文章编号：1003-1251(2017)02-0039-05

多种搜索算法的五子棋博弈算法研究

董慧颖，王 杨

（沈阳理工大学 自动化与电气工程学院，沈阳 110159）

摘 要：主要选择五子棋为研究对象，应用Alpha-Beta剪枝算法、置换表技术搜索算法，研究人工智能模拟人类思考的推算过程，实现博弈效果。在Alpha-Beta剪枝算法中引入迭代加深以及局部搜索方法，提高程序棋技。在此基础上使用Monte Carlo方法和深度学习方法结合的方式来提高下棋技巧。实验结果表明，该算法相比于上述几种方法有明显的改进。

关 键 词：人工智能；机器博弈；五子棋；置换表；Alpha-Beta剪枝算法
中图分类号：TP391 **文献标志码**：A

Study on the Variety of Search Algorithms for Gobang Game

DONG Huiying, WANG Yang

(Shenyang Ligong University, Shenyang 110159, China)

Abstract: Gobang is selected as the target for studying, which combines with Alpha-Beta pruning algorithm and replacement Table to achieve game results. Iterative deepening and local search methods are adopted to improve chess level. Monte Carlo method and depth of learning on the basis of a combination of ways are applied to improve chess skills. Experiments show that the proposed algorithm has been significantly improved in comparison with the above-mentioned methods.

Key words: artificial intelligence; computer game; gobang; replacement table; Alpha-Beta

计算机的快速发展，催生了一门新兴的学科—人工智能[1]。它是用来研究、模拟和扩展人类智慧的一门理论方法和技术应用系统的新兴学科。机器博弈[2]是人工智能方向的一个分支，很多实际问题可以在博弈的研究中得到解决，并且使计算机智力更加靠近人类智慧。70多年前，电子计算机研制成功后，科学家们开始通过电子计算机模拟人类思维并逐渐向人类智慧发起挑战，香浓与图灵提出了对棋类机器博弈程序的描述。随着计算机硬件和软件的更新换代，从1980年起，机器博弈便开始逐步大规模地向人类智慧发起冲击。在1997年时，IBM的超级电脑Deeper Blue战胜了当年国际象棋世界冠军卡斯帕罗夫，成为人工智能挑战人类智慧发展的里程碑。今年三月份谷歌研发的AlphaGo大败李世石的事情又将人工智能推向了新的高度。由此，棋类玩法不再限制于实物，电脑游戏层出不穷，而五子棋的电脑游戏也越来越多，也就是现在新出的名词“计算

收稿日期：2016-06-20

作者简介：董慧颖(1962—)，教授，博士，研究方向：模式识别与智能系统。

机博弈[3]”。人工智能中多数是以下棋为研究对象来探索博弈效果。比如,谷歌的 AlphaGo 博弈技巧关键在于深度卷积神经网络的设计与训练。本文以五子棋做为研究对象,利用计算机博弈技术,采用蒙特卡罗算法,来设计一个智能五子棋[4]操作系统,制作人机界面实现人机对弈、机器对弈,并对传统的搜索算法进行改进。

1 搜索算法

1.1 Alpha-Beta 剪枝算法搜索

Alpha-Beta 剪枝[5]搜索算法是由极大极小值搜索算法[6-8]演变而来。极大极小搜索算法,即模拟人类思考的过程。程序执行开始,机器方走棋时,先按照一定的搜索深度生成出给定深度 a 以内的所有节点,并计算已生成的叶节点评价函数的值。然后从 $a-1$ 层节点开始逆向计算:对于机器要走的根节点(选取 MAX 记为极大值点)选取其叶节点中的最大值为该根节点的值(下棋者总会选取对自己最有利的落子点);而敌手走棋的根节点(选取 MIN 记为极小值点)取其叶节点中的最小值为该根节点的值。一直到计算出根节点的值为止。当得到根节点取值的那个叶节点,即为所选的最优走棋位置。极大极小搜索算法是计算双方博弈若干步之后,从可能的走法中选取相对最优的好棋的着法来走,即在给定深度的搜索范围内进行选取。综上,评估叶节点的值很重要。为此,设定一个静态评估函数[9] f,用来判断棋局的叶节点的值,这个函数可根据叶节点的价值进行估计来定义。一般定义有利于 MAX 的评估值 $f(p)$ 取正,有利于 MIN 的评估值 $f(p)$ 取负,评估值等值 $f(p)$ 取 0。若 $f(p)=+\infty$,则表示 MAX 赢,若 $f(p)=-\infty$,则表示 MIN 赢。设计规定:顶部叶节点深度 $a=0$,MAX 表示机器方,MIN 表示敌手方,MAX 方先走。

图 1 为博弈树搜索示例。通过上述过程得出 101-101-101-101 为最佳走棋路径。而剪枝算法则是在图示博弈树内搜索过程中判断出哪一个叶节点可以胜出棋局时,其他搜索就不需要继续了。即,对选取极大极小值进行判断,剪掉获胜机率小的叶节点,减小搜索量,提高效率。

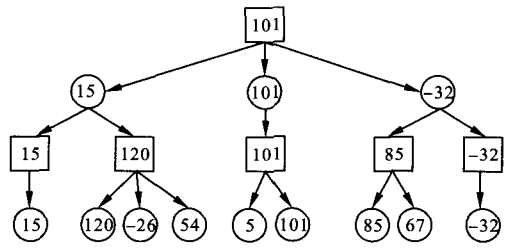

图 1 博弈树搜索过程示例

程序代码如下:

```
int Alpha-Beta(int depth,int alpha,int beta)
{if(depth = = 0){
return Evaluate();
}
GenerateLegalMoves();
while(MovesLeft()){
MakeNextMove();
val = -Alpha-Beta(depth-1, -beta, -alpha);
UnmakeMove();
if(val > = beta){
return beta;
}
if(val > alpha){
alpha = val;
}
}
return alpha;
}
```

1.2 置换表技术

在双方下棋对决的过程中,总会出现一些相似的棋局和走法,这时,如果可以把之前搜索博弈树产生的信息保存下来进行调用,可以减少大量的搜索时间。置换表中任意的一个数据项应有详细的信息说明对应博弈树的哪个节点,并且包括该节点的评估值,以及该值对应的搜索深度等。当得到了某一个叶节点的确切评估值,即可将上限值和下限值保存成相同表示方式。置换表在程序中要实时更新。机器博弈中的博弈树一般都很庞大,而 Alpha-Beta 剪枝搜索在一般情况下是边生成叶节点边搜索,且无需保存整个博弈树,因此内存占用量并不是很大。若置换表用来存储博弈树已经遍历过的全部叶节点的信息,内存占用量将十分庞大。从剪枝效率方面考虑,因为博弈树

顶层的剪枝对整体剪枝效率有着决定性的意义。所以,置换表只保存较顶层的博弈树节点信息,但还是可以显著地提高剪枝效率。

置换表一般采用 Hash 方法[10],主置换表是一个散列数组,主要实现过程如下:

```
#define hashfEXACT 0
#define hashfALPHA 1
#define hashfBETA 2
typedef struct tagHASHE {
int depth;
int flags;
int value;
MOVE best;
} HASHE;
```

2　设计实现

2.1　博弈算法结合实现

传统的计算机博弈搜索[11-12]算法,基本上都是应用上面叙述的搜索算法来实现博弈效果。时代在进步,对于算法的改进研究也在创新,都在寻找用以超越前人的方法,从而提出创新的算法。首先,通过流程图了解一下博弈树搜索位置的流程,搜索最佳位置的流程图如图 2 所示。

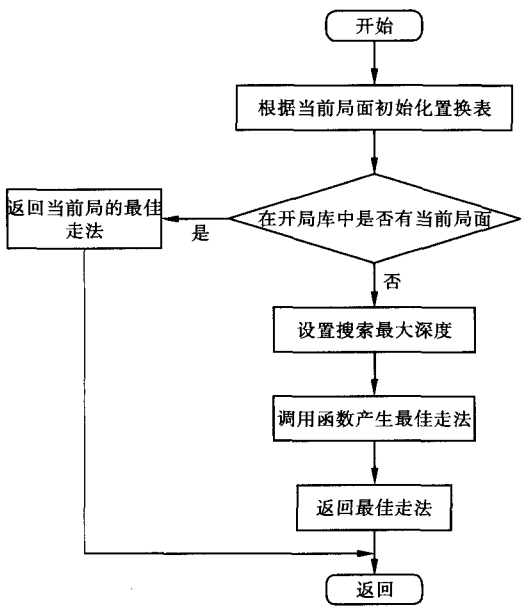

图 2　搜索最佳位置的流程图

其次,完成上面流程图所示的整个搜索过程一般要采用一种到几种搜索算法。设计先采用传统的搜索算法,实现基本博弈后,进行算法优化。在初始版本程序中,采用基本博弈树搜索,应用 Alpha-Beta 剪枝[13]搜索算法,对整个博弈树[14]进行剪枝遍历,尽可能选择最好的线路走棋。但这样的效率非常低,而且只能用递归来实现博弈。每一次遍历,博弈树的叶节点数呈指数递增。基于此,在修改与优化程序设计时,采用迭代加深的方法。它的思想是,在有限时间内,尽可能快速地搜索到一个深度。当时间充裕,可以向更深处搜索,甚至完成搜索;当时间不足时,也可以找到一步当前下棋最优点。表 1 为采用迭代加深前后搜索深度的变化。

表 1　采用迭代加深后最优搜索时间内变化次数

算法	迭代搜索消耗时间/s				
	1	2	3	4	5
迭代前	1	35	520	4892	53679
迭代后	1	42	709	7531	143562

从表 1 可以看出,这是一个很快速的搜索方法,在一定程度上提高了搜索效率。搜索过程中要用到置换表,把搜索得出的数据与信息以数字的形式存储于置换表中,在下棋时刚好遇到搜索过的情况就可以从置换表中直接调用,省去了很多重复的工作,最主要的是节省了时间,进而在最短的时间内搜索到更深的地方。置换表在每次搜索过程中所要记录的数据量是庞大的,一般采用 Hash 方法实现。设计中对此进行了改进,加入了深度优先覆盖策略,实验结果表明,这样提高了搜索效率也改善了优化性能。

深度优先覆盖策略的部分代码如下:

```
void RecordHash(int depth,intval,inthashf){
HASHE * phashe =
&hash_table[ZobristKey()&
(TableSize()-1)];
if(phashe->hashf != hashfEMPTY&&
phashe->depth > depth){
return;
}
phashe->key = ZobristKey();
phashe->best = BestMove();
```

```
phashe -> val = val;
phashe -> hashf = hashf;
phashe -> depth = depth;
}
```

2.2 Monte Carlo 方法

Monte Carlo 方法[15]是通过随机数(或者伪随机数)来处理许多计算问题。与它对应的是确定性算法。经过数千盘的模拟随机对战结果,并统计所有走法的双方输赢局数之差,差值最高并且满足一定概率的走法即为当前最优落子点。此方法实质上是对以当前局面为根节点的博弈树中的所有叶节点进行随机抽样,令抽样结果中的胜负概率与所有叶节点中的胜负概率大致相同,并以此作为决策的依据。

2.3 深度学习方法

深度学习[16-17]是最近研究比较热门的话题。它来源于神经网络的研究,设计是通过给定输入和输出,从而训练调节每一节的权重,这样可以大大提高博弈树剪枝的精准度和搜索效率。充分模拟人脑思考下棋思维,结构图以两输入为例,如图3所示。

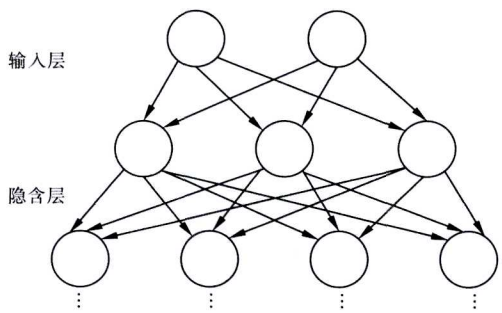

图3 深度学习结构示图

棋盘设计为14×14,在棋盘上以当前落子点为中心任意方向上9个点,即构成本文所述的棋型,其可能的情况为3^9种。

训练集生成过程:在网上查找到一个具有5581局的五子棋棋局库。对胜负局的棋谱进行统计,选择在某一状态下的一种棋型,记录该棋型在不同局面中胜负局内出现的次数。把胜负局的次数进行比值,求得该点的评估值。以此类推,得出相对全面的每种棋型的评估值,对这一评估值进行归一化到[-1,1],棋型归一化示例表如表2所示。

表2 棋型归一化列举

棋型	胜局中出现次数/负局中出现次数	归一化
棋型Ⅰ	11356/891	1
棋型Ⅱ	5478/2434	0.45
棋型Ⅲ	2456/2435	0
棋型Ⅳ	7439/9342	-0.57
棋型Ⅴ	1678/13491	-1

最后,把棋型作为输入,将归一化后的的值作为标准输出。得到包含6573个输入输出对的训练集,对神经网络进行训练。在训练中,计算机不断积累经验,得出最优的评估函数,对单个点进行评估。

2.4 禁手设计

设计使用VS2010环境搭建人机交互界面[18]。由于五子棋先手者获胜的几率很大,所以设计评估函数时分为先手和后手两类。先手的方法是注重自己的非关键棋型分值,相当于加重进攻走法的分值;后手则加重对方下棋的分值,相当于加重计算防守走法的分值。原理是利用局势的发展作为导引,在对博弈树叶节点进行评估时,文中使用静态评估函数,通过神经网络设计出的评估函数$h(i)$来衡量每一个叶节点的值。选取评估值最大值为接近或等于1的节点作为MAX(MAX设为先手方)方的最佳走棋点。最小的评估值为接近或等于-1作为MIN(MIN设为后手方)方的最佳走棋点。简单用一个例子概括,如图4所示。

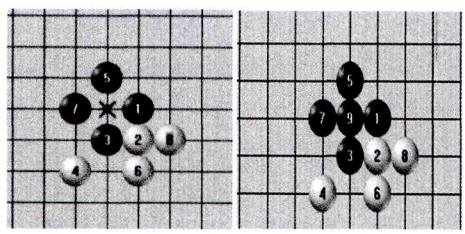

图4 禁手下棋

从图4可以看出,黑棋子9是黑色棋子的禁手,若黑棋下此处则判为黑棋方输。程序实现上处理方法为:在累加分值的时候添加一句判断语句,比如正好形成两个三连子,那么这个语句无需添加到分指标。同时,判断此点落棋为负,其他禁手使用相同的方法。

3　测试与优化过程

通过程序设计完成实验虚拟平台的搭建与设计。搭建完成的人机交互界面 - 平台界面图如图 5 所示。

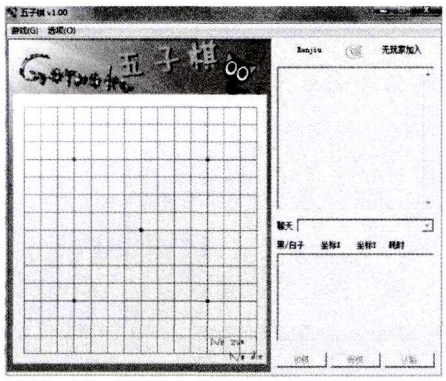

图 5　五子棋实验操作平台

界面设计了时间记录这一项，方便实验结果分析。为了在整体上提高搜索速度，考虑到使用局部搜索策略，减少搜索时间与无用的工作占用的时间。如表 3 所示，局部搜索至少提高了一半的时间，进而提高了搜索的速度。

表 3　全局搜索与局部搜索时间比较　　s

算法	搜索次数		
	10	50	100
全局搜索	15.892	39.055	81.394
局部搜索	5.106	14.993	24.069

局部搜索缩小了搜索范围，减少了搜索时间和一些不必要的节点搜索。设计中为了提高整体下棋水平，由于剪枝算法也有局限性和缺点，经过反复实验考量，决定使用 Monte Carlo 方法替代剪枝搜索算法。通过遍历博弈树的评估函数值，选取大于决策概率估值的点作为最终得到的最优走法，以提高下棋水平，来达到预期的效果。

表 4 为最终程序 SLG-Gobang-V3.0 与之前版本对弈的胜负结果对比。

表 4　程序博弈结果比较

程序名称	比赛次数	获胜次数	获胜率
SLG-Gobang-V1.0	100	95	95%
SLG-Gobang-V2.2	60	56	93%

由表 4 可以看出，获胜概率在 93% 以上，证明了算法的可行性。

4　结束语

研究了传统的博弈搜索算法的不足与改进之处，并且提出了优化程序的方法与措施。采用置换表、深度学习、Monte Carlo 方法和局部搜索等方法，丰富了程序，提高了搜索性能。在与 Alpha-Beta 剪枝搜索算法为主要博弈算法的程序进行博弈时，有较好的优化策略、博弈效果。与传统的设计相比有了很大程度的提升。

参考文献：

[1] 郑南宁. 人工智能面临的挑战[J]. 自动化学报，2016(5)：641 - 642.

[2] 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，2008，3(4)：288 - 293.

[3] 徐心和，王骄. 中国象棋计算机博弈关键技术分析[J]. 小型微型计算机系统，2006，27(6)：961 - 969.

[4] 杨云强，吴姣. 一种改进的基于博弈树模型的五子棋系统[J]. 科学技术与工程，2012，20(5)：1052 - 1055.

[5] 程宇，雷小锋. 五子棋 Alpha-Beta 搜索算法的研究与改进[J]. 计算机工程，2012，38(17)：186 - 188.

[6] 张明亮，吴俊，李凡长. 极小树叶节点数定理的补充证明及有关分析[J]. 模式识别与人工智能，2011，24(4)：521 - 526.

[7] Wang D，Tan A H. Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game[J]. Computational Intelligence & Ai in Games IEEE Transactions on，2015，7(2)：123 - 138.

[8] Korf R E，Chickering D M. Best-first minimax search[J]. Artificial intelligence，1996，84(1)：299 - 337.

[9] 张玉琪. 基于静态评估的计算机围棋 UCT 算法改进研究[D]. 南昌：南昌航空大学，2015.

[10] Zobrist A L. A new hashing method with application for game playing[J]. ICCA journal，1970，13(2)：69 - 73.

[11] Gherrity M. A game-learning machine[D]. University of California，San Diego，1993.

（下转第 83 页）

硼酸的浓度和 pH 值可促进 Cu 在三维石墨烯表面的沉积速率。

(2)镀液中络合剂柠檬酸钠浓度的提高会降低 Cu 沉积速率,但可以提高 Cu 沉积层表面的质量。

(3)当硫酸铜 10g/L、硫酸镍 1g/L、柠檬酸钠 15g/L、次磷酸钠 30g/L、硼酸 30g/L 、pH 值为 12 时,Cu 沉积层可将三维石墨烯表面均匀包覆。

参考文献:

[1] Singh V, Joung D, Zhai L, et al. Graphene based materials: past, present and future[J]. Progress in Materials Science, 2011, 56(8): 1178 - 1271.

[2] 任芳,朱光明,任鹏刚. 纳米石墨烯复合材料的制备及应用研究进展[J]. 复合材料学报,2014,31(2):263 - 272.

[3] Rao CNR, Sood AK, Subrahmanyam KS. Graphene: The new two - dimensional nanomaterial[J]. Angewandte Chemie International Edition, 2009, 48(42): 7752 - 7777.

[4] 高志强,沈晓冬,崔升. 化学镀铜的研究进展[J]. 材料导报,2007,21(1):217 - 219.

[5] 周永璋,张果金,杨朗. 影响化学镀铜液稳定性和镀速的因素[J]. 电镀与环保,1999,19(5):13 - 15.

[6] 李宁,屠振密. 化学镀实用技术[M]. 北京:化学工业出版社,2004. 245 - 287.

[7] 龚文照,陈成猛,高建国,等. 热膨胀石墨烯表面化学镀纳米镍[J]. 新型炭材料 2014,29(6):432 - 437.

[8] Caturla F, Molima F, Molima-Sabio M, et al. Electroless plating of graphite with copper and nickel[J]. J Electrochem Soc, 1995, 142(12): 4084 - 4086.

[9] FANG Jian-Jun, LI Su-Fang, ZHA Wen-Ke. Microwave absorbing properties of nickel-coated graphene [J]. Journal of Inorganic Materials, 2011, 26(5): 467 - 470.

[10] 何为,吴婧. 优化试验法在化学镀铜工艺研究中的应用[J]. 实验科学与技术,2010,8(2):35 - 38.

(责任编辑:王子君)

(上接第 32 页)

[3] Swain P, Austin R, Bally IC, et al. Development and testing of a tethered, independent camera for NOTES and single-site laparoscopic procedures[J]. Surgical Endoscopy, 2010, 24(8): 2013 - 2021.

[4] 全薇,张雷,张正正,等. 医用电子宫腔镜光学成像系统设计[J]. 沈阳理工大学学报,2014,33(5):6 - 9.

[5] 陆小建,杨琰,濮悦. 内窥镜照明方式的选择[J]. 无损检测,2013,35(3):60 - 61.

(责任编辑:赵丽琴)

(上接第 43 页)

[12] Xian M, Shijin D, Hong W, et al. Research on computer game of incomeplete information[C]//The 27th Chinese Control and Decision Conference (2015 CCDC), Qing Dao. IEEE, 2015: 5807 - 5810.

[13] Stockman G C. A minimax algorithm better than alpha-beta[J]. Artificial Inte-lligence, 1979, 12(2): 179 - 196.

[14] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep ne-ural networks and tree search[J]. Na-ture, 2016, 529(7587): 484 - 489.

[15] 曹一鸣. 基于蒙特卡罗树搜索的计算机扑克程序[D]. 北京:北京邮电大学,2014.

[16] 严文蕾,李娜. 互联网时代的教学创新与深度学习 - 美国的经验与启示[J]. 远程教育杂志,2016,35(2):26 - 31.

[17] 刘建伟,刘媛,罗雄麟. 深度学习研究进展[J]. 计算机应用研究,2014,31(7):1921 - 1930.

[18] 张佳佳. 五子棋对战平台的设计与实现[J]. 电脑知识与技术:学术交流,2012,8(8):5409 - 5411.

(责任编辑:马金发)