

五子棋机器博弈系统评估函数的设计

张明亮^{1,2}, 吴俊¹, 李凡长^{2*}

(1. 苏州科技学院 电子与信息工程学院, 江苏 苏州 215011; 2. 苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

(* 通信作者电子邮箱 lfz@suda.edu.cn)

摘 要: 针对五子棋机器博弈需借助棋型评估棋局产生的速度瓶颈, 提出了棋型识别粒度与搜索深度相关联的多层架构评估函数设计方法。实验证实粗颗粒度估值在深层搜索时整体效率更高; 多层次判读棋型能明显加快评估速度; 走法生成函数引入快速评估, 提前进行博弈树自然剪枝收效显著; 非关键棋型依据棋局浮动分值平衡搜索深度, 对五子棋亦见成效; 以及估值函数针对性地延伸评估来减少地平线效应的开销更少; 从而表明估值函数动态评估更加合理。

关键词: 博弈树; PVS; MTD(f); 评估函数; 五子棋

中图分类号: TP181 **文献标志码:** A

Design of evaluation-function for computer gobang game system

ZHANG Ming-liang^{1,2}, WU Jun¹, LI Fan-zhang^{2*}

(1. College of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou Jiangsu 215011, China;

2. School of Computer Science and Technology, Soochow University, Suzhou Jiangsu 215006, China)

Abstract: Concerning the speed bottleneck of the gobang machine game when relying on the configuration of the stones to evaluate the game states, this paper proposed a multi-layer evaluation-function method combining identification granularity of the stones' configuration with the search depth. The experimental results have proved: the overall efficiency of coarse-grained evaluation was higher when the deep-layer searches were processing; multi-layer judging for the configuration of the stones could obviously accelerate the evaluation; if the move-generating functions were introduced into the rapid evaluation, it was more efficient to prune certain branches from game-tree in advance; gobang game tree searches also benefited if floating the values of non-critical stone's configuration to balance the search depth. Therefore, dynamic evaluation of evaluation-function is more reasonable.

Key words: game-tree; Principal Variation Search (PVS); MTD(f); evaluation-function; gobang

0 引言

机器博弈被喻为人工智能领域的果蝇, 多以下棋为例研究其规律。计算机弈棋系统可分为四部分: 知识表示、走法生成、评估函数和博弈树搜索。von Neumann 和 Borel 在 20 世纪 20 年代提出的极小极大值定理^[1-3]是搜索算法的数学基础; 始于 20 世纪 50 年代的 α - β 剪枝算法^[1]在搜索效率上前进了一大步。1980 年的 PVS(Principal Variation Search, 亦称 NegaScout) 搜索算法^[4-5], 内部节点强有序时效率优于 α - β 搜索; 1994 年出现的 MTD(f) 算法^[6-7], 始终用空窗探测逼近真值, 借助“置换表”(Transposition Table) 增速完成搜索, 略优于 PVS 搜索, 两者为目前主流方法。均衡博弈树(Uniform Game Tree)^[1]时, 以上 α - β 类方法的最小搜索极限为极小博弈树(minimal game tree) 叶节点数定理(Number of Bottom Positions, NBP)^[1, 8-11]。多数游戏的博弈树异常庞大, 无法完备搜索, 常规做法是搜索至有限深度, 此时的叶节点由评估函数近似估值。评估的速度和精确性都至关重要, 两者又相互制约。评估不准现象称为地平线效应, 狭义的地平线效应专指估值严重偏离实际, 浅层搜索更易出现, 常用解决方法是静止期搜索。评估可以是全局扫描, 也可以相对某初始局面

快速地局部增量评估^[12-14]。全局估值的精确度高, 也易于动态加深评估等, 使用更广泛。时至今日, 国际象棋和中国象棋的机器博弈水平已超越最好的人类选手^[15-17]。五子棋是普及程度仅次于国际象棋的棋类游戏, 职业棋手人数同样居世界第二, 棋盘 15×15 大小, 分职业(“禁手”) 规则和业余(无“禁手”) 规则两种。无“禁手”五子棋有专业棋手普遍认可的“先行必胜”之说, 因而职业规则对先行方“禁手”限制, 规则比较繁复且仍在演化; 非职业棋手则一致偏爱业余规则。五子棋机器博弈开展得很早, 目前公认最好的系统有俄罗斯人 Victor Barykin 开发的 BlankStone(“黑石”) 及旅美华人温向东开发的 Renju Solver 等, 均使用职业规则, 属商业软件; 国内当数“连珠妙手 Fiver6”(作者署名“LWS 于合肥”) 共享软件, 2000 年完成, 主要使用无禁手规则, 对等时限的进攻棋力丝毫不逊 BlankStone, 但防守稍弱。五子棋博弈系统的水平还不及人类, BlankStone 的计算力专业棋手评价约及职业 3 段(最高 9 段), 但中局以后水平明显下滑, 并且攻强守弱。原因是五子棋博弈树比象棋类的稠密, 棋局评估速度也更慢, 除连续攻击的搜索易于完成外, 时间限制计算机无法完成足够深度的搜索。五子棋开局就攻防激烈, 双方合理招法相对有限, 无禁手时胜负的关键多取决于前 30 手棋; 借助棋类知识

收稿日期: 2012-01-16; 修回日期: 2012-03-15。

基金项目: 国家自然科学基金资助项目(60775045); 苏州科技学院科研基金资助项目(xky201010)。

作者简介: 张明亮(1965-), 男, 江苏赣榆人, 副教授, 硕士, 主要研究方向: 博弈论、计算机应用系统; 吴俊(1978-), 男, 江苏苏州人, 工程师, 硕士, 主要研究方向: 人工智能、计算机图形学; 李凡长(1964-), 男, 云南宣威人, 教授, 博士生导师, 主要研究方向: 人工智能、李群机器学习、动态模糊逻辑。

大力强化走法生成函数,变相地提前剪枝,开局阶段博弈树能比象棋的稀疏。象棋用棋子棋力值之和能大致评估棋局,结合棋子位置(灵活性)及组合等因素浮动^[14]就很精确;五子棋则需通过“棋型”评估,棋型的搭配对准确性影响也更大。不同于人,计算机棋型判读速度落后于象棋的棋子统计,棋型搭配的耗时更远多于识别棋子位置等,也难以采用象棋“比特棋盘”(bit boards)^[18-19]技术加速。因而评估速度成为五子棋博弈系统的一大瓶颈,也凸现了估值函数设计的重要性。

1 棋型分类

五子棋局面估值有两种计分方式:一种是棋型统计,耗时费力,精确性较好;另一种判读棋盘上有价值的空点,快速但不易准确。本文系统对非关键棋型用统计方式,关键棋型结合先后手情况用空点方式判读。

图1所示为五子棋的必胜棋型,分为一条直线(横、竖、正斜、反斜)上存在两个将成“五子”的空点,或在至少两条直线方向上的各有一个成五子的空点,此时一方必然可胜。图中的a模型为普通“活四”型,左右两边各有一个成五的空点;b模型为普通的双“冲四”模型,在两个方向上各有一个成五点,简称“四四”;此类模型还有多种表现形式。有趣的是,归纳时发现“禁手”规则未曾明确提及的一些必胜棋型,见图1的c、d、e模型,c模型最后招法为带“c”标识的三个落子点之一,造成一条直线上的两个成五点;d模型最后招法为“d”标识之一,同样两个成五点;e模型标识“e”的落子亦同理。三种棋型的概率依次降低,c型较易出现,计算机穷尽识别这些模型的时间开销较大。

图2所示为“准”必胜棋型,一方走出“冲四”带“活三”模型,或双“活三”模型,往往获胜,但不意味着必定获胜,因为对手的阻挡招法可能是反“冲四”。因此本文称“准”必胜型。图中f模型是“冲四”带“活三”,简称“四三”;g模型为双“活三”模型,简称“三三”,这两种模型的具体形式还有很多。h模型白方下一手能得到c必胜型,i模型黑方下一手将得到a或d必胜型。这是c、d、e模型的外推结果。

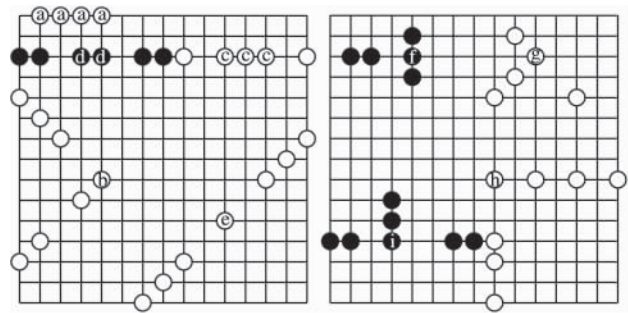


图1 必胜棋型

图2 “准”必胜棋型

图3涉及“必应”棋型。一方招法不“应对”将出现必胜型,称为“必应”,加引号是该类棋型一定的前提条件才是必应。“活三”是一种典型的必应棋型,这里继续延伸一下,能导致c、d、e必胜型的先导模型也归为“必应”型。能出现b模型的先导“必应”模型未包括,用走法生成函数去处理更好。图3中的j、k、l、m、n模型为通常的“活三”类棋型,专业术语“活三”分“连三”和“跳三”,前者如j和m模型,后者如k、l、n类模型。本文未这样分类,因其太过粗略,无法准确反映模型的具体价值。比如j类“活三”模型的必应选点只有两个,m类模型的必应选点有3个,尽管两者都是“连三”。k、l、n类的必应选点都有3个,但对方应对后的潜力依次降低。总体

而言,j类棋型攻击性更强,因为留给对手的选择余地更小,两端都有多个空点的“跳三”潜力也较大,精确评估需要按此原理细分。图3中的o类棋型则不是“必应”型,是假“活三”,这表明“必应”模型至少涉及6个点位的连续判断,细分至少需要连续7个点位。p、q、r、s模型是本文对“必应”模型的扩充,p和q型对应图1的c必胜型;r型对应d型,至少涉及连续8个点位识别;s型对应e型,至少涉及9个连续点位。通过连续的“活三”攻击获胜称为VCT(Victory of Continuous Three)。

单“冲四”模型是只在一个方向上欲连成五子,也称绝对先手,其显著特点是冲四点必成对,专业术语将两个“冲四”点分别称“冲四”和“嵌五”。“冲四”能创造新棋型的,分值可大幅增加,但其潜力不易准确评判。连续的“冲四”而获胜称为VCF(Victory of Continuous Four)。图4示意了一部分有价值的棋型,该类棋型的形式还很多,其中t“冲四”型若封闭存在潜力很小;u、v、w“冲四”另有潜力,分值应稍高。该类棋型的识别至少涉及5个落子点,一般用7个点位细分。x、y、z模型是“活二”类棋型,也应计入分值,其判断需至少6个落子点。

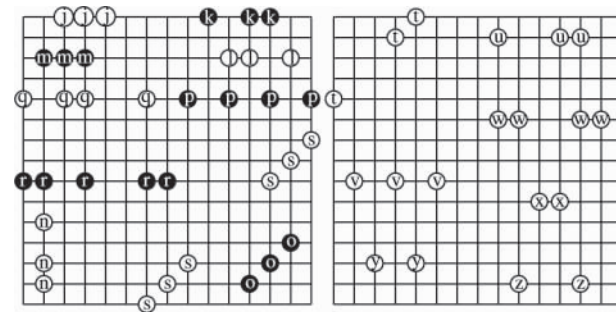


图3 “必应”棋型

图4 其他潜力棋型

图1~4将棋型归为4大类,其中,必胜型和“准”必胜型本文称为关键棋型。各种复合棋型只是举例,未全部列出。必胜型“空点”快速识别要领是:某“冲四”带来至少两个“成五”点。评估需对以上棋型加以甄别,切忌误判和重复记数。“必应”以下棋型,分值要结合棋局浮动。不同棋型存在制约关系,先要理清逻辑关系,细致缜密地处理,分值也需要很好地平衡。正确完整地判读关键棋型是基础,尤其不能将“准”必胜型简单等同于必胜型(必须进一步判断),许多对弈系统犯此通病,实验曾多次观察到,有的系统宣称几乎内必胜,是搜索到了一个“准”必胜型,实际对弈因对手反“冲四”而失算。五子棋一次缓手,棋局就基本逆转。本文对弈系统与“连珠妙手”的实验,多次出现c模型获胜,一次d模型获胜,对方招法有些后知后觉,与搜索深度不符,侧面印证其棋型识别不完备。这些都说明主要棋型完备识别的重要性。

2 估值函数的设计

评估函数判读非关键棋型的精细程度本文用“颗粒度”形容,速度和颗粒度的矛盾如何平衡,本文系统进行了很多测试,这里做一下总结。细颗粒度的估值无疑更精确,但会大幅增加时间,有时不可取;用加深搜索来解决判断的准确性,从应用角度看,是不同方式解决同一问题。浅层搜索时,不精确的评估对结果影响很大;而十几层深度的搜索仍然成立的招法,多数情况下已较易区别优劣,细致评估就往往得不偿失。12层以上的五子棋对弈实验,不同颗粒度的评估,走法惊人地一致,几乎无差别,但速度差距很大,不同搜索算法虽有差

异,也都以较粗粒度的为佳。浅层搜索细粒度的综合效果更好,增加有限的时间换来较高的精确度。因此本系统深层搜索用中等颗粒度的评估函数,若干非关键棋型不予细分地统计,省略部分潜力棋型统计。较粗粒度估值还使完全由空窗探测实现的搜索额外受益,因取值空间相对压缩,可有效减少接力式空窗探测的次数,这也提示非关键棋型取值应该尽量连续,去除不必要的断档空间。细颗粒度估值的棋型则加以细化,一般多于上述棋型举例。

五子棋的走法生成函数,原本可以非常简单,因为棋盘上所有空点都是合法招法(象棋的走法生成函数要复杂许多)。然而,五子棋开局招法就环环相扣,几乎不存在“你攻我的,我打我的”现象,通过强力过滤选择合理招法的余地大,本系统因此采用强化走法生成的策略。借助棋类知识生成可能的走法,3步内导致必胜结局的直接返回评估值(相当于提前生成自然终结节点),其他情况下使用空点(走法)评价函数,较细致地排序和过滤,一些重要棋型适当延伸判读。这样做本身开销巨大,实测时间开销占到总搜索的2/3,但能极好地裁剪博弈树。此时的博弈树已远不均衡,大大“瘦身”:一是子节点数目有效减少;二是提前两到三层完成了自然终结节点的评估,省略其展开过程;三是博弈树同步优化。实验很成功,总时间开销反而减少了50%以上。这相当于将大部分的叶节点估值开销,转化成走法生成开销。

因五子棋先手优势大,评估函数分为先手和后手两类:先手时加重己方非关键棋型分值,相当于加重进攻招法的分值;后手则加重对手棋型分值,相当于加重计算机防守招法的分值,道理是利用局势的发展趋势稍作引导,实验效果很好,明显加快后手棋的搜索,表明起到了优化博弈树的作用。这也揭示五子棋执白后手时,开局应该防守为主,不要急于对攻或一味地反击,耐心防守是高水平对弈的最佳策略。优势棋局时另外设计了制胜评估函数,主要检查必胜型、准必胜型、VCF和VCT模型,后3种模型延伸判断,相当于局部增加了3到4层搜索;VCT需仔细结合对手“冲四”处理,难于判断时交由正常搜索处理。此外,本文系统获胜棋型的评估值依据所处深度进行调整,不使用单一值,便于计算机找到最短获胜路径。因而,本文的评估函数大致分五类:浅层搜索和部分迭代深化搜索用细颗粒度评估函数,后手扳平局势时启用均势评估函数,优势时转用先手评估函数。优势较大尝试制胜走法生成函数(在博弈树最末两层,尝试逆向过滤非必胜走法),配合以制胜评估函数。

本文系统在对手走法后的偶数层叶节点进行评估,估值函数采用层次结构,由多个子函数配合完成。最后招法向估值函数传递棋型信息,对方单冲四(双方直接的必胜棋型已由走法生成函数评估返回)延伸相应走法最多3层还无法到静止期的,调用某方单冲四的估值子函数粗略评估,其中延伸1或3层的使用等待对手行棋的评估函数,估值口径不同。其他情况对应静止期评估,依次进行以下扫描:搜索计算机必胜型空点;利用已经得到的单冲四信息搜索计算机有否“四三”型必胜(要延伸判断)的空点;单向扫描棋盘统计其他棋型(包括对手的“活三”型)。延伸评估值依棋局优劣微调,来大致平衡不同深度的估值。

追求速度的程序设计对博弈系统至关重要,提速空间往往很大,与棋局表示法也关系密切,细节繁复,这里仅用C语言简单举例。棋局若用 15×15 二维数组表示,方便时当作一维使用,如`chess[4][5]`用`chess[0][65]`形式访问速度稍快。

双方棋子的表示数值可以完全拉开,如黑棋用“1”表示,白棋用“-100”不要对称表示。计算机流水线技术会造成连续的加法比连续的逻辑表达式运算速度更快,用`chess[0][65] + chess[0][80] + chess[0][95] + chess[0][110] == 3`,即可结合当前选点判断出黑棋垂直方向是否冲四,速度大为加快。双冲四必胜型的检测依上所示先从一个方向逐步相加判断出是否冲四,用不同返回值代表相加的起始位置,再后续处理,因为非冲四为大概率,上述方式提速明显;其他棋型也依照处理。以上技巧合计能使评估速度提高至少一倍。此外,因本文系统划分的棋型较多,加之要利用加法的速度,未使用“位棋盘”技术,另设计了邻接表技术(在走法生成函数中作用最大),用来表示与某点位相邻接的4条线上各4个邻接点位棋子状况,随棋局动态更新,也显著加快棋型识别速度。

3 评估函数的使用

本文系统在2006年的普通微机上成功实现较完整的14层搜索,胜负判断达到17层,速度可保证实际对弈,基本达到了业余棋手的最高水平(普通国际象棋特级大师在中局时的计算力,与计算机较完备的14层博弈树搜索的棋力大致相当。围棋和五子棋优秀职业棋手的准确计算深度会高一些)。

图5是本文系统后手与“连珠妙手 Fiver6”“Domini”级的对局,由于最高级别是针对2000年普通微机设置的,“连珠妙手”速度虽很快,但搜索深度不足。一般认为每增加一层,博弈树搜索用时约增加4到7倍。在单核Pentium 4、内存1 GB微机的整个对局,本文系统14层搜索用时大致是Fiver6的4倍。白2称为“斜指”,至黑3后成为“蒲月”开局,是职业棋手普遍认同的典型“先手必胜”开局之一。白6的招法专业棋谱(多建议应在图中“8,J”位置)上未查到,是计算机搜索到的顽强招法,除白6外的前12手棋与专业棋谱契合。黑13手棋本文系统在18层搜索时指示图中“7,E”位置必胜,黑方错过后已不易获胜,至白28手棋局大致均势。黑31手败招,一方面是搜索深度不够,加之执黑先行多以获胜为目标,系统设置上偏重于进攻招法所致。

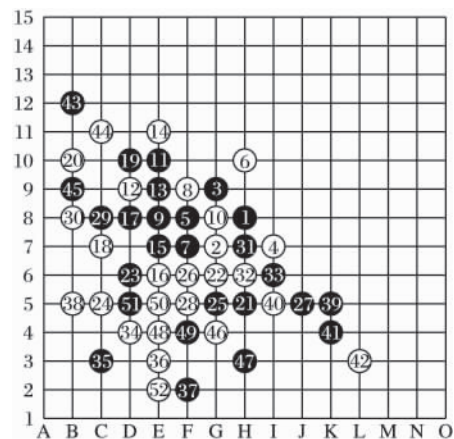


图5 计算机后手“蒲月”开局棋谱

表1是程序自动记录的搜索数据,使用BT搜索方法^[9-10],搜索深度为14层。其中“招法”包括手数和落子坐标,“用时”指整个一次搜索的全部时间,“估值”是相应叶节点棋局的评估值,介于-128至+127之间;走法优化后最多保留40个走法分支,“节点开销”指走法生成函数中的部分子函数调用开销。白棋至22手使用后手评估函数,26手由

浅层搜索出唯一应招直接完成,第 28、32、34 手改用均势评估函数,白 36 手后期的接力搜索自动转用优势评估函数,其最后估值 119 已判断出接下来的 17 手内必胜,后续招法改为用时极少的制胜搜索和制胜评估,表中略去。该棋谱反映出白棋的棋局评估比较准确有效。

表 1 五子棋后手“蒲月”开局的 14 层 BT 搜索实验

招法	用时/s	评估次数	节点开销	估值
2(7,G)	14.5	668 105	11 276 520	-5
4(7,I)	10.4	445 481	8 326 961	-6
6(10,H)	10.3	469 543	7 840 824	-6
8(9,F)	9.8	458 343	7 256 871	-7
10(8,G)	5.2	272 109	3 497 690	-9
12(9,D)	22.9	1 234 018	15 300 922	-11
14(11,E)	23.2	1 038 629	15 158 890	-13
18(7,C)	51.2	2 073 666	29 957 685	-9
20(10,B)	35.9	1 002 357	24 278 335	-5
22(6,G)	5.8	182 534	3 330 503	-4
26(6,F)	0.3	39 402	104 068	-4
28(5,F)	1.4	33 357	859 046	-1
32(6,H)	2.8	64 842	1 683 667	0
34(4,D)	15.7	409 156	8 385 067	4
36(4,G)	14.3	253 326	9 691 550	119
合计	223.4	8 644 868	146 948 599	

图 6 是指定白 2 为“直指”招法的对局,若让计算机搜索一般选择“斜指”,职业棋手普遍认为黑 3 是“直指”开局中黑方最好招法,白方将速败,本文系统若 14 层搜索对弈,白 4 将选在图中“8,G”位置形成“花月”开局,白方会在 30 手内告负。这里换用 16 层 MTD(f) 搜索对弈,整个对局用时大致是“连珠妙手”十余倍。16 层搜索的白 4 招法相对顽强,黑 5 不够准确,本文系统提示黑 5 应在图中“6,F”位置,棋局进行到黑 9 时也该选“8,F”位置。依据估值黑 11 疑问手,白 18 后棋局大致两分。黑 29 手继续侧重进攻致使局势逆转,这也反映出“连珠妙手”进入中局后确实攻强守弱,猜测是分支因子过小所致。至白 36 手系统已判断出在 19 手内必胜,后续招法同样用制胜搜索完成(注:棋局转优后,正常设计在白 32 手会尝试更深层的制胜搜索,后续用时会明显减少。另,验证显示“连珠妙手”在黑 7 和 19 手存在变招,这与黑 3 的简单随机不同)。

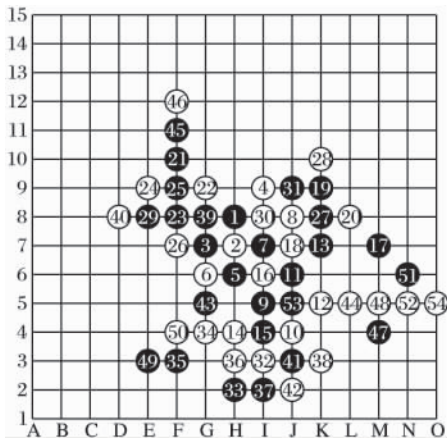


图 6 计算机后手“直指”开局棋谱

表 2 是相应的实验数据,为更多地反映评估函数设计的影响,除使用置换表和一次 4 层搜索优化外,两实验未再使用迭代加深等博弈树优化技术。白 34 手用时很多,而评估次数

和走法生成开销与之不对称,是反复空窗探测时置换表更多地发挥了作用所致。对比两个棋局可以看到本文系统 16 层搜索用时大约仅为 14 层搜索的 5 到 7 倍,与分支因子 40 很不相衬,表明前文所述大部分评估开销转移到走法生成函数的设计,确实使五子棋博弈树大为“瘦身”。这种设计在浅层搜索时体现不出优势,但越是深层搜索,倍增系数下降得越明显。这俩短对局分支因子 40 也相对富裕,减为 30 的实验证实,以上所有招法和估值都能完全再现,但用时仅减少 30% 左右,也侧面说明本文的评估函数设计相对合理。

表 2 五子棋后手“直指”开局的 16 层 MTD(f) 搜索实验

招法	用时/s	评估次数	节点开销	估值
4(9,I)	93.4	4 261 113	73 623 528	-6
6(6,G)	82.6	4 517 364	58 849 040	-8
8(8,J)	103.5	5 117 365	75 361 398	-6
10(4,J)	23.9	1 105 125	15 834 504	-7
12(5,K)	35.4	1 471 944	23 535 377	-6
14(4,H)	39.6	1 341 771	27 196 742	-6
16(6,I)	82.8	2 447 603	58 549 341	-6
18(7,J)	220.1	7 846 997	140 347 138	-2
20(8,L)	20.2	525 561	13 920 629	0
26(7,F)	32.4	1 039 773	17 760 472	5
28(10,K)	40.4	937 365	26 636 771	6
30(8,I)	9.9	283 152	5 444 682	6
32(3,I)	42.3	1 053 215	24 997 889	7
34(4,G)	228.7	5 356 388	128 711 024	19
36(3,H)	62.1	1 196 775	38 659 476	118
合计	1 117.3	38 501 511	729 428 011	

4 结语

五子棋博弈系统对评估函数的依赖程度相对更高,评估函数的设计与运用更富技巧。本文从棋型分类入手,给出棋型识别评估函数的成功实例;对设计与使用的有关问题给出了经过实战检验的有效解决方案;证实粗颗粒度的评估函数在深层搜索中综合效果更佳;估值函数延伸评估技巧同样可以减少地平线效应;根据棋局状况调整估值以及走法生成函数协助评估对五子棋亦见成效。这表明将搜索算法、搜索深度以及当前局面等因素与估值函数相结合,采用相对动态的评估方式更为有效。

参考文献:

[1] KNUTH D E, MOORE R W. An analysis of alpha-beta pruning [J]. Artificial Intelligence, 1975, 6(4): 293 - 326.

[2] KJELDSEN T H. John von Neumann's conception of the minimax theorem: a journey through different mathematical contexts [J]. Archive History Exact Sciences, 2001, 56(1): 39 - 68.

[3] 尚宇红. 极小极大值理论的历史发展[J]. 西北大学学报: 自然科学版, 2003, 33(2): 245 - 248.

[4] FISHBURN J, FINKEL R. Parallel alpha-beta search on Arachne [R]. Madison, USA: University of Wisconsin, Department of Computer Sciences, 1980.

[5] PEARL J. Asymptotic properties of minimax trees and game searching procedures [J]. Artificial Intelligence, 1980, 14(2): 113 - 138.

[6] PLAAT A, SCHAEFFER J, PIJLS W, et al. A new paradigm for minimax search, TR-CS-94-18 [R]. Edmonton, Canada: University of Alberta, Department of Computing Science, 1994.

(下转第 1990 页)

易产生“过聚类”问题的原因,本文采用 MDS 算法的一个变种——WMDS 算法降低逼近精度相对较差的多边测地距离在 MDS 距离保持中的主导作用,更加精确地保持了那些逼近精度相对较好的少边测地距离,从而在一定程度上缓解了“过聚类”问题,得到了更好的可视化结果。

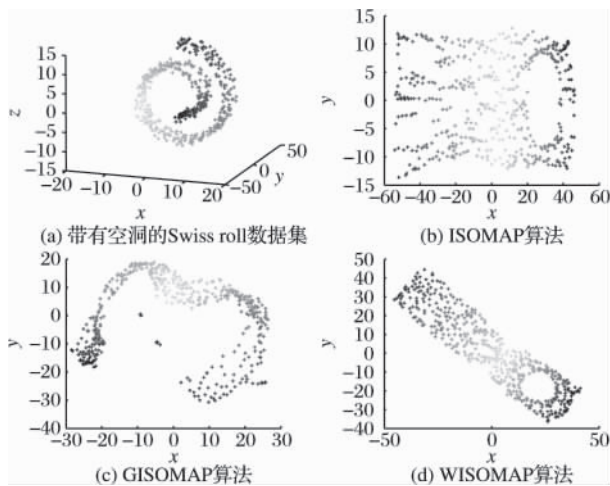


图5 不同算法对带有空洞的 Swiss roll 数据集的可视化结果($k=5$)

作为 ISOMAP 算法的一个变种,该算法的成功运行同样依赖于邻域大小参数的合适选取,并敏感于数据中存在的噪声,这将是今后改进的方向。

参考文献:

- [1] DUDA R O, HART P E, STORK D G. Pattern classification [M]. 2nd ed. New York: John Wiley & Sons, 2000.
- [2] HAUTANIEMI S, YLI-HARJA O, ASTOLA J, *et al.* Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps [J]. Machine Learning, 2003, 52(1/2): 45–66.
- [3] ZHANG C, WANG J, ZHAO N, *et al.* Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction [J]. Pattern Recognition, 2004, 37(2): 325–336.
- [4] DAWSON K, RODRIGUEZ R L, MALYJ W. Sample phenotype clusters in high-density oligonucleotide microarray data sets are revealed using ISOMAP, a nonlinear algorithm [J]. BMC Bioinformatics, 2005, 6(1): 195–202.
- [5] TENENBAUM J B, de SILVA V, LANGFORD J C. A global geometric framework for nonlinear dimensionality reduction [J]. Science, 2000, 290(22): 2319–2323.
- [6] DONOHO D L, GRIMES C. When does ISOMAP recover the natural parameterization of families of articulated images? [R]. Stanford: Stanford University, Department of Statistics, 2002.
- [7] LI Y. Distance-preserving projection of high-dimensional data for nonlinear dimensionality reduction [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(9): 1243–1246.
- [8] VLACHOS M, DOMENICONI C, GUNOPULOS D, *et al.* Non-linear dimensionality reduction techniques for classification and visualization [C]// Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2002: 645–651.
- [9] HADID A, PIETIKÄINEN M. Efficient locally linear embeddings of imperfect manifolds [C]// Proceedings of the 3rd International Conference on Machine Learning and Data Mining. Berlin: Springer, 2003: 188–201.
- [10] WU Y, CHAN K L. An extended ISOMAP algorithm for learning multi-class manifold [C]// Proceedings of 2004 International Conference on Machine Learning and Cybernetics. Piscataway: IEEE, 2004: 3429–3433.
- [11] 马瑞,王家康,宋亦旭. 基于局部线性嵌入(LLE)非线性降维的多流形学习[J]. 清华大学学报: 自然科学版, 2008, 48(4): 583–586.
- [12] 程起才,王洪元,刘爱萍,等. 基于 ISOMAP 的一种多流形学习算法[J]. 微电子学与计算机, 2009, 26(10): 115–117.
- [13] SAXENA A, GUPTA A, MUKERJEE A. Non-linear dimensionality reduction by locally linear ISOMAPs [C]// Proceedings of the 11th International Conference on Neural Information Processing. Berlin: Springer, 2004: 1038–1043.
- [14] LEE J A, LENDASSE A, VERLEYSEN M. Nonlinear projection with curvilinear distances: ISOMAP versus curvilinear distance analysis [J]. Neurocomputing, 2004, 57(1): 49–76.
- [15] KOUROPTOVA O, OKUN O, HADID A, *et al.* Beyond locally linear embedding algorithm [R]. Oulu: University of Oulu, 2002.
- [16] TROSSET M W. Extensions of classical multidimensional scaling: computational theory [J]. Computational Statistics, 2002, 17(1): 147–162.
- [17] BALASUBRAMANIAN M, SHWARTZ E, TENENBAUM J B, *et al.* The ISOMAP algorithm and topological stability [J]. Science, 2002, 295(5552): 7a–7.
- [7] PLAAT A, SCHAEFFER J, PIJLS W, *et al.* Best-first fixed-depth minimax algorithms [J]. Artificial Intelligence, 1996, 87(1/2): 255–293.
- [8] SLAGLE J R, DIXON J K. Experiment with some programs that search game trees [J]. Journal of the Association for Computing Machinery, 1969, 16(2): 189–207.
- [9] 张明亮,李凡长. 一种新的博弈树搜索方法[J]. 山东大学学报: 工学版, 2009, 39(6): 1–8.
- [10] 张明亮. 一种新的博弈树搜索算法及其应用研究[D]. 苏州: 苏州大学, 2007.
- [11] 张明亮,吴俊,李凡长. 极小树叶结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.
- [12] 叶品星. 一种博弈树静态估值算法—— Δ Feature 状态估值[J]. 计算机工程与设计, 2004, 25(7): 1214–1217.
- [13] 徐长明,马宗民,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(4): 514–517.
- [14] 王秀坤,刘健男. 优化博弈问题评估函数参数的自适应遗传算法[J]. 计算机工程与应用, 2009, 45(20): 42–44.
- [15] CAMPBELL M, HOANE A, HU F-H. Deep blue [J]. Artificial Intelligence, 2002, 134: 57–83.
- [16] 王晓鹏,王骄,徐心和,等. 中国象棋与国际象棋比较分析[J]. 重庆工学院学报: 自然科学版, 2007, 21(1): 71–76.
- [17] 莫建文,林士敏,张顺岚. 基于 TD 强化学习智能博弈程序的设计与实现[J]. 计算机应用, 2004, 24(S1): 287–288.
- [18] CRACRAFT S. Bitmap move generation in chess [J]. Journal of the International Computer Chess Association, 1984, 7(3): 146–153.
- [19] HYATT R M. Rotated bitmaps a new twist on an old idea [J]. Journal of the International Computer Chess Association, 1999, 22(4): 213–222.

(上接第 1972 页)