

六子棋博弈的评估函数^{*}

张小川, 陈光年, 张世强, 孙可均, 李祖枢

(重庆理工大学 人工智能系统研究所, 重庆 400050)

摘 要: 针对六子棋博弈问题, 提出了一种“路”的博弈策略, 构建了一种六子棋博弈的改进模型。又针对六子棋博弈评估函数参数优化中存在的问题, 改进了六子棋博弈评估函数, 并应用遗传算法进行寻优处理。实验结果表明: 改进后的评估函数能够保持遗传种群的多样性, 避免了遗传过程收敛过快, 是一种优化机器博弈评估函数的有效办法。

关 键 词: 六子棋; 机器博弈; 评估函数; 遗传算法

中图分类号: TP39

文献标识码: A

文章编号: 1674 - 8425 (2010) 02 - 0064 - 05

Research on Evaluation Functions for Computer Game of Connect6

ZHANG Xiao-chuan, CHEN Guang-nian, ZHANG Shi-qiang, SUN Ke-jun, LI Zu-shu

(Institute of Artificial Intelligence System, Chongqing University of Technology, Chongqing 400050, China)

Abstract: A game strategy of 'road' for connect6 game is proposed, and an improved model for connect6 game is constructed in this paper. After that, the evaluation function of connect6 game is improved for the problems existing in the parameter optimization of connect6 game, and then, genetic algorithm is used to optimize parameters. The results of experiments show that the improved evaluation function can keep the diversity of population, avoiding the early convergence, and it's an effective method to optimize the Evaluation Function in computer games.

Key words: Connect6; computer game; evaluation function; genetic algorithm

机器博弈是人工智能的果蝇, 即与果蝇在遗传学中的研究所起的作用相似^[1]。六子棋是台湾交通大学吴毅成教授发明的一种新型计算机博弈平台^[2], 其产生之初是为解决五子棋的公平性和因“禁手”而严重影响其娱乐性的问题。六子棋与五子棋最大的不同点就是行棋规则的不同, 即六子棋规定黑方先手可以在 19 路围棋棋盘上的任意点落下 1 颗棋子, 然后白方、黑方依次每轮连续

落下 2 颗棋子。这样的规则就最大限度地侵消了黑方的先手之利, 从而不设“禁手”, 极大提高了娱乐性和公平性^[3]。目前, 六子棋的研究正受到国内研究人员的广泛关注。

1 六子棋博弈系统

按照人工智能一般的研究方法, 模仿人类的

* 收稿日期: 2009 - 11 - 08

基金项目: 重庆市教委科技项目 (KJ080621); 重庆理工大学大学生科技创新项目

作者简介: 张小川 (1965—), 男, 四川邻水人, 教授, 主要从事人工智能、智能机器人、计算机应用等方面的研究; 通讯作者 李祖枢 (1945—), 男, 重庆人, 教授, 博士生导师, 主要从事智能控制理论及应用、人工生命、计算智能、智能机器人及模式识别与图像处理方面的研究。

思考方式和行棋流程,可将六子棋博弈系统划分为开局和中局 2 个阶段和搜索引擎、走法生成、评估函数、开局库 4 个组成部分^[4],从而构成图 1 所示的六子棋博弈流程。

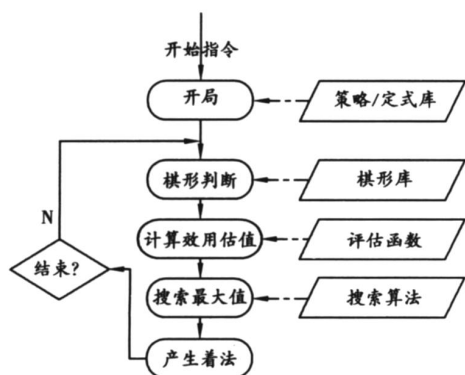


图 1 六子棋博弈系统的博弈流程

在图 1 中,开局阶段中棋盘棋子少,行棋可以通过查询策略库、定式库快速发现着法。所谓定式就是相对固定的行棋下法,通常是被证明了的经典的行棋套路。因此,开局阶段固然重要,但是它对博弈系统整体棋力的影响度相对较小。开局阶段与中局阶段的分界点在不同的博弈系统中是不同的,本文中的分界点是按照棋盘上的棋子数多少来划分的,即以 9~11 颗棋子为分界点。按照六子棋规则,黑方在围棋棋盘任意位置落下 1 棋子后,白方连续落下 2 棋子、之后黑方再连续落下 2 棋子,依次类推,因此,棋盘上存在双方共 9~11 颗棋子的情况,即双方的第 1~3 手棋。定义双方的这 3 手棋为开局,按照开局定式库的套路行棋。从第 4 手棋开始,棋局进入中局阶段。由于六子棋规则中没有象棋中的吃子着法,因此,棋盘上棋子将越来越多,这样以来,六子棋就没有残局之说了。

在中局阶段,棋盘上棋子越来越多,棋形越来越复杂,其计算量也逐渐增大,对棋形、棋局状态的准确判断越来越困难。而准确判断棋局的好、坏状态,是产生下一步着法的基础,完成棋局状态评估工作的就是棋局评估函数。因此,评估函数是六子棋博弈系统进行推理、判断的基础性函数,其研究受到了机器博弈爱好者的极大关注^[4-7]。

本文中针对评估函数,提出一种“路”的策略思想,改进了文献[3]中的六子棋博弈系统模型和评估函数,利用遗传算法对评估函数进行了离线优化。实验证明优化结果是可行的。本文中尽管是针对六子棋的研究,但对其他机器博弈系统的研究也具有积极的借鉴意义。

2 基于“路”的六子棋博弈的决策

目前,大部分六子棋博弈系统的结构都是基于棋形分类的结构,这种结构对于准确判断棋形的要求非常高,因此,有效和准确地判断棋形就成为衡量系统棋力高低的一个决定性因素。六子棋常见棋形有 19 种^[4],如活三、眠三、跳三、活四、眠四、跳四等,每种棋形又存在多种情况,且还存在这些棋形交叉和不好分割的情况,如图 2 所示。所以,如何有效和准确地判断、搜索棋形,已成为六子棋机器博弈系统的难题。

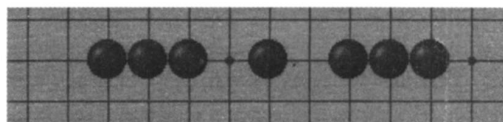


图 2 六子棋博弈的一种交叉棋形

为破解这个难题,本文中拟采用“路”的思想,构建六子棋的决策模型。所谓“路”就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,这样对棋形的判断就变得非常简单。例如,某“路”中已存在 4 颗子,就不用再去判断它到底是活四、眠四、跳四等棋形。在六子棋博弈系统中,寻找这样的“路”就如黑暗中人们看见灯塔一样,指明了前进的方向,极大地减少搜索路径数量。同时,利用面向对象程序设计思想,将“路”定义为类,就可以包括很多的属性,如代号、颜色、是否有效等信息,能方便地予以实现。比如,通过对“路”中同种颜色棋子数目的判断,即可获知该“路”是否有效。

“路”的总数较少。按照横向、纵向、左斜、右斜 4 个方向的特点和“路”的定义,可以分别计算不同方向的路数目(见图 3),计算规则: 横向

上, 19行 $\times (19 - 6 + 1)$ 路/行 = 266路; 纵向上, 19列 $\times (19 - 6 + 1)$ 路/列 = 266路; 左斜上, 14行 $\times (19 - 6 + 1)$ 路/行 = 196路, 如图 3; 右斜上, 14列 $\times (19 - 6 + 1)$ 路/列 = 196路。因此, 在 19×19 围棋棋盘上, 总共有 $266 \times 2 + 196 \times 2 = 924$ (路)。采用“路”的决策思想, 六子棋博弈系统就只需要根据“路”来估值, 计算出棋盘的状态值, 而不用对棋形进行判断。但是, 已经存在不同棋子数的若干条“路”, 究竟哪条“路”更优, 以及那条“路”是直通杀着或败着呢? 这又返回到对棋局状态的评估、判断问题上, 实质上就是棋局评估函数的建立与优化问题。通常, 对评估函数的优化问题可以采用经验加人工调节方式进行, 也可以采用智能算法寻优的方法, 本文中选择了在人工经验基础之上的遗传算法离线寻优方法。

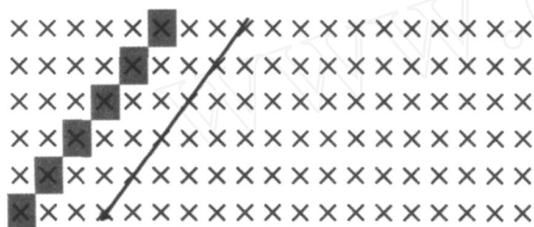


图 3 斜向“路”的形成示意图

3 基于遗传算法的六子棋棋局评估函数

3.1 评估函数的构造

采用“路”思想, 对棋局状态的评估就不再进行棋形的判定, 只需要对博弈双方“路”的情况进行计算。本文中将建立的评估函数由 EvaluateChessStatus(EvaluateColor) 和 CheckChessStatus() 2个部分组成。

1) CheckChessStatus() 初始化函数。该函数的作用是在估值前先对当前局面进行状态扫描, 洞察当前局面中含有本方 1~6 颗棋子的有效“路”数, 假设用 $NumberOfMyRoad[1] \sim NumberOfMyRoad[6]$ 表示之, 而对方的有效“路”数假设用 $NumberOfEnemyRoad[1] \sim NumberOfEnemyRoad[6]$ 表示之。

2) EvaluateChessStatus(EvaluateColor) 对估值

函数。通过分析发现, 总共 924 条“路”可以被划分为 7 种状态, 即在“路”上分别有 0~6 颗棋子的 7 种状态。假设分别用 $ScoreOfRoad[0] \sim ScoreOfRoad[6]$ 表示不同状态估值, 该值越大, 则所对应的“路”的威胁就越大。采用“路”思想, 行棋中只要关注某路上的棋子数, 而不需要关注这些棋子的棋形排列情况。比如, 某条路存在 4 颗棋子, 它可能是活四或跳四, 但采用基于“路”的搜索策略, 就不用关心是活四还是跳四, 只关心还差 2 颗棋子分胜负, 且在该路上只要对方再下一棋子, 该路就不可能再出现连 6 局面, 按照路的定义, 它不是路, 将自动被排除在搜索范围外。

根据上述思想, 对具体棋局的估值 Score 可由式 (1) 计算获得:

$$Score = \sum_{i=1}^6 (NumberOfMyRoad[i] * ScoreOfRoad[i]) - \sum_{i=1}^6 (NumberOfEnemyRoad[i] * ScoreOfRoad[i]) \quad (1)$$

由式 (1) 可知, $ScoreOfRoad[i] (i=0, 1, \dots, 6)$ 就是 7 种状态“路”的分量值。因此, 评估棋局状态就是评估这些分量值。只有式 (1) 中各参数准确, 才能正确反映棋局状态。假设“路”中无棋子, 则分量值 $ScoreOfRoad[0] = 0$; 假设“路”中存在 6 颗棋子, 则分量值 $ScoreOfRoad[6] = 10\ 000$ 。当然, 采用经验赋值也是可行的, 但主观色彩太浓。因此本文中采用遗传算法进行离线寻优。

3.2 确定遗传种群

遗传算法是一种借鉴生物界“适者生存, 优胜劣汰”遗传机制的智能算法, 其特点是直接对结构化对象进行操作, 并被广泛应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域^[8-9]。根据六子棋博弈和遗传算法的特点, 本文中采用遗传算法来实现评估函数的优化过程。

遗传算法包括选择、交叉和变异 3 个基本操作。其中种群的确定是前提, 对遗传结果的影响是不容忽视的。实际上, 初始种群如果选择不当, 将导致迭代次数增加, 进化时间加长, 甚至导致遗传算法只能获得局部最优解。如果采用随机生成初始种群的方法, 其遗传算法运算费时, 从而使得六子棋博弈的寻优过程非常长, 不能满足对抗性

需要。因此,本文中不采用初始种群随机选择的方式,而借助人工经验来确定初始种群。首先根据人工经验赋式(1)中 $ScoreOfRoad[1] \sim ScoreOfRoad[5]$ 的 5 个初始值,后设定 20% 的浮动范围,再在这些基础上“随机”生成初始种群。该方法的优点是缩小了最优解范围,提高了求解速度。

3.3 计算适应度函数

一般而言,运算中的遗传算法基本不需要外部信息,以适应度函数为依据即可进行搜索。此处适应度函数是由目标函数变换而来。由于遗传算法是针对结构化对象的,因此可以构造比较准确的目标函数。但对棋类博弈系统来说,评价棋局状态的参数复杂,结构化困难,其目标函数就很难构造和处理。本文中在文献[4]中锦标赛算法基础上,改进种群适应度函数的构造步骤。

1) 将包含个体 $individual[i]$ ($i = 1, \dots, m$) 的种群随机分成 n 个小组 $group[j]$ ($j = 1, \dots, n$)。初始时,假设个体适应度 $fitness[i] = 1$ ($i = 1, \dots, m$),先在单个小组 $group[j]$ 内部进行循环赛,每个个体都要与自己所在小组内的其他每一个个体进行交换先后手的 2 局对抗,其适应度值的计算规则是胜方加 1,败方加 0,平局为双方各加 0.5。待 n 个小组的内循环赛都结束后,将 n 个小组中每小组的最优个体提取出来,并分别记录为这 n 个小组内最优个体的适应度为 $best_fitness[j]$ ($j = 1, \dots, n$)。

2) 将各小组的最优个体组成一个精英个体种群,命名为精英小组。在精英小组进行循环赛,并分别记录精英小组中 n 个精英个体的适应度为 $elite_fitness[j]$ ($j = 1, \dots, n$)。

3) 确定包含 m 个个体 $individual[i]$ ($i = 1, \dots, m$) 的种群的适应度值由式(2)求得:

$$fitness[i] = \frac{fitness[i] \times elite_fitness[j]}{best_fitness[j]},$$

$$i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (2)$$

式(2)中: i 为个体编号; j 为该个体所在的小组编号。

3.4 交叉与变异操作

交叉的方式很多,但是由于均匀交叉更加广义化,将每个参数都作为潜在的交叉点,因此,仍

然选择均匀交叉。均匀交叉根据交叉率 P_c 随机地产生与参数个数等长的 0~1 掩码,掩码中的片段表明哪个父个体向子个体提供的变量值。利用掩码和此父个体确定了子个体。例如,假设 2 父个体为 {12, 88, 201, 771, 900} 和 {3, 65, 199, 800, 1000}, 根据交叉率 P_c 随机生成的掩码为 01001, 则交叉后的子代分别为 {12, 65, 201, 771, 1000} 和 {3, 88, 199, 800, 900}。事实上,在交叉过程中,交叉率 P_c 的大小是影响交叉效果、遗传算法行为和性能的关键,它将直接影响算法的收敛速度。为获得满意的收敛速度,此处引用自适应遗传算法,采用动态方式调整交叉率,且动态调整交叉率计算公式定义式为:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f - f_{avg})}{f_{max} - f_{avg}}, & f > f_{avg} \\ P_{c1}, & f < f_{avg} \end{cases} \quad (3)$$

式(3)中: $P_{c1} = 0.9$; $P_{c2} = 0.6$; f_{max} 为群体中最大的适应度值; f_{avg} 为每代群体的平均适应度值; f 为两交叉个中较大的适应度值。

变异是根据一个相对较小的概率(设变异率为 P_m)来改变一个或几个基因。变异是一种局部随机搜索,与选择/交叉算子结合在一起,保证了遗传算法的有效性,既使遗传算法具有局部的随机搜索能力,同时又使得遗传算法保持了种群的多样性,防止出现非成熟的收敛^[10]。在本文中构造的系统中,先以一定概率来判断某基因(设为 old_chrom)是否发生变异,如果发生变异,则先随机生成一个 $[-0.15, 0.15]$ 随机数,该随机数表示变异程度,用 $rate$ 表示。然后,按式(4)计算变异后的基因(new_chrom):

$$new_chrom = old_chrom \times (1 + rate) \quad (4)$$

例如,若变异前基因为 100, $rate = -0.11$, 则变异后的基因为 $100 \times (1 - 0.11) = 89$ 。

4 实验

为比较 2 组参数的优劣,分别使用各组参数“武装”六子棋博弈系统。除参数不同外,保持系统的其他内容不变,并按如下步骤进行对比:

步骤 1 采用文献[6]中提出的策略进行寻

优。实验表明,在第 10 代完成收敛,且第 10 代的最优个体为 {14, 66, 153, 790, 844}。

步骤 2 采用本文中提出的策略进行寻优。实验表明,在第 20 代,发现子群只在一定范围内改变。为保持种群的多样性,对策略进行修改,每次产生子代种群时,将随机生成的若干个体纳入子群,保持下代种群的多样性。采用步骤 1 相同的初始种群,经过 20 代,获得优化参数为 {17, 78, 141, 788, 1030}。

步骤 3 将步骤 1 和步骤 2 分别得到的 2 组参数组进行博弈对抗,结果采用本文中改进后的策略能取得 70% 以上胜率。

通过上述实验验证和实战性对抗,证明本文中提出的策略和算法保持了种群的多样性,避免了以前算法收敛过早的问题,改进效果明显、有效。尽管本文中的研究对象是六子棋机器博弈系统,但其思想对其他机器博弈系统也有借鉴意义。

参考文献:

- [1] 徐心和,邓志立,王骄,等. 机器博弈研究面临的各种挑战 [J]. 智能系统学报, 2008 (8): 288 - 292
- [2] 吴毅成. 一种新的挑战性疾病——六子棋 [C]// 2006 中国机器博弈学术研讨会论文集. 北京: [出版者不详], 2006: 7 - 16
- [3] 张颖,李祖枢. 棋类计算机博弈系统的主要研究方法及其在六子棋上的应用 [J]. 重庆工学院学报: 自然科学版, 2008, 22 (9): 98 - 102
- [4] 张小川,舒良. 关于六子棋计算机博弈策略的探讨 [C]// 中国人工智能进展. 武汉: [出版者不详], 2007.
- [5] 王骄,王涛,罗艳红,等. 中国象棋计算机博弈系统评估函数的自适应遗传算法实现 [J]. 东北大学学报, 2005, 26 (10): 949 - 952
- [6] 李果. 基于遗传算法的六子棋博弈评估函数参数优化 [J]. 西南大学学报: 自然科学版, 2007, 29 (11): 138 - 142
- [7] Marsland T A. Computer chess and search [D]. Edmonton: University of Alberta, 1991.
- [8] 徐志伟,张耀坤,张勇,等. 求解旅行商问题的一种改进遗传算法 [J]. 四川兵工学报, 2009, 30 (1): 74 - 75.
- [9] 杨华芬. 一种改进的遗传算法在函数优化中的应用 [J]. 重庆工商大学学报: 自然科学版, 2009, 148 - 151.
- [10] 王小平,曹立明. 遗传算法理论、应用与软件实现 [M]. 西安: 西安交通大学出版社, 2002: 18 - 55

(责任编辑 刘 舸)