ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.7, No.10, April 2011,pp.2310-2312

E-mail: xsjl@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

计算机博弈之六子棋的主要技术分析

刘雅靖

(大连交通大学 软件学院,辽宁 大连 116021)

摘要:该文主要介绍了六子棋博弈的四个主要部分:状态表示,走法生成,评估函数以及搜索算法,并分析了当前的主要技术及其优缺点。而且对搜索算法进行了一定的优化,为计算机博弈研究提供了一定的参考。

关键词:计算机博弈;六子棋;数据结构;走法生成;搜索算法

中图分类号:TP311 **文献标识码**:A **文章编号**:1009-3044(2011)10-2310-03

Main Technologies Analysis of Connect6 in Computer Game

LIU Ya-jing

(Dalian Jiaotong University, Software Colledge, Dalian 116021, China)

Abstrct: This article mainly introduced the four main parts of Connect6:state representation, move generation,evaluation function, searching algorithm, and analyzes the main teconology in current and their advantages and disadvantages. Also I optimized the searching algorithm, and provided a certain reference for the Computer Game.

Key words: computer game; connect6; data structure; move generation; searching algorithm

计算机博弈是智能性行为游戏,五六十年代,它一度是人工智能的带头领域,至今在人工智能界仍非常受重视。计算机博弈种类很多,像国际象棋,围棋,中国象棋等,研究相对成熟。而六子棋是2005年台湾交通大学资讯工程系的吴毅成教授提出的一系列K子棋中的一种。由于其特殊性,每次每方走两颗棋子,直观看,其状态空间复杂度及博弈树的复杂度会成倍提高,因此以六子棋作为研究平台,不仅能促进六子棋这项活动,而且能为计算机博弈技术带来新的发展。

六子棋(Connect6),规则是第一手黑方先下一颗子,第二手开始,双方轮流每次下两子,在横向,竖向,斜向先连成六子或六子以上为赢,若全部棋盘下完仍未分胜负,判和。

由于六子棋刚兴起不久,国内外研究相对较少,但是作为计算机博弈的一种,它与象棋围棋的研究有一定的相似处,因此它的主要技术要点也包括这些项:状态表示,走法生成,评估函数,博弈树搜索等,下面我就分析一下。

1 状态表示

我们在用计算机解决任何实际问题之前,必须要将具体问题表示成计算机能够处理的数据。在这里我们需要将六子棋的棋局状态,棋局的变化以及棋局局部特征低占用率高速度的表示出来。

下面有两种方法我们可以尝试来表示棋局状态:

1.1 矩阵表示法(又叫数组表示法)

六子棋棋盘通常为19×19,因此可以用二维数组Board[19][19]表示。当然也能用一维数组表示,占用空间361B。

1.2 比特棋盘表示法

在高性能的博弈程序里,比特棋盘被广泛应用,在六子棋中,棋盘上每个交叉点用2个bit位来表示,即00为黑,01为白,10为空,11其他,每个横线上19个点共需38bit,用一个8B的长整型来表示,那19行就有19×8B,即152B,比数组表示法占的空间要小。

六子棋的局部特征表示方法也有很多种:

1.3 用字符串结构来表示特征

我们用"OO++OO"来表示,其中"O"代表有棋,"+"代表空位,采用字符串模式匹配算法来提取局部特征,这种方式很直观,但没有区分黑白棋,因此特征不明显,而且取一种特征需遍历整个棋盘,运行速率很低。

1.4 用一个状态数组来表示特征

我们以当前棋子位置为中心,分别向相反的方向定长扫描6个点,得到长度为13的状态数组,而每个数组元素有四个可能状态:黑,白,空或棋盘外,如图1。

这种方法棋型种类巨大,数据冗余严重,尽管简单直观。

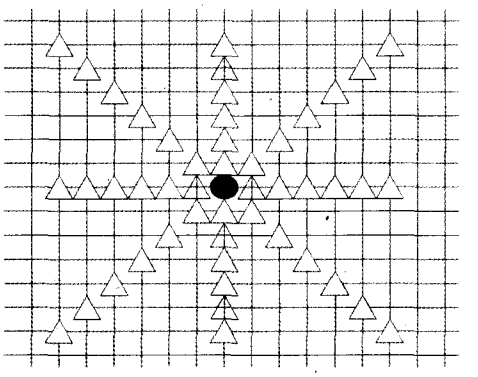

图1 棋子分布信息

1.5 一种新的6-8窗口法

收稿日期:2011-02-26

作者简介:刘雅靖(1984-),女,硕士研究生,研究方向为人工智能。

经过对六子棋深入研究,有人提出 6-8 窗口法,如图 2。

图 2 6-8 窗口法获取棋子分布信息

6 窗口主要功能是提取棋型信息并判断是否同色,8 窗口主要功能用于辅助判断是否存在迫著点,取完一个棋型后,窗口向前移动,然后提取下一个,直到遍历完整个棋盘。该方法优点是遍历完一次可统计完一种棋型的数量,完成局面估值,而且棋型少很多。但是不够直观,代码编写也较困难。

所以一种好的状态表示方法对加快搜索效率非常重要。

2 走法生成

走法生成就是将所有可能的走法罗列出来,并告知下一步走哪。

走法生成过程中常伴随着搜索进行,因此好的走法生成,可以提高搜索效率。走法生成主要有这几种:棋盘扫描法,模板匹配法,预置表法以及这些方法的结合。

棋盘扫描法:就是在棋盘内反复搜索有效区域,制约条件以及落子状况,从而确定有哪些地方可以落子,这种方法时间开销巨大。

模板匹配法:我们可以为一些动子设计模板,当选定动子,那么它的提址和落址便相对固定,只要匹配到提址,便可迅速找到落址。

预置表法:简单的说,就是把所有可能的走法预先存储起来,在生成走法时,通过查表来取代计算,即空间换时间。

在走法生成并进行搜索时,有两种方式,一种是生成一种走法就马上进行搜索,即深度优先;另一种是生成所有可能的走法然后进行搜索。经研究,第二种方法的效率更好。

3 评估函数

评估函数是模式识别和智能算法应用最广泛的领域。不管评估函数多复杂,都能表示为一个多项式。评估函数一般来说必须包括 5 个方面:固定子力值、棋子位置值、棋子灵活度值、威胁与保护值、动态调整值,每一方面的值又是由许多参数值构成的。即使最简单的评估函数也有 20 多个参数植,将其线性组合在一起得到最终评估值。

对于六子棋,它无子与子的差别,所以其评估函数只跟当前棋子的落子位置及周边棋子位置状态有关。由此可见六子棋的评估函数是 5 要素的一个子集,从而得到了简化。进行估值有以下原则 :棋型影响大,估值要高;威胁性小,估值要小;一方得胜,估值最大;对称数组,估值相同。

根据以上估值原则,棋型特征的 stepValue 值都人为的给出,但这依赖于编程者的知识和经验,尽管遵循原则,但给出的值仍然不够准确,所以我们最好通过遗传算法来对参数值进行调整和优化。

1) 参数选取和编码

首先我们人为给出参数表(假定是威胁值的参数)如图 3:

己方单 4	对方单 4	己方活 3	对方活 3	单 3	活 2	单 2	活 1	单 1
A1	A2	A3	A4	A5	A6	A7	A8	A9

图 3 参数表

每一个参数都通过二进制编码,设 A1-A9 由 10 位二进制位组成。

2) 适应度函数的计算

棋盘大小为 19×19,也就是说棋盘共有 361 个着棋点,如果一场比赛回合数为 x,则这场比赛该个体的得分为:

$$f=\begin{cases}361+\frac{1}{k}(361-x)^2, & \text{己方获胜}\\ x, & \text{对方获胜;}\\ 361, & \text{和棋;}\end{cases}$$

其中 K 为可调系数,取 6,个体得分总和设为适应度值。

3) 交叉和变异操作,在这里我们用最简单的单点交叉。

4) 自适应遗传算法:交叉概率 Pe 突变概率 Pm,按如下公式自适应调节:

$$Pe=\begin{cases}K_1\frac{f_{max}-f'}{f_{max}-f_{org}}, & f'\geq f_{org};\\ K_2, & f'\leq f_{org};\end{cases}\qquad Pm=\begin{cases}K_3\frac{f_{max}-f}{f_{max}-f_{org}}, & f\geq f_{org};\\ K_4, & f\leq f_{org};\end{cases}$$

其中 f_{max} 是群体中最大适应度值,f_{org} 是群体的平均适应度值,f 是交叉个体中较大者的适应值,f 是突变个体的适应度值,其中 $k_1k_2k_3k_4$ 在[0,1]范围内取值。

在这只简单的说明了一下遗传算法对参数进行优化的过程。

4 博弈树搜索及优化方法介绍

我们知道象棋,六子棋等类型的博弈都可用博弈树描述,因此搜索就是寻找最佳解的过程。

博弈搜索算法研究已有半个多世纪,有一定的理论基础。其中最大最小值算法是所有算法的基础,目前这一领域的算法有两大类,一种是深度优先的 Alpha-Beta 算法及一些强化算法,它几乎成为主流,另一类是最佳优先的系列算法。

1) 极大极小值算法

极大极小植的基本思想是假定我们始终站在一方(MAX 方)的立场上,MAX 将选择子节点中估值最大最有利的一个,当 MAX 确定后,另一方(MIN 方)会选择最估值最小的节点,因为对 MAX 最不利,就这样最大最小的选择就是此算法的核心思想。

2) 负极大值算法,基本思想与极大极小值算法相似。

3) Alpha-Beta 算法

① Alpha-Beta 剪枝

在极大极小值搜索中,要遍历整个博弈树,会造成冗余,所以要剔除一些分枝,Bruno 在 1963 年首先提出了 Alpha-Beta 剪枝算法。

α 值为倒推值下界,β 值为倒推值上界。Alpha-Beta 搜索首先使某一部分达到最大深度 ,从而计算出 Max 节点的 α 值,Min 节点的 β 值,然后继续搜索,不断修改这些值,在修改过程中,α 值永不下降,β 值永不增加。

② Fail-soft Alpha-Beta 算法

在 Alpha-Beta 搜索中,Alpha,Beta 分别默认为 $-\infty$ 到 $+\infty$,在递归调用过程中,窗口不断缩小,剪枝的效率越来越高,因此 Fail-soft Alpha-Beta 算法就是刚开始限定窗口大小。在 Alpha-Beta 搜索中,若搜索失败,也没有什么信息,Fail-soft Alpha-Beta 算法作了改进,即当返回值≤Alpha,表明真实值≤返回值,若返回值≥Beta,真实值≥返回值,尽管有启发信息但速度仍不高。

③ 渴望搜索

渴望搜索是一种缩小范围来提高剪枝效率的算法,它渴望真实值在所定窗口范围内,建立窗口时,令 Alpha=X-window,Beta=X+window,猜测结果在 X 附近,此法当窗口很小时,能迅速提高剪枝速度,但失败率也很高。

④ 极小窗口搜索

极小窗口搜索,也称主变量导向搜索,它是另一种缩小范围来提高剪枝效率的算法。它假设搜索的第一个节点的 Value 值是最优的,然后以(Value,Value+1)为窗口搜索,效率明显高于上述几种算法。

⑤ MTD(f)算法

该算法是多次调用 Alpha-Beta 来完成搜索,每次均采用空窗口进行搜索,然后得出一个真实值上下边界,再以此返回值修改原来的上下边界,搜索范围不断缩小,从而向真实值不断逼近,当上下边界相等时,搜索就完成了,该搜索效率更高。

4) 其它优化方法

①置换表与哈希表

置换表就是用一张表把搜索的节点记录下来,然后利用记录下来的节点结果,在后续搜索中,查看记录在表中的这些结果,

当用置换表时有查询插入两个主要操作,因此要求查询速度插入速度要非常快,查询时最好类似于随机存储,而插入的过程要考虑数据移动和排序等操作。在这样一个有限空间里进行此操作,最好的数据结构就是哈希表。

②历史启发

所谓启发,就是被定义成的一系列规则,历史启发避免了对具体棋类知识的依赖,在搜索过程中,只要给出合适走法映射和增量因子,就能轻易的加入历史启发。

③迭代深化

这种技术既能满足深度优先搜索的线性存储要求,又能保证发现一个最小深度的目标节点。通过迭代深化,连续的深度优先搜索被引入,经实践还发现,迭代深化搜索扩展产生的节点数并不比广度优先搜索产生的多很多。

5 结论

六子棋中除了这些技术外,还包括开局库设计,系统测试等要素,本文只简单介绍了六子棋的基本技术问题,通过对本文的研究,我们可以进一步掌握六子棋研究的要点,树立正确的方向,从而提高计算机博弈的综合水平。

参考文献:

[1] 董红安.计算机五子棋博弈系统的研究与实现[D].山东师范大学,硕士学位论文,2005.
[2] 徐长明,马宇民,徐心和.一种新的连珠棋局面表示法及其在六子棋中的应用[J].东北大学学报(自然科学版),2009,30:514-517.
[3] 闵文杰.六子棋计算机博弈关键技术研究[D].重庆交通大学,硕士学位论文,2010.
[4] Pablo,s.s.,Ramon G.Fernando M.,et al.Eiffcient Search using bitboard models[C].International Conference on Tools with Artificial Intelligence.Washington DC:IEEE,2006
[5] 谢闽.中国象棋机器博弈数据结构设计与搜索算法研究[D].西安理工大学,硕士学位论文,2008.
[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
[7] 夏定纯,徐涛.人工智能技术与方法[M].华中科技大学出版社,2004,P128-130.
[8] Hyatt,R.M.and cozzie,A..The Effect of Hash Signature Collisions in a chess program[J].ICGA Jounal.2005.
[9] Jonathan Schaeffer.The history heuristic and alpha-beta Search enhancements in pracice.IEEE Transactions on Pattern Analysis and Machine Intelligence,November 1989.

本栏目责任编辑:谢媛媛