如对于图5中位于左上角(8,D)位置的白色棋子,它的灵活度计算公式为: $F(a)=7/1+6/1+5/1+3/1+3/1+5/2+4/2+7/2+4/2+5/3\approx 35$ 。

	A	B	C	D	E	F	G	H	I	J	
10	3	5	5	4	4	4	4	3		2	10
9	5	8	7	6		6	6		6	4	9
8	5	7	5	35	3	41	5	7	7	5	8
7	5	7			3	35	6	7	7	4	7
6	4	5	3	3		4	4	4		2	6
5	4		33		4						5
4	4	6	4	4	2		0	0		2	4
3	4	6	5	29			2		5	4	3
2	4		31	5	3		4	7	7	5	2
1	2	3	3	4	3		3	5	5	3	1
	A	B	C	D	E	F	G	H	I	J	

图5 灵活度计算

按照以上算法,可以计算出双方所有棋子的灵活度,再根据公式(5),就可以对整个棋局上双方棋子灵活度态势做出评估。

$$m = \sum_w F(w) - \sum_b F(b) \quad (5)$$

其中 w 、 b 分别表示黑白双方的棋子, $F(w)$ 、 $F(b)$ 分别表示白方及黑方棋子的灵活度。如果 $m>0$ 时,表示该棋局中白方棋子灵活度大;当 $m<0$ 时,表示黑方棋子灵活度大。

这种通过计算棋子周围可达空格的灵活度,将这些空格灵活度的累加值作为棋子灵活度的计算方法首先要对棋盘上每个空格计算灵活度,每个空格

灵活度的求取也要通过棋盘的扫描,这个过程消耗的时间太多,在限制时间的比赛条件下,机器博弈系统会因为评估函数消耗时间过大而导致最后搜索结果的不理想,一个好的评估函数不仅要保证评估结果的准确性还要考虑评估的时间消耗。灵活度评估是为了保证棋子有着法可走而不被过早堵死,基于这个思想,文本提出一个更简单更省时的计算方法,即统计当前棋局双方棋子可行着法数目,将一方的可行着法数目作为该方的灵活度。

不管是文献[2]中描述的灵活度评估方法还是本文新提出的计算方法,都存在这样一个问题:开局阶段,经灵活度评估,一方的灵活度值比对方的灵活度要高很多,但是却出现了该方某个棋子过早被堵死的情形。为解决这个问题,本文在计算灵活度值的同时还计算了一个最小灵活度值并将该值加到灵活度评估特征里面,这个最小灵活度值是指一方四个棋子中的可行着法数目最小的那个值。加了这个值的灵活度评估可以防止一方某个棋子过早堵死的情形。

3.4 系数在比赛阶段中的重要性分析

以上 t_1 、 t_2 、 p_1 、 p_2 、 m 这五个评估因子都是针对棋局某一方面特征的评估,而亚马逊棋棋局特点在不同比赛阶段差异很大,所以五个评估因子的权重的取值不能固定不变而应该是随着比赛阶段而改变。

t_1 在整个比赛过程中的作用是越来越大,特别是进入中局、残局阶段。在比赛初期,走步比较少,设置的箭也相对较少,双方棋子对棋盘上空格的控制权还是比较均等的,所以 t_1 评估作用相对比较小,但随着比赛逐渐进入中局、残局阶段,每一个空格的控制权在哪一方越来越明显,而且进入残局阶段后,整个棋盘被划分成彼此独立的小区域,每个空格都只被一方棋子控制,这时候 t_1 的评估作用最大。所以 t_1 评估因子的权重在开局最小,中局变大,残局取最大。

t_2 在整个比赛过程中的作用越来越小,而进入残局阶段后,作用几乎为零。虽然 t_1 和 t_2 都是对空格控制权的评估,但是因为评估时基于的走步方式不一样,他们在各阶段的重要性也不一样。 t_2 基于 King 走法的评估,这个评估值的作用是保证我方棋子附近的区域圈地并防止对方棋子进攻我方棋子附近的领地,在开局阶段时这点非常重要,所以在 t_2 的评估值在开局阶段比 t_1 更重要,而进入残局阶段,任何一块空格领域只被一方棋子占领,领地也被圈分完,也不存在阻止对方棋子进攻的问题。所以 t_2 评估因子的权重取值应是开局阶段最大,中局次之,

残局时权重可以取很小或者忽略。

c_1 和 c_2 的作用也体现在开局阶段, c_1 和 c_2 是基于双方棋子相对目标空格位置的评估, 涉及到棋子的分布状况, 棋子的分布状况只有在开局阶段比较重要, 这个时候比赛双方棋子要尽量在整个棋盘上均匀分布, 实验发现当他们的权重大的时候, 整个棋盘上的棋子分布比较均匀。在残局, 棋局分布已成定局, 不用考虑棋子位置的分布情况, 所以残局不用考虑这个评估因子。

m 在比赛整个过程中的重要性也越来越小。通过灵活度因子的分析知, mobility 评估因子在比赛初期作用最大。一般情况下, 一个棋子在比赛过程中越早被堵死, 对该方越不利。在到了中后局, 已过了布局阶段, 棋子的分布已成明朗, 棋子的灵活度对于棋局的影响越来越小了。所以 m 评估因子的权重在开局最大, 中局次之, 残局不考虑灵活度了。

4 评估函数的算法实现

本文亚马逊棋的评估函数的计算算法为:

假设我方为白方, 输入当前棋局(用一个 10×10 的二维数组表示)。具体算法如下:

步骤1 对棋盘上的每个空格计算 QueenMove、KingMove 值。

步骤2 对棋盘上的每个棋子计算 mobility 值。

步骤3 按照计算公式(2)(3)(4)(5)分别求出 t_1 、 t_2 、 p_1 、 p_2 、 m 值。

步骤4 根据公式(1)计算 value 值。当 $value > 0$ 时, 表示该棋局对我方有利; 当 $value < 0$ 时, 表示对黑方有利, 值越大代表优势越大。

具体实现, 设置前 20 步为开局阶段, 21~49 为中局阶段, 50 步以后为终局阶段。通过实验得到各阶段评估因子的权重取值如表 1 所示。

表1 评估因子权重系数取值

评估特征 评估因子 权重系数	Territory		Position		Mobility
	t_1	t_2	p_1	p_2	m
	a	b	c	d	e
开局	0.14	0.37	0.13	0.13	0.20
中局	0.30	0.25	0.20	0.20	0.05
残局	0.80	0.10	0.05	0.05	0.00

将该评估函数应用到一个性能稳定的亚马逊棋博弈系统中, 取名为“分阶段评估函数”, 分别将它与未考虑阶段特点的系统, 这里称为“原始固定权重评估”, 以及文献[2]中使用的 w 阶段描述, 这里称为“含 w 描述评估函数”的系统进行对弈, 每对程序分先手

和后手, 每次对弈 20 局, 考虑到对弈时间因素, 根据每步走棋时间 T 取值得到以下对弈结果(表 2~表 4)。实验结果表明, 分阶段评估函数的博弈系统要优于原始权重评估及含 w 描述评估函数的系统, 评估时间越短, 使用分阶段的评估函数, 系统性能相对后两者优势更大。实验还统计了使用三种不同评估函数的系统在搜索时间 T 内评估的平均节点数目以及搜索深度, 发现使用本文提出的评估函数的系统评估的节点数目最大并且博弈树搜索深度更大, 这是导致“分阶段评估函数”性能优于其他两个系统的根本原因。

表2 实验结果($T=30$ s)

先手VS后手	Score
原始固定权重评估VS分阶段评估函数	6:14
分阶段评估函数VS原始固定权重评估	14:6
含w描述评估函数VS分阶段的评估函数	10:10
分阶段的评估函数VS含w描述评估函数	14:6

表3 实验结果($T=20$ s)

先手VS后手	Score
原始固定权重评估VS分阶段评估函数	4:16
分阶段评估函数VS原始固定权重评估	15:5
含w描述评估函数VS分阶段的评估函数	8:12
分阶段的评估函数VS含w描述评估函数	14:6

表4 实验结果($T=10$ s)

先手VS后手	Score
原始固定权重评估VS分阶段评估函数	2:18
分阶段评估函数VS原始固定权重评估	17:3
含w描述评估函数VS分阶段的评估函数	3:17
分阶段的评估函数VS含w描述评估函数	18:2

5 结论

本文以 territory, position, mobility 为主要特征, 结合亚马逊棋的阶段特点, 设计并实现了一个分阶段的评估函数, 虽然通过实验验证了本文提出的评估函数是可行的, 并且一定程度上能提高亚马逊棋的博弈系统的性能, 但是许多参数来自实验值, 还需进一步优化。下一步拟采用机器学习方法, 进行评估因子权重取值问题的研究, 另外除了考虑上面三个主要特征外, 评估函数还将考虑加入新的评估特征。

参考文献:

[1] 徐心和, 徐长明. 计算机博弈原理与方法学概述[C]//中国人工智能进展, 2009: 1-13.
[2] Lieberum J. An evaluation function for the game of amazons[J]. Theoretical Computer Science, 2005, 359(2): 230-244.

(下转 87 页)