



中国人工智能系列白皮书 ——机器博弈

中国人工智能学会

二〇一七年五月

《中国人工智能系列白皮书》编委会

主 任：李德毅

执行主任：王国胤

副 主 任：杨放春 谭铁牛 黄河燕 焦李成 马少平 刘 宏
蒋昌俊 任福继 杨 强 胡 郁

委 员：陈 杰 董振江 杜军平 桂卫华 韩力群 何 清
黄心汉 贾英民 李 斌 刘 民 刘成林 刘增良
鲁华祥 马华东 马世龙 苗夺谦 朴松昊 乔俊飞
任友群 孙富春 孙长银 王 轩 王飞跃 王捍贫
王万森 王卫宁 王小捷 王亚杰 王志良 吴朝晖
吴晓蓓 夏桂华 严新平 杨春燕 余 凯 余有成
张学工 赵春江 周志华 祝烈煌 庄越挺

本书编写组

邱虹坤 王 骄 徐长明 李淑琴 李学俊 吴 愚
梅 险 高 强 杨放春 王亚杰 徐心和

全书统稿：王亚杰、邱虹坤

目 录

第 1 章 引言.....	1
第 2 章 机器博弈的发展状况	5
2.1 机器博弈历史	5
2.2 机器博弈研究现状	7
2.2.1 国外的研究现状	8
2.2.2 国内的研究现状	10
2.3 机器博弈产业现状	11
2.4 面临的问题与展望	12
第 3 章 博弈的复杂度	15
3.1 概述.....	15
3.2 博弈问题的状态复杂度及其估算方法	17
3.2.1 博弈问题的状态复杂度定义	17
3.2.2 博弈问题的博弈树复杂度	20
3.3 博弈问题的计算复杂性	24
第 4 章 机器博弈的典型技术	28
4.1 概述.....	28
4.2 穷尽搜索	28
4.3 裁剪搜索	29
4.4 启发式算法	30
4.5 迭代深化	31
4.6 最佳优先算法	31
4.7 随机搜索算法	32
4.8 并行计算	33
4.9 遗传算法	34

4.10 神经网络	35
4.11 机器学习	35
第 5 章 机器博弈的平台技术	38
5.1 计算机博弈平台概述	38
5.2 计算机博弈平台的分类	38
5.3 计算机博弈平台的设计规范	39
5.3.1 人机交互接口规范	39
5.3.2 平台与引擎通信的协议规范	40
5.4 计算机博弈平台的相关技术	40
5.5 计算机博弈平台应用实例	41
第 6 章 完备信息机器博弈的专项技术	43
6.1 六子棋机器博弈	43
6.1.1 概述	43
6.1.2 六子棋机器博弈主要技术	43
6.2 围棋机器博弈	46
6.2.1 概述	46
6.2.2 围棋机器博弈主要方法	47
6.3 点格棋计算机博弈	50
6.3.1 概述	50
6.3.2 点格棋机器博弈主要技术	51
6.4 爱恩斯坦棋机器博弈	54
6.4.1 概述	54
6.4.2 爱恩斯坦棋机器博弈主要技术	56
第 7 章 非完备信息机器博弈的专项技术	60
7.1 德州扑克机器博弈	60
7.1.1 概述	60

7.1.2 非完备信息动态博弈解的主要方法	61
7.2 军棋机器博弈	63
7.2.1 概述	63
7.2.2 军棋机器博弈主要技术	64
7.3 桥牌机器博弈	67
7.3.1 概述	67
7.3.2 桥牌机器博弈的主要方法	68
第 8 章 机器博弈的国内外赛事	72
8.1 国际机器博弈赛事	72
8.1.1 国际象棋人机博弈大赛	72
8.1.2 围棋人机与机机博弈大赛	73
8.1.3 桥牌计算机博弈大赛	75
8.1.4 德州扑克人机与机机博弈大赛	75
8.1.5 其它赛事与研究成果	76
8.2 国内机器博弈赛事	76
8.2.1 中国象棋人机与机机博弈大赛	76
8.2.2 中国计算机博弈大赛	77
第 9 章 结束语.....	79
参考文献.....	81

第 1 章 引言

计算机博弈也称机器博弈 (Computer Games)，英文直译应该是计算机游戏，其覆盖面非常广泛。然而，从事计算机棋牌竞技研究的科学家们，很早便将 **Computer Games** 定义为让计算机能够像人一样会思考和决策，能够下棋。为此还成立了 **International Computer Games Association** (ICGA—国际机器博弈协会)，专门组织世界范围内的棋类（后又加入牌类）博弈竞赛和学术交流。为了和计算机游戏区别开来，**Computer Games** 中文名字便称之为机器博弈，或者计算机博弈。负责组织全国范围内的棋牌类博弈竞赛和学术交流的群众组织便是中国人工智能学会下属的机器博弈专业委员会。

机器博弈的第一个里程碑成果是 1997 年 IBM 深蓝战胜世界棋王卡斯帕罗夫。虽然此项成果震动了世界，但在中国只是成为广为流传的消息，却没有在学术界引起足够的重视和兴趣。除去中国台湾之外，在中国大陆计算机博弈还只是极个别人的个人行为。当谷歌的 AlphaGo 战胜了围棋世界冠军李世石，AlphaGo 的升级版 Master 横扫了包括中国在内的 60 位世界顶尖高手，中国人才认识到机器博弈太可怕了，而我们在这一领域已经落后了太多。

人生如棋，世事如棋，而围棋又是各种棋类中公认最难以驾驭的棋种。计算机博弈在围棋上的骄人战绩，无疑显示出人工智能即将达到登峰造极的时代，人工智能将改变人类的生产和生活方式。人们开始研究 AlphaGo 的技术和能力，这也把计算机博弈推向了新的高度。

如果要盘点一下中国的计算机博弈，那最值得一提的是电脑围棋先行者中山大学化学系教授陈志行（1931—2008）。陈老先生 1991 年退休后潜心研究电脑围棋，在苹果机上用汇编语言编写了博弈程序《手谈》，并且赢得了 1995—1997 连续 3 年的 6 项世界冠军，成为机器博弈史上的一朵奇葩。当晚年的陈老得知国内组织了计算机博弈全国锦标赛时，兴奋不已，抱病参加了在重庆理工大学举行的 2007

年第二届全国锦标赛（不计名次），还亲自指导年轻人开发软件。陈老先生是我们的楷模，也是中国人的骄傲。

再值得提及的便是许峰雄博士。他台湾大学毕业后到美国卡内基梅隆攻读博士学位，特别钟爱国际象棋计算机博弈。他在毕业后到 IBM 公司组织了深蓝课题组，并在 1997 年以战胜卡斯帕罗夫的辉煌战绩赢得了世人的尊重。

计算机博弈在中国大陆虽然“迟到”、姗姗来迟，但发展非常迅速。2005 年，中国人工智能学会成立了机器博弈专业委员会，一批热心这一领域的科技工作者开始学习国际的先进理论与算法，很快便把国际象棋的算法移植到中国象棋的电脑程序当中，并取得了令人触目的成果。东北大学的棋天大圣代表队夺得了由 ICGA 组织的 2006、2007 年国际棋类奥林匹克大赛中国象棋冠军。而且，2006 年首届中国象棋计算机博弈锦标赛在北京科技馆成功举行，期间举办了人机大战，挑战了中国象棋的顶尖高手许银川、柳大华、徐天红、卜凤波等特级大师，并取得势均力敌的战绩，令国人刮目相看，但并未掀起预期的热潮，但却推动了全国锦标赛每年一届的举行。而且，在 2011 年还得到了教育部计算机类专业教学指导委员会的赞同，开始共同主办全国大学生计算机博弈大赛，使得比赛项目数不断增加，参加的队伍规模也不断壮大。竞赛棋种不仅有完备信息动态博弈项目，如中国象棋、围棋、点格棋、亚马逊棋等，还有非完备信息博弈项目，如幻影围棋、军棋等，还有考虑随机因素的爱因斯坦棋。自 2013 年起，还增加了多人博弈的扑克项目，如斗地主和桥牌，使得关于博弈算法的研究更加全面和深入。特别值得提及的在，2015 年得到国家体育总局棋牌运动管理中心支持，将该项比赛纳入 2015 年第三届全国智力运动会，进一步扩大了比赛的影响力和知名度。

十年来，计算机博弈在中国大地上蓬勃发展，很多院校开展了以机器博弈为内容的学生科技竞赛活动，很好地带动了校园科技活动的

氛围，也很好地培养了学生的科研能力和创新意识。这使得计算机博弈活动在中国从无到有，从小到大，至今可以说中国是世界上参与人数最多、比赛规模最大的国家。

这项比赛之所以能不断发展，最关键的原因就是计算机博弈有着强大的生命力，契合了大学生对棋牌游戏天然的兴趣和喜爱。表面上看，每届参与人数有限，这是受限于比赛规则和条件，比如规定每所高校在每个项目中最多只能报名 2 支队伍，加上高校参赛差旅费用限制。因此考虑到各个学校的选拔赛在内，保守估计实际能到场参赛的人数是实际参与这项科技活动人数的 1/5，那么，实际参加计算机博弈活动的人数将是数以千计的。计算机博弈的强大生命力主要源于以下方面：1) 参赛成本低，学生只要有一台电脑就可以开展研究，参加比赛；2) 没有专业限制，会下棋、或会编程，就可以参加这项活动；3) 下棋本身就是游戏，活动本身具有很好的高趣味、强吸引力；4) 比赛具有强挑战性和不确定性，比如这届胜了，并不能保证下届还胜，因此，研究没有止境、不是一蹴而就，是创新活动的不竭研究宝库；5) 下棋规则简单、输赢结果立判，不需要专家评审、打分，真正实现公开、公平、公正；6) 适合分工合作和团队作战，在研究与开发中使得学生的技能和素质得到全面锻炼；7) 有一定的网络化和产业化前景，很容易进入信息化和互联网+的项目当中；8) 培养学生创新能力、职业素养明显，极大促进学生就业能力。

此外，中国要想成为计算机博弈强国，就要加强博弈理论和算法的深入研究。为此，我们专委会在中国控制与决策学术年会（CCDC）上开辟了《计算机博弈特邀专题》，开展成果交流，提高我国在该领域的学术水平和学术影响。

本《机器博弈白皮书》是机器博弈宣传和普及工作的继续和深化，为此专委会邀请了这一领域的同行专家共同撰写。首先，介绍了机器博弈的发展过程、国内外赛事、博弈典型技术和比赛平台，然后结合

相关棋种介绍了各种专项博弈技术，既包括完备信息的棋类比赛，还包括不完备信息的牌类游戏搜索算法，当然还包括目前最先进的AlphaGo的深度学习算法、最新的桥牌和德州扑克博弈算法等。

机器博弈的产业化前景也是很可观的。AlphaGo的成功，标志着人工智能进入了新的阶段，深度学习算法得以在各个领域的广泛重视和应用。丰富多彩的博弈搜索算法无疑可以应用到面对决策优化的各种场合。随着不完全信息博弈、随机环境博弈搜索算法的不断完善，也将在兵棋推演和战略、战役和战术博弈中加以应用。博弈是人类经济、政治、军事、反恐、治霾和日常生活中无所不在的内容，机器博弈的概念和技术也必然大有用武之地。

让计算机博弈活动在更多的学校中生根、开花、结果，这是专委会一直追求的目标。让我们走出去，在国际大赛中夺取更多的奖牌，为国争光。同时，借助计算机博弈活动，促进产学研相结合，推动我国人工智能技术的发展，加速我国早日成为人工智能领域的大国、强国。

第 2 章 机器博弈的发展状况

2.1 机器博弈历史

早在人类文明发展初期，人们就开始棋类博弈的游戏。1928 年，被称作计算机之父的冯·诺依曼（John von Neumann）通过对两人零和一类博弈游戏的分析，提出了极大极小值定理，证明了博弈论的基本原理。在冯·诺依曼和摩根斯特恩合著的《博弈论和经济行为》（1944）中，将二人博弈推广到 n 人博弈结构，并将博弈论系统应用于经济领域，从而奠定了机器博弈研究的基础和理论体系。

近代计算机博弈的研究，是从上世纪五十年代开始的。许多世界上著名的科学家，例如数学家和计算机学家阿兰·图灵（Alan Turing），信息论创始人科劳德·香农（Claude E. Shannon），人工智能的创始人麦卡锡（John McCarthy）以及冯·诺依曼等人都曾经涉足计算机博弈领域的研究工作，并为之做出过非常重要的贡献。

1950 年，著名的控制论先驱香农提出了象棋博弈的编程方案。1953 年，阿兰·图灵设计了一个能够下国际象棋的纸上程序，并经过一步步的人为推演，实现了第一个国际象棋的程序化博弈。1958 年，IBM 推出取名“思考”的 IBM704，成为了第一台与人类进行国际象棋对抗的计算机。虽然在人类棋手面前被打得丢盔卸甲，但许多科学家却对此欢欣鼓舞。1959 年，人工智能的创始人之一塞缪（A. L. Samuel）编写了一个能够战胜设计者本人的西洋跳棋计算机程序，1962 年该程序击败了美国的一个州冠军，这是计算机博弈历程中一个重要的里程碑。

随着计算机硬件和软件技术的不断发展，通过人-机或者机-机对弈，实现了计算机硬件性能和计算机软件水平的较量。科学家们开始对电脑能否战胜人脑这个话题产生了浓厚的兴趣，提出以棋类对弈的方式，向人类智能发起挑战。

上世纪八十年代中期，美国卡耐基梅隆大学开始研究世界级的国际象棋计算机程序；1988-1989 年间，IBM “深思” 分别与丹麦特级大师拉尔森、世界棋王卡斯帕罗夫进行了 “人机大战”。

从上世纪九十年代起，Tesauro 的 TD-GAMMON 西洋双陆棋程序经过上百万盘的学习训练，程序达到世界水平；“深思” 二代产生，吸引了前世界棋王卡尔波夫和世界优秀女棋手小波尔分别前来与之对抗（1990 和 1993 年）。特别是 “深蓝”（1996 年）、“超级深蓝”（1997 年）与卡斯帕罗夫的两场比赛，引起全球媒体的关注。在随后的几年里，计算机与卡斯帕罗夫和克拉姆尼克等世界顶级棋手进行了一系列的比赛，计算机逐渐负少胜多，表现得越来越聪明。

经过多年对计算机博弈进行系统的理论研究，在国际象棋、中国象棋等棋种的人机大战中，从最初人类完胜电脑，到如今电脑击败人类顶级高手，计算机博弈水平迅速上升。特别是，2016-2017 年，AlphaGo 分别与李世石、柯洁的人机围棋大战并取得胜利，这可谓是人机对抗史上的最强之战，从而掀起全球人工智能热潮。

此外，除了 AlphaGo 完备信息机器博弈领域的人工智能划时代成就外，2007 年 1 月 30 日，美国卡耐基梅隆大学开发的德州扑克博弈系统 Libratus 与 4 名人类顶尖德州扑克选手之间进行了 “人机大战”，宣告在 “多人” 博弈的非完备信息机器博弈领域，人工智能同样取得了胜利。2017 年 4 月 6~10 日，备受关注的亚洲首度人工智能与真人对打的扑克大赛——“‘冷扑大师’（Libratus 扑克机器人）与中国龙之队” 扑克巅峰表演赛在海南收官，最终以 “冷扑大师” 获胜，赢得 200 万元奖金，这是人工智能在各种棋牌博弈中对人类取得的又一个胜利。再次在全球范围加剧了对人工智能的敬畏或恐惧气氛。

在中国国内，也有一些学者从事计算机博弈方面的研究。比如，南开大学黄云龙教授和他的学生吴韧在上世纪八十年代开发了一系

列中国象棋程序；中山大学化学系教授陈志行先生在九十年代初开发了围棋程序“手谈”，并获得世界冠军；本世纪初，东北大学的徐心和教授和他的学生王骄、徐长明等研究开发了中国象棋软件“棋天大圣”，并在 2006 年的人机大战中，展现了具有挑战国内中国象棋顶级高手的实力，表现出较高的智能；南京航空航天大学夏正友教授指导学生研究开发了具有一定智能的四国军棋博弈系统；北京邮电大学的刘知青教授带领学生开发的“本手（LINGO）”围棋程序，能够战胜具有一定水平的业余围棋选手；哈尔滨工业大学王轩教授的团队开发的德州扑克博弈系统，2013-2016 年间多次参加 ACPC 二人非限制性、三人及多人德州扑克比赛，均进入决赛前 4 名。

这项活动最重要的事件是由东北大学徐心和教授发起成立的中国人工智能学会机器博弈专业委员会，以及从 2006 年起每年一届的中国大学生计算机博弈大赛暨中国计算机博弈锦标赛^[1]，至今已经举办 12 届。沈阳航空航天大学王亚杰教授在国内大力推广计算机博弈活动，规范比赛规则，吸引越来越多高校师生参与计算机博弈相关研究中来，中国计算机博弈进入了快速发展阶段^[2]。特别是教育部高等学校计算机类专业教学指导委员会参与进来成为共同主办单位，这些对我国计算机博弈技术的研究与发展起到了极大的促进作用。

近几年来，国内许多企业如腾讯、百度、联众、新睿等纷纷加入到机器博弈的研究大军中，开发出一些具有较高智能水平的产品。如腾讯人工智能实验室（AI Lab）研发的围棋人工智能程序“绝艺”（Fine Art），夺得 2017 年第 10 届 UEC 杯计算机围棋大赛冠军；北京邮电大学的刘玉璋和杨放春教授带领创业团队开发的新睿桥牌机器人，在 2017 年第 21 届世界计算机桥牌锦标赛中获得亚军。

2.2 机器博弈研究现状

DeepMind 公司创始人 Demis Hassabis 曾言：“游戏是测试人工智能算法的完美平台”。而计算机博弈被誉为是人工智能学科的“果

蝇”，通过机器博弈的过程来理解智能的实质，是研究人类思维和实现机器思维最好的实验载体。

2.2.1 国外的研究现状

在计算机博弈研究的早期阶段，研究的主要内容涉及如何建立有效、快速地评价函数和评价方法，使评价的效率更高，花费的时间和空间的代价更小，以及如何在生成的博弈树上更准确有效地找到最优解，并由此衍生出搜索算法的研究成果。在随后的几十年里，专家和学者们在计算机博弈搜索与评估方面进行了大量深入探索和实质性的研究，产生了许多计算机博弈技术，如极大极小搜索、负极大值搜索、Alpha-Beta 剪枝、并行搜索算法等^[3-7]。

特别值得讲述的是，2006 年多伦多大学教授 Geoffery Hinton 发表文章提出了基于深度信念网络（Deep Belief Networks, DBN）可使用非监督的逐层贪心训练算法^[8,9]，在学术界掀起了对深度学习（Deep Learning）的研究热潮。随着并行计算、基于人工神经网络的深度学习^[10,11]等技术的突破性进展，成功解决了机器博弈中抽象认知的难题。使得深度学习等技术被成功应用于计算机博弈及相关领域中，从而将机器博弈水平带上了一个新的台阶。

Google、百度等国际大公司争相跟进，研发出相关的机器博弈产品。尤其是 Google 公司的围棋软件 AlphaGo，作为完备信息博弈代表，它具有极强的自学能力，如图 2-1 显示了 AlphaGo 的算法组成。

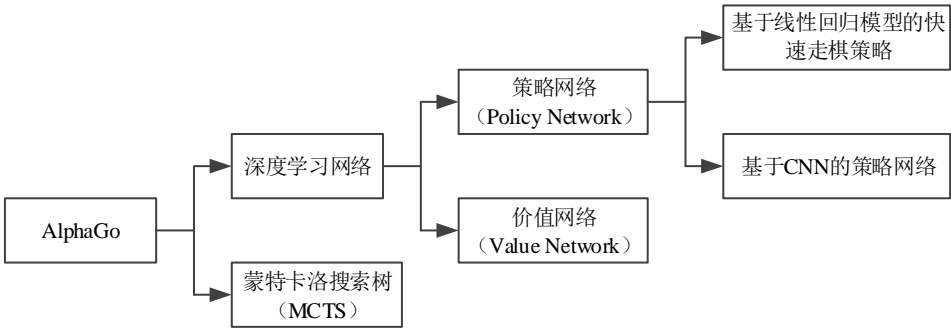


图 2-1 AlphaGo 算法组成

AlphaGo 的技术突破被学者们总结为两个关键技术：棋感直觉和搜索验证。其中，棋感直觉通过深度学习（Deep Learning）获得，它分为落子棋感和胜负棋感，AlphaGo 通过对 3000 万的经典棋局进行深度学习得到快速走棋网络和策略网络，而快速走棋网络就是落子棋感；胜负棋感是通过深度学习得到的策略网络不断进行自对弈来得到；搜索验证指的是搜索引擎主要采用蒙特卡洛搜索树根据落子棋感和胜负棋感不断展开搜索树^[12]。围棋程序 AlphaGo 的成功充分验证了深度学习与计算机博弈技术结合的实用性。Google 公司宣布将其应用于医疗诊断等领域，以扩大深度学习应用领域。

此外，不完备信息博弈以美国卡耐基梅隆大学研发的德州扑克博弈系统 Libratus 为代表，被学者们总结为主要包括三个关键模块：

（1）赛前纳什均衡近似（Nash equilibrium approximation before competition）。这个模块把最重要的博弈信息（例如针对某一手牌对应的战略）进行抽取，然后再应用强化学习等方法，继续寻求提高和改进。这里使用了一个新的算法：蒙特卡洛反事实遗憾最小化。在这个模型的帮助下，Libratus 自己学会了德州扑克，而且比以前速度更快。

（2）残局解算（Endgame solving）。这是 Libratus 最重要的部分，因为一局德扑只需要几个回合，耗费时间短。因此 Libratus 的开发者们选择从下往上构建博弈树，这样最下面节点的状态是容易算出来的，用这个状态反过来指导设计上面的博弈树，并使用蒙特卡罗方法，每次选一些节点去更新它们上面的策略。也就是说，Libratus 不仅仅是在比赛前学习，而且还能在比赛中学到东西。

（3）持续自我强化（Continual self-improvement）。比赛中人类高手会寻找 Libratus 的漏洞，并展开有针对性的攻击。这个模块的作用就是发现问题所在，找到更多细节进行自我强化，然后得到一个更好的纳什均衡。

2.2.2 国内的研究现状

国内计算机博弈研究相对起步较晚，曾经一度参与者寥寥无几，参考文献匮乏，计算机博弈氛围沉寂。本世纪初，东北大学徐心和教授带领团队对中国象棋、六子棋、点格棋等机器博弈展开了深入研究，在国内高校举办各类计算机博弈技术培训与讲座，申请控制与决策国际会议（CCDC）机器博弈专题，极大地促进了国内计算机博弈的研究与发展。

近几年，学者们结合棋牌各自特点，针对计算机博弈搜索、评估与优化等方面开展了深入研究。例如，对于局面估值问题，文献[13]提出结合时间差分算法和反向传播神经网络，设计一种局面估值算法实现评估函数参数的自动调整；针对六子棋，东北大学徐长明对局面表示等关键技术进行研究^[14,15]，重庆理工大学张小川教授提出应用遗传算法优化评估函数^[16]，安徽大学李学俊教授等提出基于局部“路”扫描方式的博弈树生成算法^[17]；针对中国象棋，东北大学王骄教授对计算机博弈开局库^[18]、循环判定规则^[19]、评估函数参数组的自动调整和优化^[20]展开了研究；针对苏拉卡尔塔棋，北京信息科技大学李淑琴教授等根据棋子的数量、移动范围等不同参数，对局面评估函数进行了研究^[21]；辽宁石油化工大学张利群教授提出了一个网络博弈平台吃子算法^[22]；针对亚马逊棋，沈阳航空航天大学邱虹坤等对搜索算法进行了分析^[23-24]，李淑琴教授根据特征权重值给出一个分阶段的评估函数^[25]；中央民族大学李霞丽和吴立成教授提出一种围棋多模态算法，及基于小样本的藏棋博弈算法^[26,27]；哈尔滨工业大学王轩教授团队针对德州扑克等博弈系统，开展了非完备信息机器博弈中风险及对手模型的研究^[28-30]。

AlphaGo 和 Libratus 等机器博弈 AI 的成功，引发了国内外学者和企业广泛地关注与研讨。国内知名企业如百度、腾讯等设计并实现了具有深度学习能力的机器博弈 AI。

总之，经过国内外学者们多年努力，对于状态空间和搜索树复杂度都较大的完备信息人机博弈（例如围棋、中国象棋）而言，经过学习训练，计算机可以战胜人类顶级棋手。在二人零和完备信息博弈研究方面，尤其是关于深度学习技术的研究与运用，机器博弈为人工智能提供了很多重要方法和理论。此外，在非完备信息人机博弈方面，德州扑克项目也取得了战胜人类的惊人成果，达到了新高度。

2.3 机器博弈产业现状

近几年，机器博弈不仅在学术界掀起了对其研究的热潮，还带动与之高度密切相关的游戏产业飞速发展。人工智能不仅提高了游戏的趣味性，还使游戏变得更精致。

从 2001 年到 2017 年，游戏产业从冬眠期醒来，经历了起步、成长、成熟的过程。根据产业调查显示，全球游戏市场总体增速在 10% 左右，主要驱动力在手机游戏。2015 年移动游戏收入占移动应用市场总收入的 85%，达到 348 亿美元，2016 年移动游戏的收入达到 415 亿美元，预计 2020 年将增至 746 亿美元。

相对于国外，我国游戏产业发展较快，2014 年中国游戏的营业收入已超过 1100 亿元，是电影票房的 3 倍以上，其游戏收入的 63.5% 源自国产游戏。2016 年中国游戏整体营收超过 1400 亿人民币，超过美国和日本，成为世界最大游戏市场。另外，国家对电子竞技的发展给予了大力支持，除了将电子竞技纳入第 99 个体育项目外，还将电子竞技纳入教育部增补专业。

在机器博弈领域，学术界与产业界结合日趋紧密。庞大的机器博弈产业吸引了众多公司争相跟进，企业积极与从事机器博弈领域研究的专家学者展开多方位的合作，将学者们的科研成果转化为具有更高人工智能水平的产品。比如北京邮电大学刘玉璋教授创建的新睿桥牌机器人，开启了崭新的计算机游戏新商业模式，已经发展成为全球最大的桥牌游戏平台，目前他们又引进重庆理工大学张小川教授带领团

队研制的斗地主全国冠军引擎，旨在具有最广泛群众娱乐活动的斗地主游戏中，探索机器博弈发展的商业新模式，为机器博弈发展注入新动力。

除了游戏产业，机器博弈技术还逐步被应用到智慧医疗、智能交通、航空、航天等相关产业中，特别是与军事国防领域的产业相结合，催生出许多新型智能武器与系统。例如：军用无人机通过引入机器博弈等相关人工智能技术，操控方式可由传统的遥控逐渐过渡到系统自主智能决策与控制^[31]；在智能化战争模拟（兵棋推演）系统中，可以自动模拟敌方复杂的行为，更加真实地模拟战争的场景，通过虚拟仿真推演军事对抗，将其结果作为决策系统重要可信的依据^[32-34]；在以理解、推理、决策为代表的军用信息系统中，结合深度学习与机器博弈相关技术，推动系统向智能化发展^[35]。

因此，随着机器博弈与相关领域产学研相结合，机器博弈技术真正进入实用阶段，在我国智能化建设中展示了巨大的潜在应用价值。大批机器博弈科研人才将成为民用、军工企业发展的强大技术引擎，对引领未来的机器博弈相关产业的发展，必将产生深远影响。

2.4 面临的问题与展望

在过去的几十年里，尽管机器博弈研究成果对于推动人工智能的发展具有重要意义，但在计算机博弈领域仍存在不同程度的局限性。具体表现为：

（1）在学术研究方面，尽管深度学习等技术在围棋方面取得了前所未有的成功，但在其应用拓展方面，仍有许多值得研究和探索之处。另外，对于具有模糊性和随机性的麻将、桥牌、斗地主、多国军旗等非完备信息博弈，虽然在基于案例的策略研究方面有了一定进展，但因其相关研究还不成熟，开发的程序智力有限，目前仍难以战胜人类顶级高手，尚有很大的提升空间。

（2）在相关技术产业化方面，产学研结合仍有不足。表现为：

一方面，相关企业缺乏机器博弈领域的专业人才，特别是缺少机器博弈领域顶级专家的技术支持；另一方面，机器博弈领域专家、学者们缺少相关部门、企业给予的研发资金支持。

从学术研究和相关技术产业化来看，我国对机器博弈技术的研发与应用相对于国外存在较大的差距。开发具有高效自主学习能力与抽象思维能力的智能博弈系统，特别是在非完备信息和不确定性机器博弈方面，还有很长的路要走。此外，只有将机器博弈技术作为战略核心予以关注，不断加大投入，在未来竞争中，我国才能处于不败之地。

国务院近期印发的《新一代人工智能发展规划》给我国人工智能发展指明了道路，为机器博弈发展注入新的活力并带来更多机遇，规划中明确提出：开展综合深度推理与创意人工智能理论与方法、非完全信息下智能决策基础理论与框架、数据驱动的通用人工智能数学模型与理论等研究；支持开展人工智能竞赛。相信在国家层面政策的支持下，我国计算机博弈领域的研究与应用将进入快速发展的新阶段。未来计算机博弈将呈现多学科技术融合、人机协同、产学研相结合等趋势^[36]。具体体现在：

（1）计算机博弈研究的内容将不断拓宽，处理的问题复杂程度越来越高，信息量将越来越大。为解决某类特定问题，技术方法将集成化，计算机博弈技术将与并行计算、大数据、知识工程等相关技术紧密结合。

（2）计算机博弈软件与硬件的结合越来越密切，固化博弈系统的智能硬件产品将越来越多的出现在人们的生活中，典型的应用包括：有博弈思维能力的机器人、智能决策控制系统的无人驾驶汽车和无人机等。

（3）计算机博弈将融入各个领域的应用中，在此基础上可以开展一系列人工智能领域的科学研究。计算机博弈越来越注重实际工程应用，紧密地结合经济、医疗、航空航天等领域，解决实际问题。特

别在航空航天领域的多学科协同综合设计、虚拟现实仿真及人机交互智能游戏与教育方面，拥有广阔的应用前景。

（4）计算机博弈技术将呈现高度智能化趋势，通过与遗传算法、人工神经网络、类脑思维等人工智能技术进一步融合，类似基于神经网络深度学习的智能技术将大量涌现，使得计算机博弈程序的类脑智能越来越高。

（5）合理拓展现有博弈技术，深入研究更加智能的普适算法，构建一个通用计算机博弈系统，将成为未来计算机博弈研究的重点。

（6）作为计算机博弈技术交流与验证的平台，中国计算机博弈比赛将越来越被社会所认同。各种新技术将会被越来越多地运用到计算机博弈中。

（7）学术界与产业界的结合日趋紧密，计算机博弈研究学术成果加速向产业化转变，助力游戏开发、智能医疗、航空航天企业，促进计算机游戏、智能医疗、航空航天、国防等相关产业发展。

可以预见，在计算机博弈领域越来越多的人机博弈项目中，人类终将被战胜。机器智能的胜利，既是人类创造力与智慧的结晶，也是科学发展的必然，同样也是人类最终的胜利。

第3章 博弈的复杂度

3.1 概述

计算机博弈属于人工智能领域的一个重要分支。计算机的博弈水平代表了计算机的智能水平。计算机博弈问题的状态复杂度和博弈树复杂度是衡量其复杂程度的两个重要标准。图 3-1 给出了求解博弈问题一般采取的四种策略。比如，博弈问题常用到的剪枝算法，它是为了避免分支过于庞大而采取的一种策略。这种策略可以节省计算机的内存空间，提高搜索效率，但也存在一定的风险，即如果估值函数不能准确地评估局面的话，这种算法可能将存在最佳着法的分支剪掉。当然，若博弈树复杂度比较小，就可以采用蛮力搜索的方式，只要时间允许，就可以找到最佳着法。

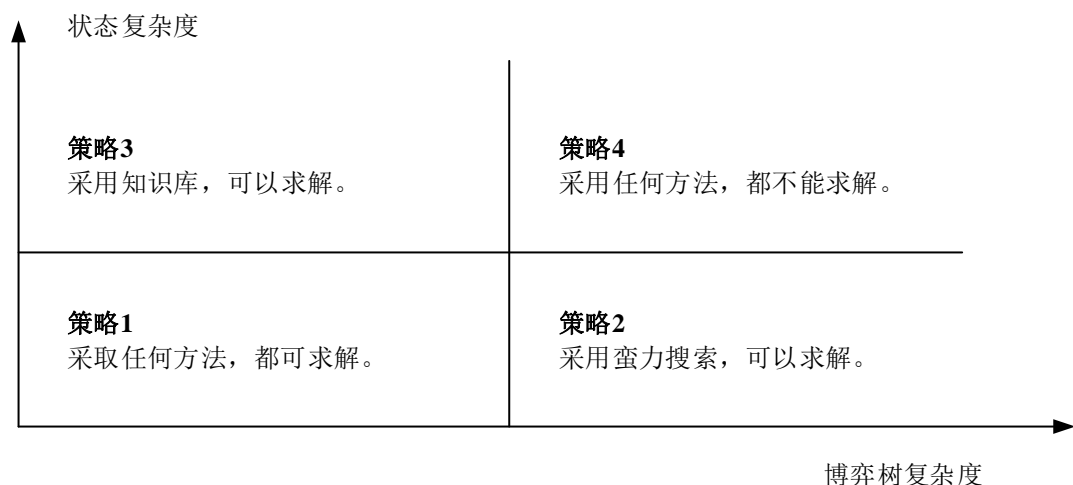


图 3-1 博弈问题的复杂度与求解策略之间的关系

文献[37]讨论了博弈问题的状态复杂度及其博弈树复杂度对寻找博弈问题理论解的意义，认为一个较低的状态复杂度比一个较低的博弈树复杂度对求解博弈问题所起的作用更大，因为状态复杂度为通过完全列举求解博弈问题的复杂度提供了一个边界值；而较低的博弈树复杂度主要是对搜索效率产生较大的影响。文献[37]还列出了一些

常见博弈问题的状态复杂度及博弈树复杂度（见表 3-1）。

表 3-1 一些博弈问题的状态复杂度和博弈树复杂度

棋种	状态复杂度	博弈树复杂度
西洋跳棋 (Checkers)	10^{21}	10^{31}
国际象棋 (Chess)	10^{46}	10^{123}
中国象棋 (Chinese Chess)	10^{48}	10^{150}
日本将棋 (Shogi)	10^{71}	10^{226}
围棋 (Go) (19×19)	10^{172}	10^{360}

文献[37]中，以 tic-tac-toe（三子连珠棋）为例，估算了此博弈问题的状态复杂度和博弈树复杂度。tic-tac-toe 共有 9 个位置可以落子，能够形成的局面较少，因此其复杂度的估算相对容易，具体估算过程如下：

（1）对于其状态复杂度，由于棋盘上每个位置有三种状态（双方的棋子和空白），因此，状态复杂度可估算为 3^9 ，根据此博弈问题的走棋规则，在棋盘上形成连 3 则游戏结束，出现两个以上的连 3 的局面属于非法局面。而对称相同的多个局面应该只算作一个局面。将这些考虑在内，则更精确的状态复杂度为 5478；

（2）对于其博弈树复杂度，平均深度约为 9，第 i ($1 \leq i \leq 9$) 层时，走棋方可能的走法有 $9-i$ 个，因此，此博弈树的叶子节点数（即博弈树复杂度）为 $9!$ 。

计算机博弈的最高境界就是找到该棋种的理想解，即不败解。而计算机博弈的最大困难和无法逾越的障碍则是问题的计算复杂性。对问题的计算复杂性进行分类，可以了解该问题被求解的难易程度，如果问题被证明是难解的（比如 NP-complete、PSPACE-complete 及 EXPTIME-complete），则不必将大量的精力花费在寻找问题的理论解上，而只能去寻求某种近似解。事实上，当前多数人工智能问题都

是追求满意解，大部分原因就是问题的复杂度远远超过当前计算设备能以合理性价比的求解代价，从而推动着各类博弈算法、策略的研究。

3.2 博弈问题的状态复杂度及其估算方法

3.2.1 博弈问题的状态复杂度定义

博弈过程的局面称之为状态，博弈问题的状态复杂度是指从初始局面出发，产生的所有合法局面的总和。然而，精确计算博弈问题（比如：国际象棋、围棋等）的状态复杂度几乎是不可能的^[38]。一般以该棋类可能的局面总数的上限值为标准。它为通过完全列举求解博弈问题的复杂度提供了一个边界值。

8×8 的西洋跳棋（Checkers）于 2007 年得到了理论解^[39]，证明过程中，采用了三种方法：证据计数法，残局阶段采用了数据库，通过两个程序实现对节点的估值。不仅证明了一种不败的策略，而且计算了 8×8 的西洋跳棋可能会产生 500,995,484,682,338,672,639（约 5×10^{20} ）个合法局面。可见，只有得到了理论解的博弈问题，才能比较精确地计算其状态复杂度。估算博弈问题的状态复杂度，与各个博弈问题的走棋规则密切相关，下面以亚马逊和苏拉卡尔塔棋为例，估算这两个博弈问题的状态复杂度。

3.2.1.1 亚马逊棋的状态复杂度

1. 亚马逊棋的走棋规则

棋盘与棋子（棋盘见图 3-2）：

- 1) 棋盘规模为 10×10 个方格，相邻两个方格填充的颜色不同；
- 2) 比赛双方各有 4 个棋子；每个棋子都相当于国际象棋中的皇后，它们的走棋规则与皇后相同^[40]。

走棋规则：

- 1) 每次开局由位于棋盘下方的玩家先手；
- 2) 当轮到一方走棋时，此方只能而且必须移动 4 个棋子中的一个，在移动完成后，该棋子必须释放一个障碍，障碍的释放方法与棋

子的移动方法相同（不能穿过障碍）；

3) 当某个走棋方在完成一次移动后，对方的 4 个棋子均不能再移动时，对方输掉比赛。

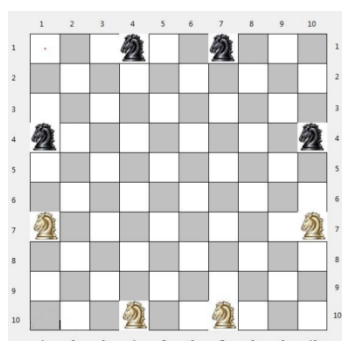


图 3-2 亚马逊棋棋盘

2. 估算亚马逊棋的状态复杂度

如图 3-2 所示，亚马逊棋棋盘的大小为 10×10 ，共 100 个格，双方棋子的走法与国际象棋的皇后相同（就是说棋子可以移动到任意一个格子上），因此每个格子可以出现白方棋子、黑方棋子、障碍、空白四种情况，所以亚马逊棋的状态复杂度可以估算为 4^{100} ，以 10 为底的形式表示，大小约为 10^{60} 。在产生的这些局面中，有很多非法的局面，根据亚马逊棋的规则，棋盘上双方各有 4 个棋子，而这种估算方法，显然有很多局面中出现的棋子数超过了 8 个。

这里从另一个角度来估算其状态复杂度，由于棋盘上始终存在双方的各 4 个棋子，也就是说棋盘上有 8 个交叉点需要用来放置这 8 个棋子，而棋盘上的其他交叉点可能的状态为空白或障碍，因此亚马逊棋的状态复杂度可估算为 10^{41} 。

3.2.1.2 苏拉卡尔塔棋的状态复杂度

1. 苏拉卡尔塔棋棋规

棋盘与棋子（见图 3-3）：

（1）横竖各 6 条边构成正方形棋盘，36 个交叉点为棋位，各边

由 8 段圆弧连接，通常用 2 种不同颜色表示。

(2) 红黑双方各 12 枚棋子。初始状态：棋子各方底线排成 2 排。

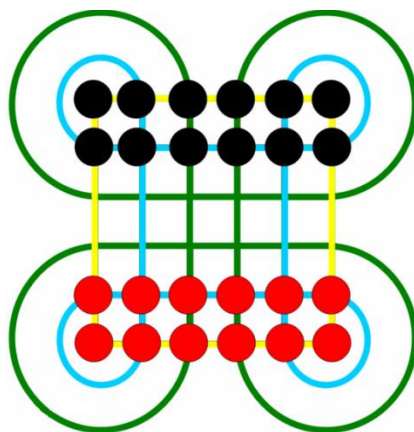


图 3-3 苏拉卡尔塔棋棋盘

走棋规则：

- (1) 双方轮流走棋，每次走动一枚棋子；
- (2) 除了吃子之外，每枚棋子只能沿着垂直、水平或对角方向走动一格，只能走向空位；
- (3) 吃对方棋子时必须经过至少一个完整的弧线^[41]；
- (4) 赢棋标准：吃掉所有对方棋子或最后剩余棋子多的一方获胜。

2. 估算苏拉卡尔塔棋的状态复杂度

棋盘上总共有 36 个交叉点，每个交叉点有三个可能的状态（即双方的棋子或空白），所以苏拉卡尔塔棋的状态复杂度可估算为 3^{36} ，以对数值（10 为底）的形式表示，则大小约为 17.2，这些局面中存在一些非法的局面，比如：根据规则，棋盘上最多只能出现 24 个棋子，而以这种估算方法，显然有一些局面中棋子的个数已经超过了 24 个。因此，可以换一种角度来估算，根据苏拉卡尔塔的规则，初始状态时，双方各有 12 个棋子，棋盘上最多会出现 24 个棋子，因此应该将棋盘上出现 36 个棋子、35 个棋子、...、25 个棋子的这些局

面去除掉。因此，苏拉卡尔塔棋更精确的状态复杂度以对数值（10为底）的形式表示，则大小约为 16.9。

3.2.2 博弈问题的博弈树复杂度

3.2.2.1 博弈树搜索算法原理

对于完备信息的博弈系统，其搜索一般都是基于博弈树搜索算法，如图 3-4 所示，这样可以展开双方所有可能走法产生的局面（博弈树中的节点），再通过估值函数来评价各个局面的优劣，从而可以找到最佳走法。

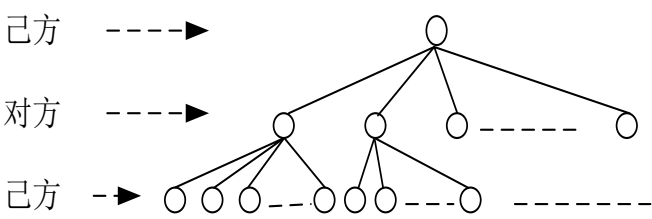


图 3-4 一个博弈树实例

博弈树是一颗根在上叶在下的树，分若干层，每一层代表某一走棋方可以走出的合法局面。对于只添子的棋类博弈系统（如五子棋、六子棋），博弈树中各个节点（局面）互不相同（这里指在一盘棋局中节点的状态，若交换先后手，则存在完全相同的节点）；对于存在移动棋子的棋类博弈系统（如国际象棋、中国象棋），博弈树中就会存在完全相同的节点（局面）。

3.2.2.2 博弈树复杂度的定义

博弈树复杂度是指从初始局面开始，其解决树（Solution tree，解决树是指得到理论解所需展开的最小搜索树）的所有叶子节点的总和^[38]。精确计算出博弈问题（如：国际象棋）的初始局面解决树的所有叶子节点总和几乎是不可行的，若能实现，那实际上就已经找到了该博弈问题的理论解（国际象棋被证明是难解的^[42]）。因此，对于比较复杂的博弈问题（如：围棋、中国象棋、亚马逊棋、六子棋等），一

一般采用平均分支因子^[43]的方法来估算博弈树复杂度。文献[44]阐述了如何采用平均分支因子的方法来估算国际象棋、围棋、日本将棋的博弈树复杂度。对于相对简单（如：**tic-tac-toe**）的博弈问题，可以采用蛮力搜索来精确地计算其博弈树复杂度。下面，以六子棋和点格棋为例，估算这两个博弈问题的博弈树复杂度。

3.2.2.3 六子棋的博弈树复杂度

1. 六子棋棋规

棋盘与棋子（如图 3-5 所示）：

（1）国际比赛中，六子棋棋盘规模为 19×19 ，共 361 个交叉点，初始状态时棋盘上无任何棋子；

（2）六子棋的棋子与围棋的棋子一样，只有黑、白两种颜色的棋子。

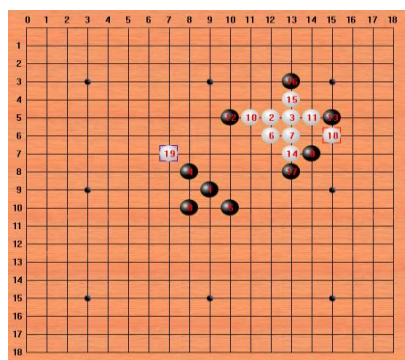


图 3-5 六子棋棋盘

走棋规则：

（1）黑方先手，第一步只下一个子，随后每一方须下两子；

（2）先走出同色连六（方向可为横、纵、斜 45° 、斜 135° ）的一方获胜^[45]。

不难看出走棋方能够下的点比较多，若完全采用博弈树展开的话，产生的节点数是巨大的。这严重影响了系统的执行效率。因此六子棋的搜索算法一般采用 **VCF**（**Victory of Continuous Fours**）迫着算法（与五子棋类似的搜索算法）。但由于 **VCF** 算法对于复杂的局面

考虑的不够全面，所以对于特殊局面（估值很高或很低的局面）可以适时地采用基于博弈树的搜索算法进行深层次的搜索以获得更好的走法。

2. 估算六子棋的博弈树复杂度

根据六子棋的走棋规则，不存在吃子的情况，只向棋盘添子，因此不同的走法所产生的局面互不相同。走棋方从第二步开始，每次下两子形成一个局面。因此六子棋博弈树复杂度的计算方法如下：

根据文献[45]，六子棋的博弈树搜索平均深度为 30。在游戏开始时，第一步有 361 个点可下，第二步有 C_{360}^2 个点可下，以此类推，一直到它的平均深度，即第 30 步，有 C_{302}^2 个点可下。其中应排除棋盘上所有的连珠数等于六的情况（六子连珠为六子棋获胜的条件），这里包括黑子或白子在横向（出现同色六子连珠的局面数为 $14 \times 19 \times 2$ ）、纵向（出现同色六子连珠的局面数同横向）及所有斜 45° 和 135° 线（出现同色六子连珠的局面数为 $(1+2+3+4+5+6+7+8+9+10+11+12+13+14) \times 2 \times 2$ ，即 420）上所有产生同色六子连珠的局面，即棋盘上可能出现的连六总数为 $\text{Count}(6)=952$ 。为了排除博弈树中出现的这些非法局面，我们将博弈树各层产生的结点总数都减掉 $\text{Count}(6)$ 。

根据六子棋的走棋规则，从第六层开始，棋盘上存在出现六子连珠的情况，因此从第六层开始，每层都减去 $\text{Count}(6)$ ，具体计算如下：

第六层产生的节点数为：

$$\text{Countpoints}(6) = 361 \times C_{360}^2 \times \cdots \times C_{352}^2 - 952$$

第七层产生的节点数为：

$$\text{Countpoints}(7) = \text{Countpoints}(6) \times C_{350}^2 - 952$$

以此类推，第 n 层产生的节点数为：

$$\text{Countpoints}(n) = \text{Countpoints}(n-1) \times C_{352-(n-6) \times 2}^2 - 952$$

因此，经计算得出六子棋的博弈树复杂度的上限值为 10^{145} 。

3.2.2.4 点格棋的博弈树复杂度

1. 点格棋的棋规

棋盘和棋子（如图 3-6 所示）：

（1）对于 6×6 点格棋，棋盘的初始状态为 36 个点，而没有任何连线；

（2）点格棋没有棋子的概念，完成一盘点格棋的对局需要 60 个边（邻近两点连成的一个边）。

走棋规则：

（1）某方走棋会占至少一个边，所以邻近两个点之间有两种可能（有或无边），不可越点，不可重复连线；

（2）当一个格子的四条边均被占满，则最后一个连线者获取这个格子，并且该走棋方将继续走棋（选择某两个邻近点进行连线）；

（3）走棋的结束标志是该走棋方连线后未获取格子；

（4）游戏结束的标志：所有的邻近点均被连线，也就是说所有的格子被俘获。占领格子更多的一方获胜^[46]。

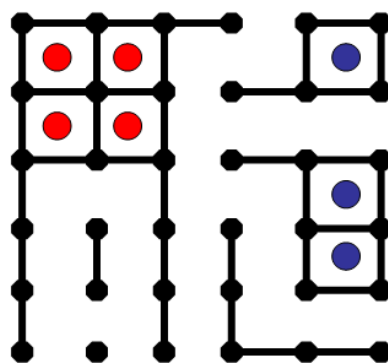


图 3-6 点格棋棋盘

2. 估算点格棋的博弈树复杂度

估算点格棋的博弈树复杂度的方法与 tic-tac-toe 类似。根据点格棋的规则，双方交替在 6×6 棋盘上的邻近两点之间连一条直线，最终将完成所有邻近两点的连线（共 60 个），即比赛结束。也就是在

比赛开始时，某走棋方可以在 60 个连线中，选择一个，从而完成第一步走棋；接下来，另一走棋方能够在剩下的 59 个连线中，选择一个来完成走棋，以此类推，即使根据规则，某一方在完成一步走棋后，形成了一个闭合的格子，该走棋方同样要从所有未形成连线的某两个邻近点之间来继续连线。最终结束的标志是所有的邻近点都被连线。因此，点格棋的博弈树复杂度可估算为 $60!$ ，如果以 10 为底的形式表示，约为 10^{82} ，这是一个天荒地老的巨型数值，即使当今最先进的计算设备也不能在机器博弈规定的有限时间（比如 15 秒/步）内完成。

3.3 博弈问题的计算复杂性

国外有很多学者都在研究计算机博弈问题的计算复杂性，比如：

（1）国际象棋^[42]和西洋跳棋^[47]被证明属于 EXPTIME-complete 问题，这两个棋种的计算复杂性证明，在构建模型的过程中，在广义化的棋盘上模拟进行一种已被证明为 EXPTIME-complete 问题的 G_3 游戏^[48]，并最终证明了 G_3 游戏可多项式时间内归约到被广义化的国际跳棋（西洋跳棋）；

（2）围棋被证明属于 PSPACE-hard 问题^[49]（围棋也被怀疑属于 EXPTIME-complete 问题^[42]），五子棋^[50]、六子棋^[51]、奥赛罗棋^[52]被证明属于 PSPACE-complete 问题，这些棋种的计算复杂性证明，都用到了广义地理学游戏（Generalized Geography Game^[53]）；亚马逊被证明属于 PSPACE-complete 问题^[54]，在证明过程中，它采用了一种公式博弈（Formula Game^[53]）。

国际上被广大学者认可的机器博弈问题，在比赛规则上对参与比赛的任何一方都是十分公平的。因为只有是公平的，这样的博弈问题才具有长期存在的意义，但也正是这种公平性，大大增加了计算机博弈问题求解的难度。因此被广泛认可的博弈问题，其计算复杂性一般都属于某复杂性类的 hard（困难问题）或 complete（完全问题）（属于此类计算复杂性类的问题被认为是难解或是最难解的）。

对博弈问题的计算复杂性证明，前提条件是该博弈系统对应的棋盘是广义的，即无限大（一般为 $n \times n$ ）。如果是固定大小的棋盘（如围棋的 19×19 ），其复杂性是常量（无论在空间上还是在时间上），对问题的计算复杂性证明没有意义。

对于连珠类的博弈系统的计算复杂性证明已有一些成果。连珠类的博弈系统属于填子游戏，其一般的棋盘采用 $n \times n$ 类似围棋的棋盘，该博弈系统的游戏规则是走棋双方交替地向棋盘填子，直到形成同色的 k 子连珠，则该方获胜，如广为流行的四子棋（Connect-Four）、五子棋（Go-moku）和六子棋（Connect6）。这一类博弈系统一般用函数 $\text{Connect}(m,n,k,p,q)^{[51]}$ 表示，其中 m 、 n 表示横、纵交叉点的数量， k 表示走棋规则中获胜的条件，即形成同色连珠所包含的棋子数。 p 指第一手落几个棋子， q 规定第一手之后走棋方每轮落子的数量。

证明该类型博弈问题的计算复杂性属于 **PSPACE-complete** 问题，要根据 **PSPACE-complete** 的定义^[52]逐条进行证明。根据此定义的第二个条件，我们需要证明所有属于 **PSPACE** 的问题都归约到该判定问题（即该类型博弈问题），这基本是不可能做到的，需要换一个思路，如果找到一个被证明属于 **PSPACE-complete** 问题的解决方法，那么利用该判定问题的解决方法就可以解决所有的 **PSPACE** 问题，因此只需要找到一个已经被证明属于 **PSPACE-complete** 的判定问题，并证明它可归约到该类型博弈系统。证明问题属于 **PSPACE-complete** 的一般选择的是广义地理学游戏。

证明此类博弈问题属于 **PSPACE-complete** 的思路基本相同，证明过程如下：

- （1）证明该博弈问题属于 **PSPACE** 问题；
- （2）根据 **PSPACE-complete** 定义的第二个条件，需要找到一个已经被证明属于 **PSPACE-complete** 的判定问题；
- （3）在属于该博弈问题棋盘上构建一个归约模型，即特定局面；

(4) 在该模型中，模拟解决步骤 2 中选择的判定问题；

(5) 若在步骤 4 中，找到了一个解决方法（即必胜策略）。则说明步骤 2 中所选择的判定问题可归约到该博弈问题。即满足了 PSPACE-complete 定义的第二个条件；

(6) 根据步骤 5 和步骤 1，可知该博弈问题满足定义的两个条件。由此得证该类型博弈问题的计算复杂性属于 PSPACE-complete。若步骤 1 不成立，而步骤 5 成立，则该博弈问题只满足定义的第二个条件，因此根据相关定义可知该博弈问题的计算复杂性属于 PSPACE-hard。文献[51]和文献[52]就是采用了上述的证明思路。在问题的计算复杂性证明过程中，构建一个适当的归约模型至关重要，这种模型要模拟进行广义地理学游戏，也就是该模型是一个二维有向图（需要将此二维有向图嵌入到连珠棋棋盘上^[51]），由于广义地理学是一种两个参与者的游戏，因此必须迫使双方每轮走棋都只有一个走法可选择，若不选择该走法，则该走棋方会立即输棋。

在文献[51]中，在 $n \times n$ 的棋盘上，构建一个特殊模型（见图 3-7），其中包括：1 个仿真区域、1 个获胜区域、 $p-1$ 个辅助区域。其中仿真区域和辅助区域模拟进行广义地理学游戏，每轮双方分别向仿真区域落一个棋子、向辅助区域落 $p-1$ 个棋子，直到仿真区域已满，这样该走棋方将向获胜区域下一个棋子，该走棋方的迫着数将大于 1，对方必输（对方只有一个棋子可以落在获胜区域，只能解决一个迫着）。由此得到某走棋方的必胜策略。进而说明广义地理学在连珠棋盘上可以被求解。也就是说广义地理学多项式时间可归约到该博弈系统，从而说明其他的属于 PSPACE 的判定问题可归约到该博弈系统，因此满足 PSPACE-complete 定义的第二个条件，得证。

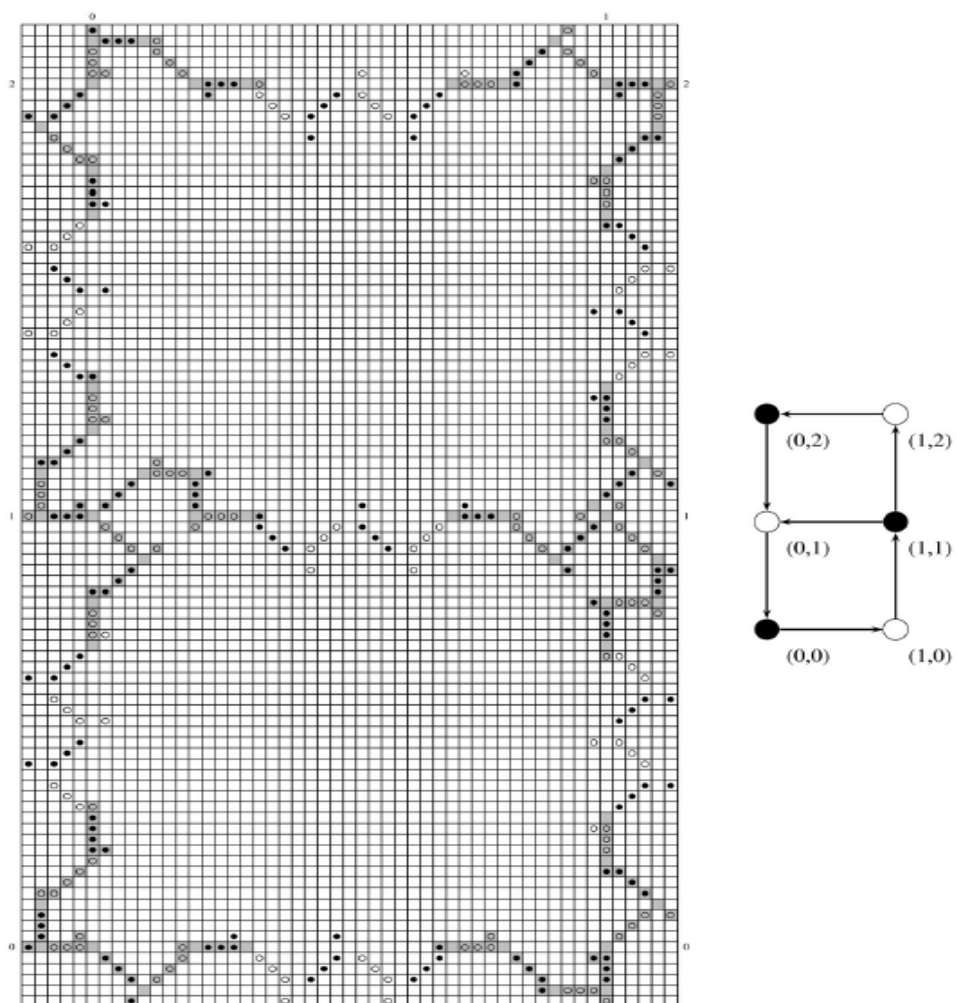


图 3-7 一个六子棋的特定模型

第 4 章 机器博弈的典型技术

4.1 概述

计算机博弈系统中,典型的关键技术主要包括搜索、评估与优化、学习与训练等技术,它们是决定博弈结果的关键因素。以中国象棋、国际跳棋、亚马逊棋等为代表的传统二人零和完备信息博弈,其搜索理论已经很成熟。典型的博弈搜索算法,从搜索方向考虑,可以分为深度优先搜索和宽度优先搜索;从控制策略考虑,可以分为盲目搜索和启发搜索;从搜索范围考虑,可以分为穷尽搜索、裁剪搜索。

研究表明,随着搜索深度加深,棋力增强,但信息处理量也大幅提升。宽度优先搜索、穷尽搜索和盲目搜索算法时间和空间开销巨大,难以做到很深的搜索。而且,在实际计算机博弈棋牌项目中,常常还有博弈时限限制和实时博弈要求,因此,基本上不可能直接使用此类算法去解决相关问题。

另外,不能单纯依靠加大搜索深度提高计算机博弈能力,还需要将必要的相关博弈知识引入到相应的博弈搜索中,实现剪枝。另外,对博弈局面评估得越准确,也就是先验知识越丰富、越正确,其获胜的几率就越高,这也就是计算机博弈活动中,有专业级或下棋水平较高的作者参与,胜率高的主要原因。当然,只有确保搜索算法与评估函数高度协调,博弈系统才能真正发挥有效的或高效的作用。

近几年来,随着计算机硬件和神经网络、机器学习、大数据等技术的发展,特别是 GPU 并行计算技术的广泛应用,使得深度学习变得更加便宜、快速、实用与有效,机器博弈系统的逻辑思维与计算能力也得到大幅提升。

4.2 穷尽搜索

极大极小算法是最基本典型的穷尽搜索方法,它奠定了计算机博弈的理论基础^[55]。通过极大极小算法可以找到对于博弈双方都是最优

的博弈值，但该算法对博弈树的搜索是一种变性搜索，算法实现相对麻烦。

负极大值算法是在极大极小算法基础上进行的改进算法，把极小节点值（返回给搜索引擎的局面估值）取绝对值，这样每次递归都选取最大值。

4.3 裁剪搜索

裁剪算法也称剪枝算法，是计算机博弈中最常用的主流算法，它包括深度优先的 **Alpha-Beta** 剪枝搜索^[56]和以此为基础改进与增强的算法，如渴望窗口搜索（**Aspiration search**）^[57]、**MTD(f)**（**Memory-enhanced Test Driver with f and n**）搜索^[58]等。在具体应用中，合理地交叉使用各种搜索方法，可以具有更高的效率。

1. Alpha-Beta 剪枝^[56,59]

Alpha-Beta 剪枝是在极大极小算法基础上的改进算法，是其它剪枝算法的基础。目前，多数博弈程序都采用负极大值形式的 **Alpha-Beta** 搜索算法。为保证 **Alpha-Beta** 搜索算法的效率，需要调整树的结构，即对搜索节点排序，确保尽早剪枝。

2. 渴望搜索^[57,60]

渴望搜索是在 **Alpha-Beta** 搜索算法基础上，缩小搜索范围的改进算法。渴望搜索从一开始就使用小的窗口，从而在搜索之初，就可以进行大量的剪枝。通常，渴望搜索与遍历深化技术结合使用，以提高搜索性能。

3. MTD(f)搜索^[58]

MTD(f)算法实际上就是不断应用零窗口的 **Alpha-Beta** 搜索，缩小上界和下界，并移动初始值使其接近最优着法。**MTD(f)**算法简单高效，在国际象棋、国际跳棋等博弈程序里，**MTD(f)** 算法平均表现出色。

此外，还有各种在 **Alpha-Beta** 搜索基础上优化的算法，例如，有

学者提出在博弈树同层结点中，用广度优先搜索，接力式空窗探测，平均搜索效率高于 MTD (f) 搜索^[61]。通常，裁剪算法需要与置换表技术相结合，以减少博弈树的规模，提高搜索效率。

4.4 启发式算法

“启发”（Heuristic）是指通过排序让 Alpha-Beta 剪枝的搜索树尽可能地接近最小树，优先搜索好的着法。启发通常有置换表启发、历史启发和杀手启发等常用的算法。

1. 置换表启发^[62,63]

置换表是一个大的直接访问表，用来存储已经搜索过结点（或者子树）的结果，下次搜索遇到时直接运用。置换表的构造，一般使用 Hash 表和 ZobristHash 技术来实现。

合理使用置换表，可以提高搜索效率，当博弈树的深度很大时，置换表对内存空间要求巨大。通常的对策是对置换表分配有限大小，并采用散列方式管理存取。具体应用到各个棋种中时，还要根据实际局面的节点类型进行处理。

置换表启发是置换表与 Alpha-Beta 剪枝算法相结合的产物。在中国象棋等棋种中，通过引进置换表启发技术来增强搜索效率。

2. 历史启发（History Heuristic）^[64]

历史启发也是迎合 Alpha-Beta 搜索对节点排列顺序敏感的特点来提高剪枝效率的。它维护着法历史，每当遇到好的着法，就给其历史得分一个相应的增量，使其具有更高的优先被搜索的权利。

3. 杀手启发（Killer Heuristic）^[65]

杀手启发可以看作是历史启发的特例。它把同层中引发剪枝最多的节点称为杀手，当下次搜索到同一层时，如果杀手移动是合法的话，就优先搜索杀手。杀手启发可以对着法进行动态重排序，提高了置换表的使用效率。

研究表明，历史启发与置换表技术结合可以大幅减少博弈树空

间，在残局阶段应用杀手启发算法可以明显节约时间^[64,65]。

4.5 迭代深化

迭代深化（**Iterative Deepening**）也称为遍历深化，是一种常用的蛮力搜索机制，经常使用在深度优先搜索中^[66]。迭代深化最初是作为控制时间的机制而提出的，通过对博弈树进行多次遍历，并逐渐提高搜索深度，一直到指定的时间停止。

迭代深化利用 **Alpha-Beta** 剪枝算法对子节点排序敏感的特点，使用上次迭代后得到的博弈值，作为当前迭代的搜索窗口估值，以此为启发式信息计算当前迭代的博弈值。另外，它利用时间控制遍历次数，只要时间一到，搜索立即停止。在关键的开局和残局，由于分支较少，可以进行较深层次的搜索。**Alpha-Beta** 剪枝经过一系列技术如置换表、历史启发、迭代深化等增强后，其性能可大幅提高。

4.6 最佳优先算法

最佳优先的搜索算法，不受节点排序的影响，其搜索空间小于深度优先的最小树，理论上应该优于深度优先。实际上，最佳优先算法仍处于理论研究阶段。最佳优先算法分为两类：采用极大极小算法取值的 **SSS***算法^[67,68]和 **DUAL***算法，不采用极大极小方法取值的 **B***^[69]和 **PB***^[70]算法。

1. SSS*和 DUAL*算法

SSS*和 **DUAL***算法都属于状态空间搜索（**State Space Search**），把极大极小树看成状态图，在不同的分支上展开多条路径，并且维护一个关于状态图的全局信息表。这两种算法是两个操作相反的过程，前者在搜索深度为偶数的极大极小搜索中表现较佳，后者则在深度为奇数搜索中较佳。

SSS*和 **DUAL***算法都过于复杂，难于理解，且时间和空间开销较大，在计算机博弈中实际应用较少。

2. B*和 PB*算法

B*算法用一个乐观值和一个悲观值来评价节点。当根节点的一个子节点的悲观值不比所有其它节点的乐观值差的时候，**B***算法就结束了。算法搜索控制的关键是尽快找到终止条件。由于它对局面估值的依赖性太强，估值的可信度将直接影响最终结果。

PB*算法就是基于概率的 **B***算法，这个算法对概率的准确估计比较敏感，实现困难。

4.7 随机搜索算法

随机搜索有两种算法：拉斯维加斯算法和蒙特卡罗算法。采样越多，前者越有机会找到最优解，后者则越接近最优解。通常，要根据问题的约束条件来确定随机算法，如果对采样没有限制，但必须给出最优解，则采用拉斯维加斯算法。反之，如果要求在有限采样内求解，但不要求是最优解，则采用蒙特卡罗算法。在计算机博弈中，每步着法的运算时间、堆栈空间都是有限的，且仅要求局部优解，适合采用蒙特卡罗算法。

1. 蒙特卡洛树搜索（MCTS, Monte Carlo Tree Search）^[71-74]

在人工智能的问题中，蒙特卡洛树搜索是一种最优决策方法，它结合了随机模拟的一般性和树搜索的准确性。由于海量搜索空间、评估棋局和落子行为的难度，围棋长期以来被视为人工智能领域最具挑战的经典游戏。近年来，**MCTS** 在类似计算机围棋等完备信息博弈、多人博弈以及其它随机类博弈难题上的成功应用而受到快速关注^[75]。理论上，**MCTS** 可以被用在以{状态，行动}定义并用模拟预测输出结果的任何领域。

基本的 **MCTS** 算法根据模拟的输出结果，按照节点构造博弈树，其过程如图 4-1 所示，包括路径选择（**Selection**）、节点扩展（**Expansion**）、模拟实验（**Simulation**）、反向传播（**Backpropagation**）四个步骤。

MCTS 算法适用于非完备信息博弈，也适用于有较大分支因子的

博弈程序，例如，AlphaGo 就是采用 MCTS 算法进行搜索^[10]。

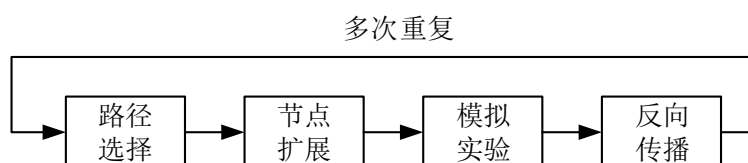


图 4-1 构造 MCTS 博弈树的过程

2. UCT 搜索算法^[76,77]

UCT 算法，即上限置信区间算法（UCB for Tree），是一种基于 MCTS 发展的博弈树搜索算法，该算法通过扩展 UCB（Upper Confidence Bound）到极大极小树搜索，将 MCTS 方法与 UCB 方法相结合而产生。

相对于传统的搜索算法，UCT 时间可控，具有更好的鲁棒性，可以非对称动态扩展博弈树，在超大规模博弈树的搜索过程中，表现出时间和空间方面的优势。目前，UCT 在搜索规模较大的完备信息博弈、复杂的多人博弈、非完备信息博弈以及随机类博弈项目中，表现出色^[78-80]。据不完全统计，国内机器博弈比赛成绩较好的队伍，绝大部分采用了 UCT 搜索算法。

4.8 并行计算

并行计算^[6]是为了提高计算速度，结合云计算、工作站集群、多核高性能计算机、并行机系统等技术^[81-84]，充分发挥计算机强大的并行处理能力。在机器博弈中，运用并行技术把博弈树动态分开，同时执行多个指令，可以在不裁剪和缩小博弈树的规模情况下，提高搜索、训练、分析的速度，优化系统性能。

并行计算主要有两种体系，单机体系 SMP（Symmetric Multiprocessor，对称多处理器）和分布式体系 Cluster（计算机集群），对应多线程并行和多机并行。两者最大的区别是，前者可以共享存储器（并且共享同一地址的存储单元），后者则必须通过网络来交换数

据。

近年来，网络并行计算及多 CPU、多核技术的研究日趋成熟，尤其是 CPU/GPU 异构混合并行系统，以其强劲计算能力、高性价比和低能耗等特点，成为新型高性能计算平台^[85-89]。计算机博弈中，由于博弈搜索通常需要用到置换表，故适合采用基于 SMP 方式的多线程并行计算。随着大数据、云计算等技术的成熟与完善，CPU/GPU 集群等并行技术被越来越多地运用到计算机博弈，特别是用于深度神经网络模型训练中，达到快速训练深层模型的目的。

4.9 遗传算法

遗传算法是人工智能领域的关键技术，它是一种非数值、并行、随机优化、搜索启发式的算法，通过模拟自然进化过程随机化搜索最优解。它采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向、不需要确定的规则，同时具有内在的隐并行性和更好的全局寻优能力^[90]。

遗传算法是解决搜索问题的一种通用算法，在计算机博弈中，遗传算法通常被用于搜索、自适应调整和优化局面评估参数。它的基本思想是将博弈树看作遗传操作的种群，博弈树中由根节点到叶子节点组成的所有子树为种群中的个体。根据优化目标设计评估函数，计算种群中每个个体的适应度函数值，依据适应度函数值的大小确定初始种群，让适应性强（适应度函数值大）的个体获得较多的交叉、遗传机会，生成新的子代个体，通过反复迭代，可得到满意解。

采用遗传算法优化局面估值时，可根据博弈程序与其他程序对弈的结果，检验某一组参数获胜的几率。经过多次试验，通常可以找到较好的估值参数。传统的算法一般只能维护一组最优解，遗传算法可以同时维护多组最优解。在实践中，遗传算法被引入了中国象棋、国际象棋、亚马逊棋以及禅宗花园游戏等博弈系统的智能搜索与评估优化中，效果还是很明显的^[20,90-95]。

4.10 神经网络

人工神经网络（Artificial Neural Network，即 ANN），简称为神经网络或类神经网络。它是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的输出——激励函数（Activation function）。每两个节点间的连接都代表一个对于通过该连接信号的加权值，这相当于人工神经网络的记忆。网络的输出则依网络的连接方式，依权值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近，也可能是对一种逻辑策略的表达。

人工神经网络研究以多伦多大学的 Geoffrey Hinton^[8,9,96-101]为代表，目前已有前馈神经感知网络（Feed forward neural networks, FF or FFNN）、径向神经网络（Radial basis function, RBF）、霍普菲尔网络（Hopfield networks, HN）、深度信念网络（Deep belief networks, DBN）、卷积神经网络（Convolutional neural networks, CNN）、深层玻尔兹曼机（Deep Boltzmann Machine, DBM）、堆叠自动编码器（Stacked Auto-Encoder, SAE）等数十种模型。

近年来，人工神经网络的研究取得了很大的进展，尤其是实现了以超算为目标的并行算法的运行与概念证明后，在机器博弈、计算机视觉、模式识别等人工智能领域与深度学习相结合^[102]，成功地解决了许多现代计算机难以解决的实际问题（例如围棋、中国象棋博弈中的估值、学习与训练等），表现出了良好的智能特性^[10,103,104]。

4.11 机器学习

机器学习（Machine Learning）的根本任务是数据的智能分析与建模，正成为发展新学科的基础^[105]。它利用经验来改善计算机系统自身性能^[106]，让计算机系统具有人类的学习能力，以便实现人工智能^[107]。

与传统为解决特定任务、硬编码的软件程序不同，机器学习是用大量数据进行训练，使用各种算法来解析数据并从中学习，做出决策和预测。当前主流机器学习技术包括度量学习、多核学习、多视图学习、集成学习、主动学习、强化学习、迁移学习、统计关系学习、演化学习、并行机器学习、哈希学习等，其中强化学习（Reinforcement Learning，也称为增强学习）被列为机器学习的四大研究方向之一^[108]。

强化学习研究学习器在与环境的交互过程中，如何学习到一种行为策略，以得到累积利益最大化^[109]。在机器博弈中，强化学习的设定可用图 4-2 来表示，学习器所处的环境为博弈规则，学习器根据当前博弈状态输出着法，以博弈收益作为每步着法的结果，反馈给学习器，以期望最终的利益最大化。

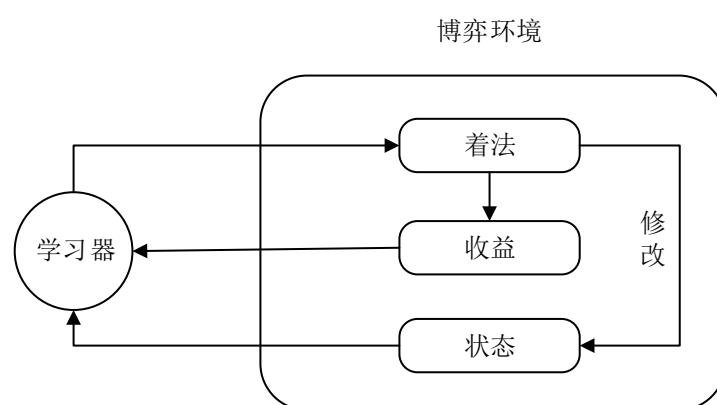


图 4-2 机器博弈强化学习设定

在实际应用中，由于强化学习的优化目标通常涉及多步决策，相对复杂，且策略的搜索空间巨大，优化比较困难。另外，强化学习还面临着特征表示、泛化能力等诸多挑战。

深度学习是基于多层网络结构的特征学习方法，把原始数据通过多层神经网络非线性变换，逐层提取抽象特征，完成复杂的目标函数系统逼近。深度学习典型的网络模型包括卷积神经网络、深层玻尔兹曼机和堆叠自动编码器等。利用 GPU 来训练深度神经网络，充分发

挥其并行计算能力，大幅缩短海量数据训练所耗费的时间，因此 GPU 并行计算已经成为业界在深度学习模型训练方面的首选解决方案。

相对于传统的机器学习方法，深度学习能够学习多层次抽象的数据表示，能够发现大数据中的复杂结构，对于解决强化学习中策略评估和优化的问题有明显优势。深度学习被成功地用于机器博弈中^[110]，例如采用基于深度学习和 Q-Learning 的 Deep Q-Network^[111]技术的博弈系统已达到人类玩家水平，而 AlphaGo^[10,11]则可以战胜人类顶级高手。

尽管深度学习技术在围棋机器博弈方面取得了前所未有的成功，但在拓展应用方面仍面临一些问题，例如深度学习训练耗时、非凸函数模型在理论研究中存在困难等。如何合理利用深度学习方法增强传统学习算法的性能，提升计算机博弈水平，仍是今后研究的重点。

第 5 章 机器博弈的平台技术

5.1 计算机博弈平台概述

计算机博弈平台系统本身并不具有下棋或出牌的逻辑决策功能，但是它可以加载其它一个或多个决策引擎程序，使这些引擎程序以选手的角色参与对局^[36]。在对局中主要起到规则判定和输入输出交互界面作用。它为对局的参与者提供了更高的执行效率、更方便的操作方法和必要的规则评判，使计算机博弈对局更加公平、公正和高效。

通常，一个完整的计算机博弈系统至少需要包含输入模块、逻辑计算模块、输出模块等几大部分。研究与开发计算机博弈平台的意义在于：

(1) 设计简明清晰的交互协议、搭建美观好用的计算机博弈平台，将更好体现牌局或棋局状态，有助于初学者快速进入计算机博弈本质性研究，促进计算机博弈的普及和推广。

(2) 在研究和开发计算机博弈引擎的过程中，需要通过大量的测试验证引擎的逻辑性能。好的计算机博弈平台不仅可以实现快速自动对局过程，提高对局效率，还具有棋谱记录和分析的功能，方便沟通交流，为调校和改善计算机博弈引擎程序提供了高效的解决方案。

(3) 计算机博弈比赛中人工操作难免出现意外或拖延，从而干扰了检测的准确度。利用计算机博弈平台可以自动实现对局操作和规则判断，避免因人工操作引起的不必要的分歧。

5.2 计算机博弈平台的分类

根据计算机博弈项目不同划分标准，计算机博弈平台可以分为如下几类：

1. 完备信息博弈平台和非完备信息博弈平台

根据计算机博弈项目中参与者信息的差异程度，计算机博弈平台也分为完备信息博弈平台和非完备信息博弈平台。完备信息计算机博

弈只需要博弈平台实现良好的显示效果、便捷的输入操作，并不需要承担信息生成、存储和发布。而不完备信息项目如果不使用博弈平台则需要人工裁判或辅助棋盘才能完成对局，对局的效率和公平性将会受到不同程度的影响。

2. 单引擎博弈平台和多引擎博弈平台

只能加载一个博弈引擎程序的平台称为单引擎博弈平台，通常用于人机对局形式。能加载两个或更多博弈引擎程序的平台称为多引擎博弈平台，可以快速实现比赛对局。

3. 单机博弈平台和网络博弈平台

单机博弈平台不具备网络通讯功能，需要对局者在距离较近的场地空间开展对局。网络博弈平台通过网络通信协议使距离较远的选手也能实现对局。

4. 程序级博弈平台和模块级博弈平台。

程序级博弈平台已经在多个项目中被广泛使用，博弈平台通过加载完整的可执行程序完成对局，程序相对于平台是封闭、不透明的，因此平台无法避免引擎程序实施博弈逻辑决策以外的计算和操作，导致比赛中具有一定的潜在危险。模块级博弈平台尚处于探索研究阶段，博弈平台通过对选手的模块级源程序代码实现检测和编译运行，可以进一步避免非法操作或实现针对某一特殊博弈项目情景进行专项测评^[112]。

5.3 计算机博弈平台的设计规范

为了方便交流和竞赛，计算机博弈平台应在人机交互、通信协议等方面遵循一定的规范。

5.3.1 人机交互接口规范

1. 显示输出规范

计算机博弈平台应考虑显示界面是否符合操作者日常操作习惯，充分利用显示区域醒目地突出棋局或牌局的重要信息，也应酌情考虑

颜色、对比度、分辨率、或者图案心理因素等对操作者的影响。

2. 操作输入规范

计算机博弈平台的操作也应尽量符合大众的操作习惯，比如采用鼠标的定位和点击左键实现棋子的拾取、落下。

3. 其它约定规范

计算机博弈平台应对引擎决策耗时进行记录 and 统计，并及时显示用时情况。还应对引擎超时或异常、不当操作引起的错误做出相应判罚。

5.3.2 平台与引擎通信的协议规范

1. 对局状态和过程的信息编码

棋类项目的计算机博弈平台协议应对棋盘位置、双方棋子、参与者的每一个决策行动进行编码约定，牌类项目的计算机博弈平台协议应对纸牌的花色和点数进行编码约定，编码规则建议参考大型国际或国内比赛相关规定及现有协议约定，或按有利于开发和维护的方案进行设计。

2. 引擎与博弈平台的通信协议

博弈平台通过与引擎的通信发布棋局状态信息、获取引擎决策行棋反馈信息。这些通信协议通常包括对局参与者身份信息、先后手角色信息、棋局或牌局初始化信息、对局过程信息、胜负判定信息和异常判定信息。

3. 引擎示例代码

博弈平台的开发者应提供简单容易理解的引擎示例源代码，便于初期参与者快速理解规则和协议。示例代码应能完整合法地实现对局过程，但不需要具有太高的行棋决策能力，以便给博弈平台的用户留下更广阔开放的发挥空间。

5.4 计算机博弈平台的相关技术

1. 匿名管道通信技术

现有的计算机博弈平台与引擎之间的通信普遍采用匿名管道技术。管道是一种进程间的通信机制。博弈平台启动引擎进程，并创建管道与引擎之间进行通信。

2. 数据保存和加密技术

为了记录对局过程和结果，计算机博弈平台通常将对局过程和结果以文件的形式记录下来，供赛后重现局面和分析测试。一些非完备信息博弈项目在生成初始数据和保存行棋记录时应采用加密方法。目前计算机博弈平台普遍使用的是对称式加密技术，即加密和解密使用同一个密钥^[113]。

3. 在线对局技术

为了实现远程异地的计算机博弈交流和竞赛，计算机博弈平台可以通过 TCP 或 UDP 协议与远程的引擎建立数据通信。采用这种形式的竞赛应考虑其它辅助检查手段，避免非计算机因素（如人工操作）干扰对局结果。

5.5 计算机博弈平台应用实例

国内许多高校、企业和研究机构都在努力研发各种计算机博弈平台系统。其中，在中国大学生计算机大赛及其官方网站上推荐以下 4 项博弈平台系统。

1. 哈尔滨理工大学军棋博弈平台系统

哈尔滨理工大学军棋博弈平台系统是 2012 年最早被指定为全国大学生计算机博弈大赛暨全国锦标赛的比赛专用平台。它解决了军棋作为非完备信息博弈需要裁判进行规则判定的问题^[114]。

2. 哈尔滨理工大学二打一（斗地主）博弈平台系统

哈尔滨理工大学二打一（斗地主）博弈平台系统是 2014 年被指定为全国大学生计算机博弈大赛暨全国锦标赛的比赛专用平台。它解决了斗地主作为同时存在对抗和合作性质的多方不完备随机博弈的数据初始化和规则判定的问题^[115]。

3. 北京理工大学苏拉卡尔塔棋博弈平台系统

北京理工大学苏拉卡尔塔棋博弈平台解决了苏拉卡尔塔的引擎图形化界面和对局操作的问题。

4. 沈阳航空航天大学通用计算机博弈对战平台

沈阳航空航天大学研发的通用计算机博弈对战平台实现了六子棋、亚马逊棋、苏拉卡尔塔棋、幻影围棋、不围棋、点格棋等项目的引擎图形化和对局操作问题。这种在一个博弈平台上适应多个博弈项目的探索为发掘博弈项目之间的联系起到了重要的作用^[116-118]。

一些研究机构和企业也发布了他们的计算机博弈平台，比如象棋百科全书网站提供了用于交流的中国象棋的博弈平台和源代码。联众开发了桥牌网络在线博弈平台，分别应用于 2015 年全国智力运动会和 2017 年中国计算机博弈大赛的桥牌比赛中。2014 年起步的新睿桥牌社区采用桥牌机器人 AI 陪人竞赛的模式开拓并验证了机器博弈的现实应用模式，目前已有 17 万用户，每天有近 3 万人在线竞技，是国内最大最活跃的桥牌平台。

第 6 章 完备信息机器博弈的专项技术

6.1 六子棋机器博弈

6.1.1 概述

在五子棋的基础上，台湾交通大学教授于 2005 年提出了六子棋^[46,119]，并泛化出一系列 k 子棋。五子棋 $connect(n=15, k=5, p=1, q=1)$ 的两个主流版本 Renju^[120]和 Go-Moku^[121]分别于 1995 年和 2001 年被弱解决（Weakly Solved），两种规则下皆为“先手（黑）方必胜”。六子棋比五子棋复杂得多，形式化为 $connect(n=19, k=6, p=2, q=1)$ 。

六子棋无禁手，一般采用 19×19 的棋盘。 k 子棋博弈是动态的、二人的、完备信息的、非合作的博弈问题。设博弈双方分别为 $side_1$ 和 $side_2$ ， $p_0 \sim p_n$ 是局面序列，初始局面为 p_0 ，六子棋的棋局演化过程如图 6-1 所示。

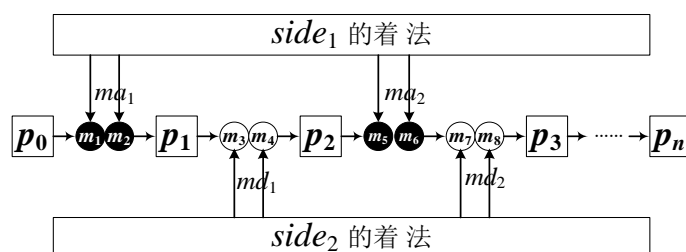


图 6-1 棋局演化过程

六子棋有如下显著特点：1) 平均分枝因子大。普通的博弈树搜索的深度太浅，在一定程度上抑制了搜索的作用。2) 开局、中局、残局的策略差异不显著。3) 一次走两颗子的规则，导致六子棋的状态空间、博弈树空间复杂度与围棋相近。4) 存在广泛适用的判定胜负的特定搜索策略——迫着搜索。

6.1.2 六子棋机器博弈主要技术

6.1.2.1 知识表示

知识表示影响问题的求解难度。基于六子棋规则，文献[14]提出了“棋盘 三进制线 二进制模式 点”的分层表示方法，实现了领域知识的有效表示、复用，提供了引入知识解决六子棋计算机博弈问题的一个接口。三进制（黑子、白子、空点）的线可等价地分解为多个二进制（有子、无子）的模式，如图 6-2 所示。二进制的模式可以简单穷举，并对其进行细致分析，从而形成模式知识库。

文献[45]首次定义了较为完备的模式类型；文献[14]进一步完善了模式的定义，提出了基于演化关系的既定性又定量的知识表示体系，约简并抽取出了知识表达的主要维度，给出了迭代生成全部模式的具体方法，提供了实现知识库的完整方法。

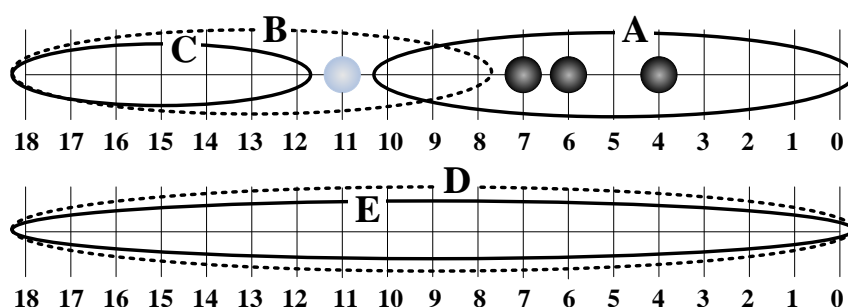


图 6-2 三进制线拆分成二进制模式的示例

全部棋形共计 1,048,512 个，在文献[14]中被划分为 15 个等价类，命名为 15 种类型：胜、必胜、活五、死五、活四、眠四、死四、活三、眠三、死三、活二、眠二、死二、活一、其它。部分类型的棋形举例如图 6-3 所示。

表 6-1 为六子棋知识库中全部二进制模式的类型、数目、比例的简单统计结果。常见模式，如其它、活一、死二、眠二、活二、死三、眠三、活三等所占的比例较少，总共约占 15%。同样类型的棋形，其价值相差无几。但是，在实际对弈中的统计数据表明，所包含的棋子

数越多，则冗余度也越大，出现的概率也就越低。所以，虽然棋形的可能组合数目较大，但真正会出现的，只是其中很少一部分。

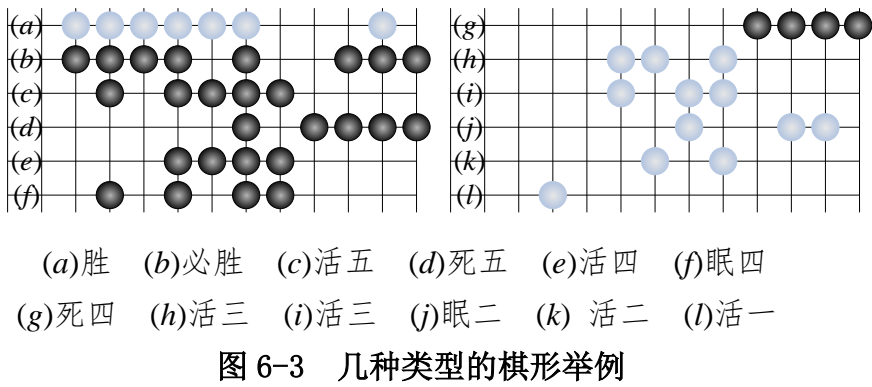


表 6-1 connect(19, 6, 2, 1)中，不同类型模式的数目和比例

棋形类型	#	%	棋形类型	#	%
获胜	112896	10.767	活三	86913	8.289
必胜	192916	18.399	眠三	55085	5.254
活五	181781	17.337	死三	2186	0.208
死五	45952	4.383	活二	14425	1.376
活四	149319	14.241	眠二+死二+活一+其它	5239	0.450
眠四	191025	18.219	总计	1048512	100.00
死四	10775	1.028			

6.1.2.2 搜索和推理

除了常见的基于 Alpha-Beta 的搜索策略，以及利用基于探索与利用均衡的抽样方法来弱化对专家估值需求的 UCT 策略之外， k 子棋研究者在文献[121]提出了两种有效的搜索方法：证据计数搜索 PNS（Proof Number Search），迫着空间搜索 TSS（Threat Space Search）。这两种算法成为最终解决 Renju 和 Go-muku 的主要技术。

PNS 是一种最佳优先搜索策略，尝试以尽可能低的状态空间复杂度给出关于赢/不赢这类二元问题的肯定或否定的解答。TSS 是一种基于回答特定问题而根据规则进行剪枝的高效搜索算法，这种剪枝是无风险的。在六子棋中，由于一次可以走两颗子，迫着搜索情形更多，也更为复杂。采用 TSS 搜索策略已成为所有六子棋程序的必备选项。

之一。

在分层表示的情况下，增量更新^[111]是一种非常有效的状态演化方法，在实践中常被采纳。

6.1.2.3 机器学习

机器学习方法在博弈问题中越来越重要。文献[111]介绍了击败李世石的 AlphaGO 方法, AlphaGO 主要采用深度学习、强化学习和 UCT 技术。这为六子棋的相关研究提供了良好的思路。

六子棋的机器学习相比于围棋有更多的优势：第一，基于分层描述的六子棋知识表示，在策略（policy）表达上比围棋更容易。第二，TSS 有助于构建大规模有监督的训练集。第三，六子棋基础知识库较小，可以围绕该知识库，通过学习，扩展和构建实用的高级知识的知识库。

总之，由于难度和围棋有可比性，加上近年来以深度学习、强化学习等为代表的新技术突破，构建水平更高的六子棋程序越来越容易。但是，实时获得六子棋博弈问题的解依然困难重重，这需要探索更多的方法。

6.2 围棋机器博弈

6.2.1 概述

围棋之所以被视为人类在棋类里面最后的堡垒，是有其内在原因的，围棋的空间复杂度极大。而且局面非常难于评价。根据 Allis 对几种双人、零和、完备信息的棋类游戏的复杂度估计^[122]，显然，19 路围棋的状态空间复杂度和博弈树复杂度都远远高于其它棋类。

针对高复杂度完备信息博弈问题，其研究主要集中在围棋上（博弈树复杂度 10^{360} ）。由于其极大极小树的分支因子过大，Alpha-Beta 搜索及其优化方法无法搜索足够的深度，导致其失去了效力。在很长一段时间内，静态方法成为了研究的主流方向^[123,124]，其顶峰为“手谈”和 GNUGO 两个程序，在 9*9 的围棋中达到了人类 5 至 7 级水平。

这种趋势在 2006 年被 S. Gelly 等人提出的 UCT 算法^[125]彻底的改变。该算法在蒙特卡洛树中使用 UCB^[126]解决了探索和利用的平衡，并采用随机模拟对围棋局面进行评价，极大地提升了计算机围棋的水平。其在 9 路围棋中已经可以偶尔击败人类职业棋手，但在 19 路围棋中还远远无法与人类棋手抗衡^[127]。此后的十年中，围棋的研究基本限于 UCT 的搜索框架而展开，围棋领域知识难以有效提炼，进展并不令人满意^[128,129]，直至 D. Silver 等利用深度学习对围棋领域知识进行学习^[130]。该方法对专家棋谱进行监督学习和自博弈强化学习，使用策略网络和估值网络实现招法选择和局势评价，通过与蒙特卡洛树搜索算法的结合，极大地改善了搜索决策的质量；同时提出了一种异步分布式并行算法，使其可运行于 CPU/GPU 集群上。在此基础上开发的 AlphaGo 于 2016 年击败了韩国九段棋手李世石；其升级版本“Master”于 2017 年 60 连胜人类顶级高手；2017 年，AlphaGo 的新版本以 3:0 的比分完胜围棋世界排名第一的柯杰，引起了巨大的轰动。这些人机大战是人工智能的划时代事件，并将极大推动人工智能的大发展。

6.2.2 围棋机器博弈主要方法

6.2.2.1 UCT 方法

2006 年，Kocsis 和 Szepesvari 提出了基于蒙特卡罗的 UCT^[125]算法，UCT 的全名为 UCB for Tree。UCB(Upper Confidence Bound)^[126]是用来解决老虎机吃角子问题而提出的，属于统计学领域的方法。

$$Gen_i = X_i + \sqrt{\frac{2 \ln N}{T_i}} \quad (1)$$

公式 (1) 为 UCB 的计算公式，其中 Gen_i 表示第 i 台机器新的收益， X_i 表示第 i 台机器目前为止的平均收益， T_i 表示第 i 台机器玩过的次数， N 表示全部机器玩过的次数。UCT 其实就是把 UCB 的公式用于围棋全局搜索中，是一种最佳优先的算法。它把每个叶子节点都

当作一个老虎机吃角子问题，收益由执行随机对弈的模拟棋局得到，胜负结果将更新树中所有节点的收益值。**UCT** 算法不断展开博弈树并重复这个过程，直到达到限定的模拟对局次数或耗尽指定时间。收益最高的子根节点成为 **UCT** 算法的最终选择。

UCT 算法是将蒙特卡洛方法和 **UCB** 的思想结合到树搜索的算法中，利用每个节点在蒙特卡洛模拟结果中的收益作为博弈树节点展开的依据，对树进行展开。蒙特卡洛树搜索算法的过程如图 6-4 所示。蒙特卡洛树搜索算法包含四个过程：选择、拓展、模拟和反馈。在选择过程中，搜索算法首先从树的根节点开始根据一定的策略选择一个到达叶节点的路径，并对到达的叶节点进行展开（拓展过程），之后对这个叶节点做蒙特卡洛模拟对局并记录结果（模拟过程），最后将模拟对局的结果按照路径向上更新节点的值（反馈）。蒙特卡洛树搜索算法迭代进行这四个过程，直到达到终止条件，例如到了规定的最大时间限制、或者树的叶节点数和深度达到了预先设定的值。

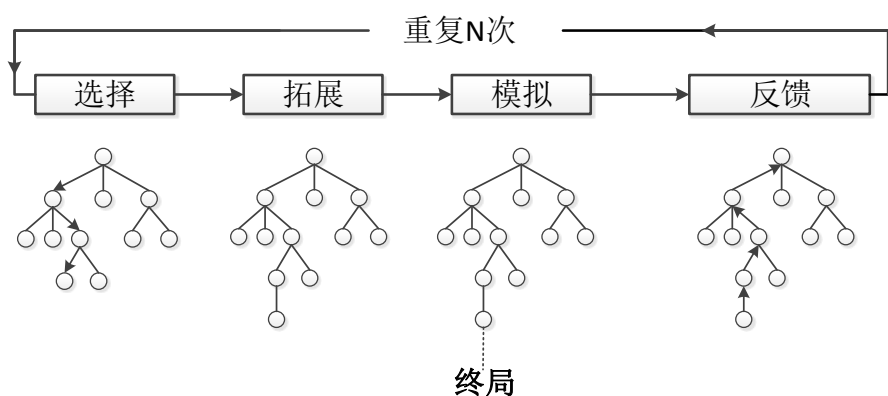


图 6-4 蒙特卡洛树搜索过程

简而言之，**UCT** 搜索过程使用 **UCB** 作为博弈树展开的依据，利用蒙特卡洛过程进行叶子节点的评价，评价价值回溯并更新展开的子树，作为节点的收益，即公式（1）中的 X 。

近几年，以 **UCT** 算法为基础的围棋机器博弈仍然处于一个高速

发展的过程中，文献[131]提出了模拟对局的 RAVE 增强算法，文献[132,133]将机器学习加入到全局搜索中，Chaslot 和 Gelly 的研究成功地将 UCT 并行化^[134,135]，王骄等提出了利用 4*4 的 Pattern 库提高模拟棋局的质量^[136]，使用 OOV 算法进行特征学习^[137]。谷歌的 AlphaGo 围棋程序在 UCT 搜索中加入了使用深度学习结合强化学习方法创建的策略网络和估值网络，并使用庞大的 CPU 集群和 GPU 集群进行计算支持。它以 4 比 1 完胜人类九段棋手李世石，在世界范围内引起了巨大的轰动。

6.2.2.2 深度学习与 UCT 结合

AlphaGo 中，利用深度学习的方法训练了两个网络，即 Policy Network（策略网络）和 Value Network（估值网络），如图 6-5 所示。两个网络的训练过程都包括两步，即监督学习（学习专家棋谱）和增强学习（自博弈）。

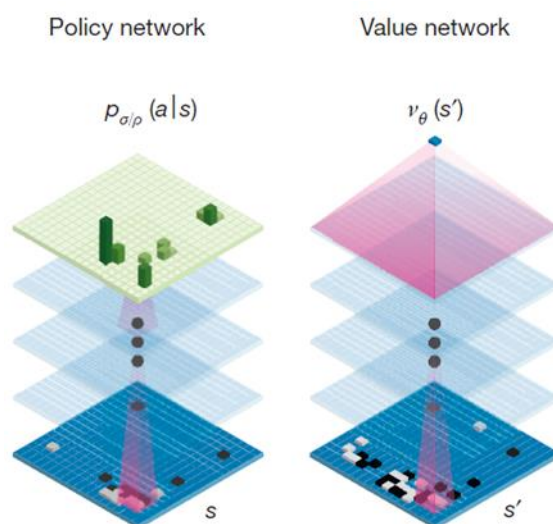


图 6-5 策略网络与估值网络

策略网络输入一个 state（局面），给出一个招法 a （实际上给出的是所有走子点的概率排列，需要保证随机性，并不是一直选最大概率的招法）。估值网络输入一个 state，给出评价 v 。AlphaGo 里面用

了两个策略网络（一个复杂，另一个简单但是执行速度快），一个估值网络。**AlphaGo** 的搜索框架，稍微对标准的 **UCB** 选择有所改进（公式 1），加入了一项先验概率（其实这个想法不是新的），由复杂的策略网络给出（每个展开的节点只需要执行一次）。简单的策略网络则用于蒙特卡洛模拟过程（可以叫做 **rollout** 或 **simulation**），而且并不是完全由策略网络进行模拟对局，是作为有效知识的补充（例如 **Atari**、**Extension**、**Capture** 这些明显的走子）。叶子节点进行蒙特卡洛模拟过程的同时，也用估值网络进行评价（也只做一次，重复出现不用再做），模拟过程的结果与估值网络给出的评价值加权求和，作为此节点最后的估值。搜索树更新的过程与传统 **UCT** 也是类似的。

更多细节详见 **DeepMind** 发表在 **Nature** 的论文^[10]，论文对学习过程的细节做了更为详细的描述。需要解释的是，学习专家棋谱过程中，即输入一个局面信息，输出一个招法，但实际输入的并不只是 19×19 的棋盘信息（0 空点，1 黑子，2 白子），还包括了特征信息（**Features**），见 **Extended Data Table 2** 和 4。也就是说，按照不同角度衡量每个点周围的情况，一起做编码。在近期东北大学所做实验中，采用 22 位的编码，网络的输入即是 19×19 个 22 位数。当然，编码过程也要时间（除非是增量更新，但是会很麻烦），所以快速策略网络不能加太多的特征。

6.3 点格棋计算机博弈

6.3.1 概述

点格棋又称之为点点连格棋、围地棋等等，是国外的一种添子类游戏。点格棋已经被纳入国际计算机奥林匹克大赛多年，2010 年正式成为中国计算机博弈比赛棋种。

点格棋虽然规则简单，但是其状态空间巨大，**Barker** 和 **Korf** 使用 **Alpha-Beta** 搜索首次完全解决了 4×5 棋盘尺寸的点格棋问题，并得出结论，这一尺寸下，棋局一定可以以平局结束^[138]，这也是目前

被完全解决的最大尺寸点格棋问题。目前，中国计算机博弈比赛点格棋采用 6×6 棋盘规格。

6.3.2 点格棋机器博弈主要技术

6.3.2.1 点格棋棋盘表示

棋盘表示是博弈的基础，好的棋盘表示可以获得更高的执行效率。目前点格棋常用的棋盘表示有矩阵表示、十字链表表示等方法。相对而言，采用十字链表表示可以与点格棋棋盘较好的匹配，同时还可以获得较高的效率。除此之外，棋盘表示中还会增加一些特殊字段来优化这种表示，如 `hash` 值、链接度等等。

1. 矩阵表示法

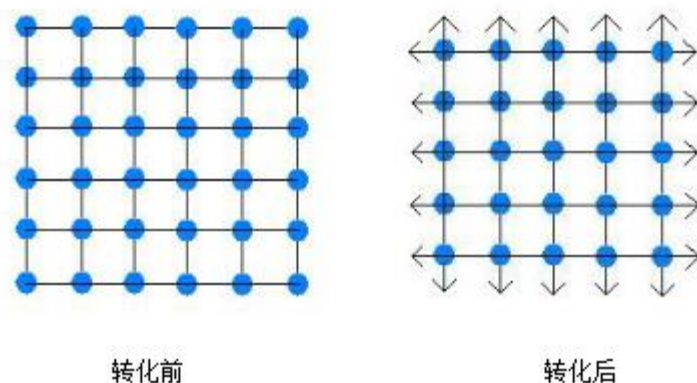
目前一般表示点格棋棋盘的方法是将棋盘表示为一个 6×6 的二维点阵数组，一个 2×2 的“子点阵”叫做一个格。两个点 (i, j) 和 (k, l) 当且仅当 $|i-k|+|j-l|=1$ 时叫做邻近的。邻近的两点连成一条边，每个格子由这样的四条边围住时，格子被俘获。文献[139]按此方法实现了棋盘表示和棋局局面的判断，这种表示方法重点保存的是点，考虑点之间的连接。文献[140]提出了一种新的棋盘表示方法，该方法重点保存的是格，考虑格之间的连接。

2. 十字链表表示法

为方便点格棋棋局状态分析，文献[141，142]采用了对点格棋的棋盘做如图 6-6 所示的等效变换。原棋盘中的竖边对应于变换后的横边，原棋盘中的横边对应于变换后的竖边，原棋盘中的每格各自转化为一个点，图中方点称为地。因此游戏转化为每步选择删去一边，当某点所连的四条边全部被删去后，此点由删去最后一边的一方获得，当游戏结束时，得到点数多的一方获胜。

由于棋局有 30 条横边与 30 条竖边构成，每条边有存在和被删去两种状态，因此可以用 2 个 32 位整型数 `H, V` 表示，0 表示该边未被删去，1 表示该边已经删去。此外，可以通过 `(H, V, S0, S1)` 唯一地表

示一个棋局局面，其中 H, V 为边的状态， $S0$ 为当前走棋一方的得分， $S1$ 为另一方的得分。



6-6 转化前与转化后的棋局

6.3.2.2 估值函数

目前，大多数 AI 程序使用的是静态估值，即按照已知的策略和技巧对棋局评估。这些方法在很大程度上依赖于开发者对游戏规则的理解和经验知识，通常要求开发者掌握较高的水平，而评估质量难以保证，并且这些规则的确定需要一个漫长的总结积累过程。

文献[142]利用人工神经网络 ANN(Artificial Neural Network)进行估值，设计 ANN 模型的关键之处在于选择合适的局面特征使其可以反映出局势情况的内在规律。在点格棋局面特征的选取上主要有两种方案，一种是使用原始的局面，将其用二进制压缩表示的形式作为人工神经网络的输入；另一种是统计局面中链、环等信息，将原始局面信息抽象为易于人类分析的形式。前者的优势在于没有信息丢失，每个输入唯一对应于一种局面状态，但问题是输入信息过大，网络规模大，运算速度慢，且内在规律不明显，训练难度大；后者存在信息丢失，但是模型节点数较少，计算速度快，规律明显，训练难度低。

6.3.2.3 搜索技术

搜索是着法选择的过程，也是程序中最耗时最复杂的部分。点格

棋的分支因子比较大，因此不可能对所有局面进行搜索，选择一种高效的搜索算法尤为重要。在过去的几十年中极大极小搜索不断得到改进，Alpha-Beta 剪枝，迭代加深，置换表，启发式算法等的综合利用可以使搜索效率提高几个数量级。为了避免基于极大极小值搜索的游戏状态树搜索过程中，对游戏状态评估经验的依赖，蒙特卡洛树搜索（MCTS）算法^[143]应运而生。它通过大量随机对局模拟来解决博弈问题，具有很好的通用性和可控性。在 DeepMind 团队将卷积神经网络 CNN（Convolutional Neural Network）技术引入计算机博弈^[10]之后，集成深度学习^[144]方法在计算机博弈领域得到了广泛关注。

1. UCT 与 ANN 相结合的方式

文献[142]中采用 UCT 与 ANN 相结合的方式。ANN 具有近似估计局面优劣的特性，将 ANN 用在对叶子节点的评估上，可以不必将游戏进行到结束即可近似计算出可能的双方获胜概率，以减少单次模拟用时。由于 ANN 是近似估计，错误不可避免，这就需要通过大量模拟以消除少量错误估计带来的影响，UCT 算法正是通过大量模拟，通过在线学习的方式来判断走法好坏的算法。UCT 与 ANN 的结合使用，一定程度上减少了模拟时间，而准确性上不会有太大损失。

2. CNN 集成的 Alpha-Beta 搜索

毫无疑问，一次 Alpha-Beta 完全搜索可以提供最精确的游戏局面评估，但是在游戏早期阶段，一次完全搜索将耗费太多时间。通常，在非完全的 Alpha-Beta 搜索中，需要人工定义基于知识工程的复杂局面评估函数，开发难度高，时间开销大。一个经过充分训练的 CNN 模型可以立即给出对一个游戏局面的评估，但是 CNN 的评估精度尚不能与一次完全 Alpha-Beta 搜索的结果相比。将 CNN 与其他算法集成，通常能以少量时间效率为代价提高算法的整体评估精度。

一个集成深度学习的方案是当被搜索局面的回合数处在卷积神经网络的置信回合区间中时，卷积神经网络模型将直接充当

Alpha-Beta 搜索的局面评估函数，为博弈搜索树的叶节点提供局面评估。

另外，强人工智能 AlphaGo 与 DeepStack 都使用了集成深度神经网络的 MCTS 方法。事实上，在点格棋中 CNN 也可以与 MCTS 搜索算法进行集成。

总之，在点格棋实际开发与应用中，利用十字链表法表示棋盘，使用监督学习方法离线训练得到的人工神经网络模型作为点格棋局面的评估函数，结合 UCT 搜索算法，使点格棋博弈系统达到了较高的智力水平，弥补了仅使用单一算法的不足。北京信息科技大学依据此方法编写的程序在全国计算机博弈大赛中连续三年获得冠军。

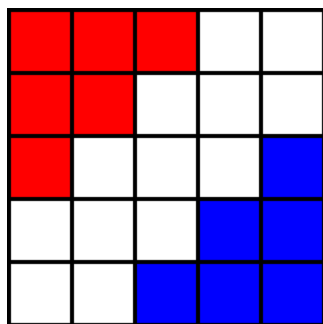
6.4 爱恩斯坦棋机器博弈

6.4.1 概述

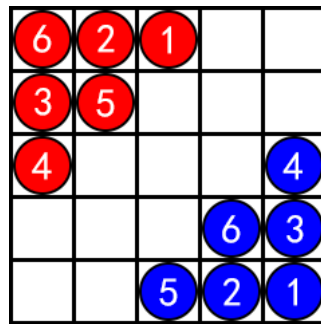
爱恩斯坦棋，德语 EinStein würfelt nicht! ^[145]，是德国耶拿的应用数学家 Ingo Althöfer 在 2004 年发明的棋盘游戏。爱恩斯坦棋是比较新颖的棋盘游戏，简单的规则下隐藏着极其复杂的分析计算与难以控制的博弈决策，蕴含着丰富的数学理论分析。引入骰子决定移动的棋子体现博弈过程中的不确定性，因此，爱恩斯坦棋是一种随机性的完备信息博弈。

爱恩斯坦棋规则如下：

(1) 爱恩斯坦棋采用 5*5 的方格棋盘^[1,146,147]，方格是棋子的移动位置，对弈双方分别标记为红方和蓝方，如图 6-7(a)所示。红方的出发位置位于棋盘左上角的三角区域，蓝方的出发位置位于棋盘右下角的三角区域。初始红蓝双方各有 6 颗标有 1-6 数字的棋子，游戏开始前，双方可在出发区位置自由布置己方棋子^[145]，6 颗棋子必须全部置于出发区之内，如图 6-7(b)所示。



(a) 棋盘开局

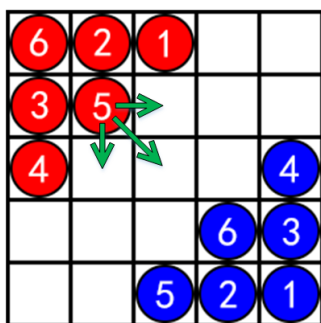


(b) 棋盘布局

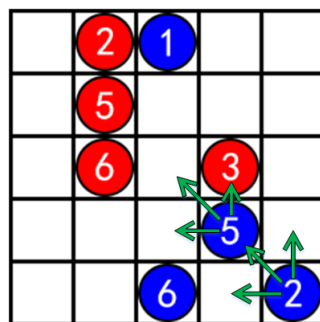
图 6-7 棋子开局与布局

(2) 布局完成之后，开始移动棋子，双方轮流行棋，但是每次行棋一方在移动棋子前须投骰子。如果骰子点数对应数字的棋子存在于棋盘上，必须走动棋子号与骰子数相同的棋子。如图 6-8(a)所示，轮到红方行棋，骰子投到 5，这时红方只能移动红 5。棋子每次只可移动一格，并且是相邻位置，不可跳格，移动方向为水平、竖直以及对角线方向。红方为右、下、右下，蓝方为左、上、左上。

(3) 如果骰子数对应数字的棋子不存在于棋盘上，就遵循就近原则，可以选择走动最接近此数字的棋子。如图 6-8(b)所示，蓝方投到骰子 3，现有蓝 1、蓝 2、蓝 5 和蓝 6，那么既可以选择走动蓝 5，或者蓝 2。



(a) 红方骰子投到 5 的行棋



(b) 蓝方骰子投到 3 的行棋

图 6-8 双方行棋

(4) 行棋过程中可以吃子。只要在棋子的行棋范围内，不论是吃掉己方棋子以增加己方其他棋子的灵活性，还是吃掉对方有威胁的

棋子都是允许的。如图 6-8(a)中，红 2 可以选择吃掉己方的红 1 或者红 5，图 6-8(b)中蓝 5 也可以吃掉有威胁的红 3。

(5) 赢棋的方式有两种：一种是率先占领对方的角部位置；另一种就是吃光对手所有的棋子。对弈只分胜负，不存在平局。

6.4.2 爱恩斯坦棋机器博弈主要技术

6.4.2.1 棋盘表示

爱恩斯坦棋棋盘可表示为一个 5*5 的矩阵，用一个二维数组来表示^[146]，存储棋子编号 ID，如图 6-9 所示。ID=20 + num (num 取值 1 到 6，表示 6 个棋子) 表示红方棋子，ID=10 + num 表示蓝方，ID=0 表示无棋子。棋盘上棋子发生变化时，数组中的数据值将发生改变。

$$\begin{bmatrix} 21 & 23 & 24 & 0 & 0 \\ 22 & 25 & 0 & 0 & 0 \\ 26 & 0 & 0 & 0 & 16 \\ 0 & 0 & 0 & 15 & 12 \\ 0 & 0 & 14 & 13 & 11 \end{bmatrix}$$

图 6-9 棋盘表示

6.4.2.2 棋局策略

对弈双方通过掷骰子来决定行棋，行棋一方无法得知对方下一步的行棋棋子，使得行棋方需要综合考虑对方所有在场棋子的行子概率及其应对策略，增加了棋局评估的复杂性。比赛的双方通过占领对方角位置或者吃完对方的棋子才能获胜，双方可以通过吃掉己方的棋子增加己方棋子的灵活性或者吃掉对方威胁性较大的棋子，从而占据优势^[148,149]。对弈过程中，对方可以吃掉我方的威胁性较大的棋子，同样我方也可以吃掉对方的威胁性较大的棋子。因此在相同的条件下，先手就显得特别重要。通常，每次掷骰子之后优先吃掉对方威胁性较大的棋子，或者不存在对方较大威胁性的棋子时可以吃掉己方的棋子来提高我方棋局灵活性。

6.4.2.3 搜索算法

搜索算法一直是计算机博弈研究中的热点，也是博弈系统中极其重要的组成元素。搜索算法的研究实质就是对博弈树的搜索研究。由于爱恩斯坦棋的随机性，其搜索算法大多采用在传统的极大极小算法的基础上引入概率，即期望搜索算法，又叫期望极大极小（Expect Minimax）算法。该算法通常被应用于双人零和博弈的特殊棋类游戏如西洋双陆棋等，这类棋类不但考验棋手的弈棋能力，运气也是很有影响的^[150]。

1. 期望搜索算法

在传统的 MAX 层与 MIN 层之间加入 CHANCE 层，用来评估投骰子随机事件发生的预期期望值。在传统的极大极小树中，MAX 层和 MIN 层交替出现直至达到固定搜索深度，MAX 和 MIN 层分别取其子结点效用值的最大和最小。但是在期望极大极小树中，如图 6-10 所示，CHANCE 层结点是交错穿插在 MAX 和 MIN 层之间，计算的方式也是取其子结点的加权平均效用值^[13,151,152]。

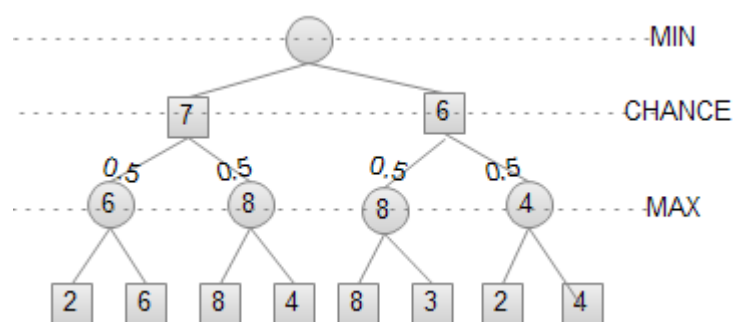


图 6-10 期望极大极小算法

期望搜索算法是通过对对手下一步行棋的预测评估来解决搜索过程中的随机性问题，将随机性问题转换为可以计算量化的概率问题，使得能够构建博弈树并进行搜索。

2. 攻防兼备的期望搜索算法

攻防兼备的期望搜索算法(An Offensive and Defensive Expect Minimax Algorithm, ODEMA)^[152]考虑爱恩斯坦棋规则的特殊性, 结合期望搜索算法, 在 MAX 层与 MIN 层之间加入 DICE 层, 用来模拟投骰子的过程, 构建博弈树。该算法综合进攻和防御双方面的考虑, 设计了进攻性估值、防御性估值以及威胁度估值 3 个方面的估值来全面评估棋子的进攻性和防御性, 力求准确高效地评估局面情况。

(1) 进攻性估值。爱恩斯坦棋主要赢棋的方式就是占领对方的角部位置, 从这个思路出发, 定义某一方的进攻值就是占领对手角部位置的评估值。

(2) 防御性估值。防御性为对方占领我方角部的期望的负值, 即对方进攻性的负值。计算对方到我方角部位置的期望是为了拦截对方到我方角部距离很近的对方的棋子。

(3) 威胁度估值。爱恩斯坦棋赢棋有两种方法, 一种是占领对角, 另一种是吃掉对方全部棋子。进攻性估值和防御性是从占领对角的方面考虑的, 该估值从吃子的角度来考虑。所谓威胁度, 就是当轮到敌方落子时, 我方棋子被对方棋子吃掉的期望, 亦即我方棋子受到对方棋子威胁的期望值, 也称对方对我方的威胁度。同理轮到我方落子时, 有我方对敌方的威胁度。

在攻防兼备的期望搜索算法中输入模拟棋子的相关属性包括棋子号, 棋子颜色、位置等, 设置搜索深度, 以及当前轮到何方下棋。搜索过程中如图 6-11 所示, 其中 MAX=0 属于骰子层。

从图 6-11 中, ODEMA 所构造的博弈树并不是一颗标准的完全博弈树, 在博弈树搜索扩展过程中, 随时可能出现末端结点即棋局结束。对于这些末端结点的估值就需要进行特殊对待, 同时对于骰子层也需要进行必要定义, 以此确定其模拟的是 MAX 层还是 MIN 层的投骰子过程。

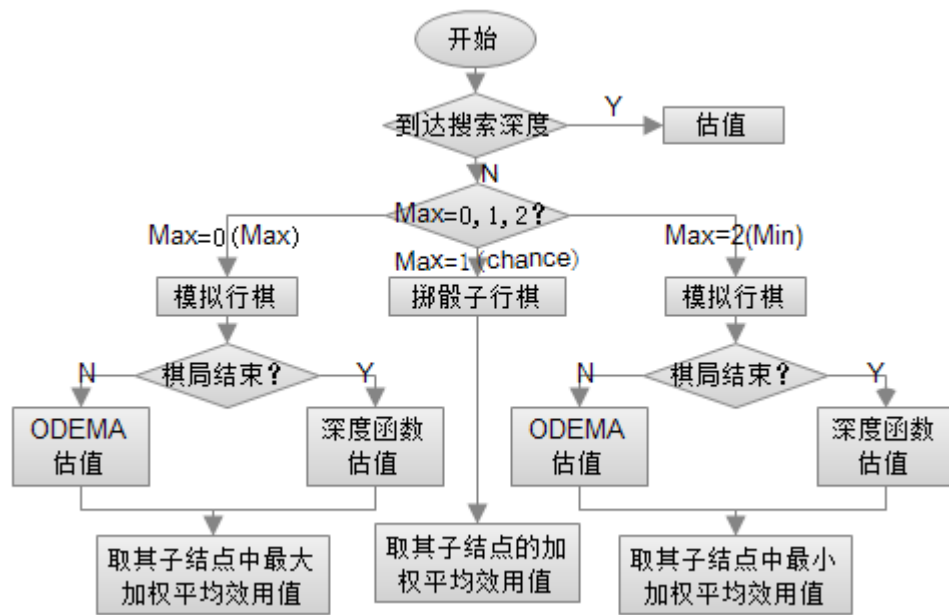


图 6-11 ODEMA 示意图

当然，在爱恩斯坦机器博弈中，常常会综合应用上述各类方法，甚至建立专家知识，通过搜索技术快速发现更优的或专家式着法。

第 7 章 非完备信息机器博弈的专项技术

7.1 德州扑克机器博弈

7.1.1 概述

德州扑克与围棋的区别在于德州扑克属于非完备信息博弈问题，是计算机博弈的另一分支。非完备信息机器博弈问题已被证明是一个 NP 难问题^[153]，一对一有限注德州扑克的状态复杂度约为 3.16×10^{17} ，包含其中的状态大多是无法确认的，有极大的随机性和不确定性，因此，德州扑克也是人工智能领域非常具有挑战性和代表性的博弈课题。图 7-1 展示了德州扑克的牌局实例。



图 7-1 德州扑克牌局

2008 年，德州扑克博弈系统 **Polaris** 首次战胜了职业扑克选手。2009 年，蒙特卡洛方法被引用于无限注德州扑克，并开始普遍应用。**Boris Iolis**^[154]提出了一种适用于扑克牌问题的选择策略，该策略以决策行为被选择的概率大小为依据，取得了较好效果；**Johannes Heinrich**^[155]提出了一种 **Kuhn poker** 的近似纳什均衡策略；2011 年，文献[156]中首次应用了模式匹配算法研究德州扑克游戏。2015 年，加拿大阿尔伯特大学发表了关于一对一有限注德州扑克系统的研究

成果，得到了该博弈问题的理论解。该研究小组开发的系统采用了反现实悔恨值最小化（Counterfactual regret minimization, 简称 CFR）算法，该算法通过多次的自对弈与评估过程，通过迭代得到近似的纳什均衡。2017 年，阿尔伯特大学在 Science 发表了关于一对一无限注德州扑克的 DeepStack 算法研究^[157]，DeepStack 是首个打败职业扑克玩家的计算机程序。

7.1.2 非完备信息动态博弈解的主要方法

7.1.2.1 CFR 算法

2007 年，加拿大阿尔伯塔大学的 Zinkevich 和 Johanson 提出了基于悔恨值最小化的 CFR 算法^[158]。CFR 算法的全称为 Counterfactual Regret Minimization，其中，Regret Minimization 即为悔恨值最小化。算法的核心在于博弈中的纳什均衡探寻。

$$R_i^T = \frac{1}{T} \max_{\sigma_i \in \Sigma_i} \sum_{t=1}^T (u_i(\sigma_i^*, \sigma_{-i}^t) - u_i(\sigma_i^t)) \quad (2)$$

悔恨值是在线学习中的概念。在扩展式博弈中，平均悔恨值的计算方法如公式（2）。其中， σ_i^t 是玩家 i 在第 t 轮游戏中所使用的策略，u 为玩家收益。悔恨值最小化算法就是将每步策略的收益与平均收益相比较，得到差值，并根据差值大小选择下一次的相应策略。在零和游戏中，如果双方玩家的平均悔恨值均小于 ε ，则可以看作达到了一个 2ε 均衡。

CFR 算法与普通悔恨值最小化算法的不同之处在于其将平均悔恨值分解为一系列的可加悔恨值项，即反现实悔恨值（counterfactual regret），因此可以分别进行最小化。反现实悔恨值定义在独立的信息集上，而平均悔恨值受限于反现实悔恨值之和。

$$R_i^T(I, a) = \frac{1}{T} \sum_{t=1}^T \pi_{-i}^{\sigma_i^t}(I) (u_i(\sigma_i^t |_{I \rightarrow a}, I) - u_i(\sigma_i^t, I)) \quad (3)$$

对于信息集 I 中的每一个 $\pi_{-i}^{\sigma_i^t}$ 可选行动 a，玩家 i 在时间 T

的反现实悔恨值如公式（3）所示。其中， σ_i 表示除玩家 i 外其他玩家依据策略 σ 达到当前信息集的概率。图 7-2 展示了 CFR 算法的迭代求解过程。

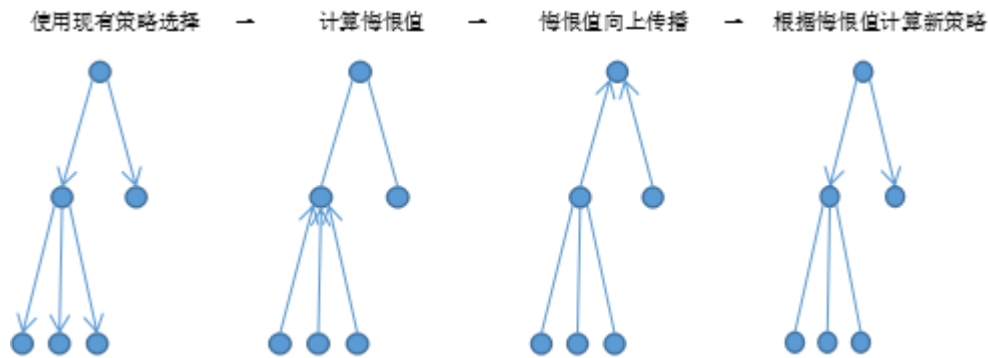


图 7-2 CFR 算法迭代过程

在近年来,CFR 算法及其变形广泛应用于扑克游戏中近似纳什均衡解的计算。在 2015 年,阿尔伯塔大学的 Bowling, Burch 与 Johanson 等研究人员以 CFR 算法为基础,提出了一种叫做 CFR+的新算法^[159],完成了一对一有限注德州扑克的求解。CFR 算法截取博弈过程的一部分进行迭代,而 CFR+算法对整棵博弈树迭代,且规定悔恨值必须为正。

7.1.2.2 DeepStack 算法

DeepStack 算法是于 2017 年由 CFR+算法的研究团队提出的又一新算法。与 CFR 算法不同的是,DeepStack 算法解决的是一对一无限注德州扑克问题^[157]。相对于一对一有限注德州扑克,无限注德州扑克的复杂度更高,因此也更难解^[160]。

DeepStack 算法由三个部分组成:针对当前公共状态的本地策略计算(local strategy computation)^[161],使用任意扑克状态的学习价值函数实现有限深度的前瞻(depth-limited lookahead)^[162],以及预测动作的受限集合^[163]。

此外,DeepStack 还采用了深度神经网络(Deep neural networks ,

DNNs)^[164]分别训练了在发下三张公共牌后（flop network）、发下第四张公共牌后（turn network）价值的估计。深度神经网络使用了七个全连接隐含层，每层 500 个节点。训练样本分别为 1,000,000 盘与 10,000,000 盘游戏。网络得到的输出为各玩家在各种手牌情况下评估值组成的向量。

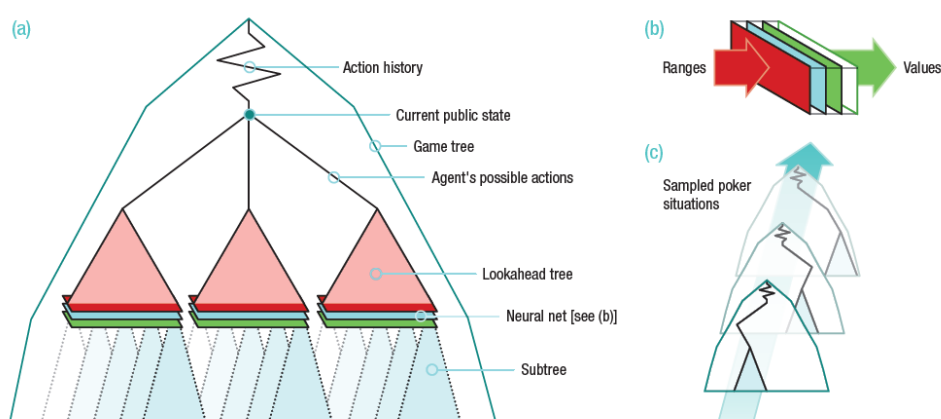


图 7-3 DeepStack 算法概览

图 7-3（a）中，在每一个公共状态中，DeepStack 使用有限深度的前瞻估计当前局面，前瞻时子树的估值使用训练好的深度神经网络（b）计算。而（b）中神经网络的训练样本为由（c）随机生成的扑克局面。

7.2 军棋机器博弈

7.2.1 概述

军棋又称为陆战棋，是我国广大人民群众在抗日战期间发展完善的。常见的有二人军棋和四国军棋。相对四国军棋，由于二人军棋不需要对家配合，而且棋局状态相对简单，适于作为研究非完备信息博弈的入门项目。2012 年，二人军棋首次被纳入中国计算机博弈大赛。

二人军棋在开局时，只能根据军旗布子、炸弹布子、地雷布子等规则限定，估计对手棋子军阶分布信息，结合人类以往布局经验，获

得初始每个位置布子可能性。在博弈过程中，通过双方碰子、走子情况，进一步获得对手每个棋子可能性信息。残局时，随着大量碰子走子，双方收集对手棋子信息越来越多，棋局由暗棋趋变为明棋，可以使用完备信息博弈技术求解。

二人军棋人人博弈时，经常会出现骗招、无意义磨棋、心理对抗，但在人机博弈或是机机博弈过程中，当前计算机博弈技术没有充分考虑人类的行为，这是国内二人军棋计算机博弈程序不能与人类中等水平抗衡的根本原因之一。

二人军棋棋局存在大量异型等价的状态，且其具有随机性非完备信息博弈的特点使得每次棋子碰撞的结果都不确定。通常，博弈搜索深度不需要太深，搜索深度 10 步以内完全可以应付一些极端情况。传统的评估函数设计相对简单，因此应更多考虑静态子力价值，适当考虑位置控制因素，对于有可能安置军旗的位置（如大本营）重点控制。

7.2.2 军棋机器博弈主要技术

在军棋博弈技术中重点需要解决以下三个问题：对手棋子可能性矩阵进行更新问题；欺诈走法的选择和判定；搜索技术的选择。下面分别针对以上三个问题进行说明：

1. 对手棋子可能性矩阵更新

随着棋局的变化，棋盘上的棋子分布概率也会发生一些变化。举个具体的例子，假定对手 A 棋子初始可能性向量是(0.06, 0.06, 0.08, 0.08, 0.08, 0.08, 0.12, 0.12, 0.12, 0.12, 0.08, 0)，表示的是（令，军，师，旅，团，营，连，排，兵，雷，炸，旗）的可能性，所有可能性和为 1。当 A 棋子与己方营长碰撞，A 棋子胜利，那么 A 棋子就只可能为令，军，师，旅，团，可能性向量可以直接转换为(0.06, 0.06, 0.08, 0.08, 0.08, 0, 0, 0, 0, 0, 0, 0)，但是可能性和要求为 1，将其按比例简单归一计算得到(0.17, 0.17, 0.22, 0.22, 0.22, 0, 0, 0, 0, 0, 0, 0)。但是这样的可

能性更新会间接影响到对手 B 棋子的可能性。假定原来对手 B 棋子可能性向量也是(0.06, 0.06, 0.08, 0.08, 0.08, 0.08, 0.12, 0.12, 0.12, 0.12, 0.08, 0)，由于 A 棋子胜过营长，那么 B 棋子大于营长的可能性就应该减小。根据 A 棋子更新过后的可能性向量，B 棋子的可能性必然发生更新，并且也要求作归一处理。再进一步，需全盘考虑，不能只看对手 A、B 两个棋子，而是要把对手所有棋子统一综合考虑，所以 A 棋子的可能性向量更新将会导致对手所有棋子的可能性向量更新。25 个对手棋子，应该构成一个对手棋子 25×12 的可能性二维矩阵，如图 7-4 所示。

令	军	师	旅	团	营	连	排	兵	雷	炸	旗
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.02, 0.02, 0.03, 0.03, 0.03, 0.03, 0.05, 0.05, 0.05, 0.17, 0.04, 0.50,											
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.02, 0.02, 0.03, 0.03, 0.03, 0.03, 0.05, 0.05, 0.05, 0.17, 0.04, 0.50,											
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.03, 0.03, 0.06, 0.06, 0.06, 0.06, 0.09, 0.09, 0.09, 0.33, 0.08, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.09, 0.09, 0.09, 0.09, 0.14, 0.14, 0.14, 0.00, 0.13, 0.00,											
0.05, 0.05, 0.11, 0.11, 0.11, 0.11, 0.16, 0.16, 0.16, 0.00, 0.00, 0.00,											
0.05, 0.05, 0.11, 0.11, 0.11, 0.11, 0.16, 0.16, 0.16, 0.00, 0.00, 0.00,											
0.05, 0.05, 0.11, 0.11, 0.11, 0.11, 0.16, 0.16, 0.16, 0.00, 0.00, 0.00,											
0.05, 0.05, 0.11, 0.11, 0.11, 0.11, 0.16, 0.16, 0.16, 0.00, 0.00, 0.00,											

图 7-4 棋子可能性矩阵

棋子 A 是各种军阶的可能性总和为 1，即每一行总和为 1。每个军阶可能是哪些棋子的可能性初始总和为 1，即每一列初始总和为 1。但是随着棋局动态更新，无法同时行、列归一。

通过棋子走子、碰子结果，借鉴图像学领域中的图模型推理棋子概率分布。常见的推理方法主要分为精确法和近似法，理论上，所有的图模型推理都可以用精确算法实现。但 Cooper 于 1990 年指出了概率模型下的精确推理是 NP-hard 问题^[165]，直接使用精确推理方法效率很低。

信念传播算法^[166]是一种迭代求解概率图模型^[147,167]的推理方法。该算法精髓是计算局部消息传递，从而可以计算结点的边缘概率分布。当裁判给出信息之后，比如棋子碰撞之后，所得结果为胜、负、平，该结果信息对于对手发生碰撞的棋子的可能性有一定影响，会间接影响到对手其他棋子的可能性。通过信念传播算法可以较好的解决可能性更新的问题。

2. 欺诈走法的选择和判定

在人类二人军棋暗棋对战中，常常发生欺诈走法，欺诈走法虽然不能直接获取利益，但是可以让对手判断失误，在战略上赢得筹码获得主动。欺诈走法运用了人类心理层面的一些东西，理论上使用传统博弈技术无法根本解决。在人机博弈中，只有建立欺诈数学模型并且结合搜索技术求解、实现可以实施欺诈走法和判定欺诈的 AI，才能够战胜人类。

3. 搜索技术的选择

由于军棋的暗棋特性，将传统完备信息博弈技术应用于军棋，效果并不好。现在很多非完备博弈程序都直接或者间接的使用直接转换的方法，生成一个基于当前的信息完备局面，再进行着法搜索。因为直接转换是猜测可能性最高的局面，但在中、前期搜集信息还不充分情况下，猜测局面和真实局面相差太远，因此在实践当中，直接转换

的方法效果并不理想。

军棋规则导致可能出现大量磨棋的走法，博弈程序难以找到一个稳妥有效或者有风险激进的策略，只能寻找貌似风险最小、但毫无意义的招法（比如棋子来回进出行营）。使用蒙特卡洛方法进行模拟时，当受时间空间所限，程序不能进行足够多的模拟，得出决策就会与上述情况类似。因此，必须需要一些新的方法进行改进博弈策略。

确定性聚合 UCT 算法是通过对多种状态空间（即多个可能的完备信息棋盘状态）进行搜索。虽然前中期搜集到的棋盘信息不充分，但是可以对可能局面进行抽样，根据已搜集到的少量信息，排除不可能局面，留下可能性高于阈值的局面。针对每个局面使用 UCT 算法进行搜索，求解每个局面行棋着法的胜率，再根据该局面可能性权重加权求和，取最大聚合胜率的行棋着法为行棋策略。

7.3 桥牌机器博弈

7.3.1 概述

桥牌是由 17 世纪的一种叫做“惠斯特”的纸牌玩法演化而来的，起源于英国。桌上四人，南北为一队，东西为一队，按顺时针方向进行游戏。开始打牌前，双方通过叫牌确定定约。确定定约后，庄家的下家首先攻牌，然后庄家的队友把自己的牌亮开让大家都能看见，称为明手，之后明手由庄家指挥出牌。最终，根据庄家完成定约的情况进行计分。图 7-5 中分别显示了叫牌过程和首攻后的打牌状态。

桥牌和一般牌类不同的地方在于，通过叫牌阶段的一些约定，可以传递一些实力、花色长度、牌型、是否做庄的意图等信息；而打牌阶段，防守方还可以约定一些出牌顺序、出牌花色等防守信号，以此传递自己对某个花色的鼓励、反对、奇偶牌张、转攻花色等合作态度信息。这些信息的传递是桥牌博弈中的重要组成部分。例如，叫牌阶段可以通过一些叫牌过程促使队友做出对我方有利的攻击，甚至是防守方通过欺骗信息或出牌、跟牌，以欺骗对手并让己方获利、完成己

方目标的目的。



图 7-5 桥牌叫牌和首攻后状态

因此，桥牌的博弈过程可以归纳为三个方面：

- (1) 信息收集、分析、传递与对抗；
- (2) 同伴之间的合作协议设计（包括叫牌规则和防守信号）；
- (3) 计划、决策实施与计划调整，做出叫牌和打牌计划，并根据实施过程中的情况及时调整。

7.3.2 桥牌机器博弈的主要方法

桥牌的牌面分布的复杂度是 $C_{52}^{13}C_{39}^{13}C_{26}^{13}C_{13}^{13} \approx 10^{30}$ ；

叫牌阶段看不见别人的牌，只能看到自己的牌，其他人牌面的可能性为 $C_{39}^{13}C_{26}^{13}C_{13}^{13} \approx 10^{17}$ ；

打牌阶段能看见自己和明手的牌，另外两手牌的可能性为 $C_{26}^{13}C_{13}^{13} \approx 10^8$ ，每手牌的出牌可能性约为 10^{21} ，因此打牌阶段最复杂的情况大约在 10^{29} 。

在这样一个量级的问题规模，采用常规的暴力搜索是不能解决桥牌机器博弈问题的。

7.3.2.1 双明手算法

如果在一副牌完全确定的情况下，即包括四家的牌、庄家、定约都是知道的，称之为双明手。双明手情况下的打牌就变成了一个完整信息的搜索最优解的过程。寻找最佳打牌路线的博弈树的规模大约是

$13! \cdot 2^{39} \approx 10^{21}$ 。

双明手算法早在 1996 年,纽约州立大学的 Chang 博士的论文《构造一个快速双明手求解器》就提出了。结合桥牌专家技术,利用 Alpha-Beta 剪枝、哈希表、单套分析等技术,绝大部分牌例都能够在很短的时间内得到结果。在这个基础上,重庆大学、辽宁科技大学都有算法优化的改进论文发表,提升了算法的运行效率^[168,169]。图 7-6 是一副牌双明手结果的呈现。

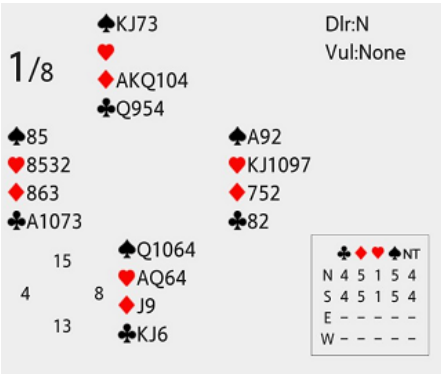


图 7-6 双明手结果图

双明手算法的突破是目前解决桥牌机器博弈的基础,它把不确定性的问题转换为基于不确定性的猜想进行确定性的计算。

7.3.2.2 蒙特卡罗抽样模拟

结合在叫牌和打牌过程中传递的信息,可以对各家进行信息建模,包括大牌分布、花色分布、关键牌张信息等。利用这些信息,可以对其它各家的牌进行抽样分析,这样可以得到若干个牌面明确的样本,对每个样本再使用双明手算法获得确定的结果,进而得到在当前局面下叫牌和出牌回报的数学期望。AI 可以结合数学期望和必要的专家知识进行决策,从而实施叫牌和出牌。

出牌信息如图 7-7 所示,机器人根据叫牌信息模拟,分析其它方可能的牌张,并对各种情况进行双明手计算,综合各种情况下的回报

期望，最终选择出牌红心 A。

```
[0 ] 873 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK5.KT6.J3.KJT72 T2.Q7532.QT64.83 -> HA(3/0/2-9/0/9)
[1 ] 4216 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 A5.KQ5.Q64.KJT72 KT2.T7632.JT3.83 -> HA(3/0/3-10/0/8)
[1 ] 3230 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 K2.KQ62.QJ3.KJ82 AT5.T753.T64.T73 -> HA(6/0/3-7/0/7)
[0 ] 59098 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AKT.KT5.Q6.KJ873 52.Q7632.JT43.T2 -> HA(3/0/2-10/0/9)
[5 ] 47781 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AKT.Q752.QJT6.K3 52.KT63.43.JT872 -> HA(4/0/3-9/0/5)
[1 ] 10815 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AT2.KQ62.QT.KJ87 K5.T753.J643.T32 -> HJ(4/0/2-9/0/7)
[1 ] 23587 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 KT.KQT2.QJ4.KJ87 A52.7653.T63.T32 -> HJ(3/0/3-10/0/5)
[1 ] 45008 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK5.Q65.QJT6.KT7 T2.KT732.43.J832 -> HA(3/0/3-10/0/6)
[0 ] 6623 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK5.K762.J4.KJ32 T2.QT53.QT63.T87 -> HA(3/0/2-8/0/8)
[0 ] 26756 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 K5.KQ7.QJ6.KJT82 AT2.T6532.T43.73 -> HA(5/0/3-8/0/8)
[4 ] 39933 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AKT.KQT.QJ6.JT83 52.76532.T43.K72 -> HJ(2/0/2-11/0/6)
[3 ] 50768 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK5.KQT.QJT3.872 T2.76532.64.KJT3 -> HJ(2/0/2-11/0/6)
[0 ] 34716 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK2.KQT.QJT43.J7 T5.76532.6.KT832 -> HJ(2/0/2-11/0/6)
[0 ] 52969 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AT5.KQ7.QJ6.K873 K2.T6532.T43.JT2 -> HJ(3/0/3-10/0/6)
[2 ] 34424 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK5.KQT.T643.KT7 T2.76532.QJ.J832 -> HA(2/0/2-8/0/8)
[2 ] 36136 0 : N:Q74.984.K9875.AQ J9863.AJ.A2.9654 AK2.0652.0JT.KJ7 T5.KT73.643.T832 -> HA(3/0/3-10/0/6)
```

图 7-7 出牌信息图

由于蒙特卡罗抽样的依据来源于信息建模，因此敌方和同伴信息的有效性和完整性分析是能否获得最佳解的关键点。同时有效向同伴提供信息、对敌方隐藏甚至提供欺骗信息是桥牌获胜的重要技术。

信息既有对抗又有合作，这也是目前其它棋牌机器博弈中不具备的特点，是未来桥牌 AI 战胜人类必须突破的关键点。

7.3.2.3 专家系统

桥牌机器博弈中专家系统主要体现在叫牌体系的设计上。一套优秀的叫牌体系设计，能够让同伴获得更清晰的信息，并且减少敌方获取更多我方信息，从而在博弈过程中获得优势。

目前常见的做法是基础的框架采用专家编写叫牌博弈树，AI 查表，后期采用蒙特卡罗模拟结合自然叫牌规则实现。专家编写叫牌博弈树的复杂度从几万到几十万不等。

7.3.2.4 桥牌机器博弈技术的未来

由于桥牌问题的复杂度足够大，而要让 AI 具备甚至超过人类牌手的水平，需要让 AI 的思维向人类一样细腻、严谨，并具备人类牌手在心理上的合作、对抗能力，而这依靠传统的搜索算法、专家系统和模拟决策过程是不够的，这样的 AI 很难根据对手的特点和不同的

局面及时调整自己的策略。

因此，桥牌 **AI** 的发展，以下几个方向值得探索：

（1）信息收集分析和置信度的动态调整，提升蒙特卡罗抽样模拟的有效性；

（2）降低学习的状态空间，使得向人类牌手学习乃至自博弈的增强学习成为可能；

（3）学习德州扑克 **AI** 的成功经验，建立对手模型和伙伴模型；

（4）通过机器自博弈，找到一套最佳的叫牌、打牌约定系统。

第 8 章 机器博弈的国内外赛事

8.1 国际机器博弈赛事

8.1.1 国际象棋人机博弈大赛

(1) 1958 年，名为“思考”的 IBM704 成为第一台能同人下棋的计算机，思考速度每秒 200 步。

(2) 1983 年，BELLEAT&T 开发了国际象棋硬件，达到大师水平。

(3) 1987 年，“深思”以每秒钟 75 万步的思考速度露面，其水平相当于拥有国际等级分为 2450 的棋手。

(4) 1988 年，“深思”击败丹麦特级大师拉尔森。

(5) 1989 年，“深思”已经有 6 台信息处理器，每秒思考速度达 200 万步，但与世界棋王卡斯帕罗夫的人机大战对阵，以 0 比 2 败北。

(6) 1990 年，“深思”第二代产生，使用 IBM 的硬件，吸引了前世界棋王卡尔波夫与之对抗。

(7) 1993 年，“深思”二代击败了丹麦国家队，在与前女子世界冠军小波尔加的对抗中获胜。

(8) 1996 年 2 月，“深蓝”诞生，使用新的集成电路将思考速度提高到每秒 300 万步，其棋力（性能）高于“深思”数百倍，但在美国费城与卡斯帕罗夫的挑战赛中，“深蓝”以 2 比 4 失利。

(9) 1997 年，“更深的蓝”开发出了更加高级的“大脑”，4 名国际大师参与 IBM 的挑战小组，为电脑与卡斯帕罗夫重战出谋划策，最后“更深的蓝”以 3.5 比 2.5 击败了卡斯帕罗夫，卡斯帕罗夫要求重赛，但没有得到回应。这场比赛，面对棋王卡斯帕罗夫而坐的并不是计算机，而是深蓝研制小组的代表、华人许峰雄博士。这场胜利引起了世界范围内的轰动，它表明计算机智能战胜了人类天才。

(10) 2001 年，一家德国公司开发的国际象棋软件“更弗里茨”在德国波恩击败了卡斯帕罗夫、阿南德以及除了克拉姆尼克之外的所

有排名世界前十位的棋手。

(11) 2002 年 10 月，“更弗里茨”与克拉姆尼克在巴林进行人机大战，思考速度为每秒 600 万步，双方以 4 比 4 战平。

(12) 2003 年 1 至 2 月，由两位以色列电脑专家研究出的“更年少者”与卡斯帕罗夫举行人机大战，双方 3 比 3 战平。

(13) 2004 年 6 月，国际象棋特级大师诸宸首度代表中国棋手参与到了人机大战中，但是她在与“紫光之星”笔记本电脑的对抗中以 0: 2 败下阵来。

(14) 2005 年，九头鸟以 5.5 比 0.5 大胜英国棋王亚当斯。

(15) 2006 年 11 月，连续三届奥林匹克冠军克拉姆尼克再战更弗里茨，结果 2 比 4 败北。

8.1.2 围棋人机与机机博弈大赛

(1) “UEC 杯”世界计算机围棋比赛 2007 年始于日本，每年邀请各国高水平 AI 齐聚东京比赛，促进相关学术及科技的交流。日本的 DeepZenGo、法国的“疯石”（Crazy Stone）、美国 Facebook 公司的“黑暗森林”（Dark Forest）等世界著名计算机围棋程序先后在 UEC 杯折桂获奖。

2017 年 3 月，第 10 届 UEC 杯计算机围棋大赛在东京落幕，今年共有 30 支软件参赛，但 AlphaGo 没有参加。腾讯 AI Lab（腾讯人工智能实验室）研发的围棋人工智能程序“绝艺”（Fine Art）首次参加比赛便一路过关斩将，夺得本届 UEC 杯冠军，

“电圣战”是由电气通信大学（简称：UEC）与日本棋院于 2013 年创办的，是“UEC 杯”世界计算机围棋大赛的姊妹赛事，由“UEC 杯”冠亚军与人类职业棋手对战。2017 年的“绝艺”和“DeepZenGo”均战胜了日本年轻棋手一力辽七段。

(2) 2015 年 11 月，美林谷杯首届世界计算机围棋锦标赛暨人机大战在北京工体网鱼电竞举办，来自中国、韩国、日本、美国、法国、

捷克和中华台北等 7 个国家和地区的 9 支代表队参加了比赛。在 9 支参赛队伍中，韩国队的围棋软件石子旋风（DolBaram）夺得冠军，日本“老将”Zen 屈居亚军，美国队的 Many Faces of Go 获得季军。

中国围棋名人战的新科冠军连笑七段与“石子旋风”进行了让子棋的三番棋决战，最后成绩为 2:1，“石子旋风”在受让六子的情况下，勉强扳回一局。

(3) 2016 年 1 月，谷歌公司的团队在自然杂志（Nature）上发表封面论文称，他们研发出能够在极其复杂的围棋游戏中击败专家级人类选手的计算机。实际比赛发生在 2015 年 10 月，在英国伦敦，谷歌公司下属的 DeepMind 团队开发的 AlphaGo 电脑程序以 5:0 的战绩击败了三届欧洲围棋冠军、职业围棋二段樊麾（Fan Hui）。

(4) 2016 年 3 月，在韩国首尔，谷歌公司的 AlphaGo 以 4:1 的战绩战胜世界围棋冠军李世石，在学术界产生了空前的影响，这是机器博弈发展史上的一次重大胜利。

(5) 2017 年 1 月，Master 围棋（最新版 AlphaGo）在棋类网站上与中日韩数十位围棋高手进行快棋对决，取得了网测 60 场全胜的战绩。这一事件再一次引起了学术界的轰动，引发了科技人员对人工智能更深层次的关注和思考。

(6) 2017 年 5 月，在中国浙江的乌镇桐乡，升级版的围棋人工智能 AlphaGo 对战排名世界第一的中国棋手九段柯洁，柯洁 0:3 负于 AlphaGo，再次验证了超算、大数据和深度学习算法相结合的技术实力。

(7) 2017 年 8 月，首届世界智能围棋赛在内蒙古自治区伊金霍洛旗收官，日本智能棋手 DeepZenGo 斩获冠军，台湾的人工智能程序 CGI 获亚军，腾讯的人工智能“绝艺”尽管拼尽全力，但最终半决赛遗憾告负。本次比赛，来自台湾交通大学吴毅成教授团队的人工智能程序 CGI 成为赛场中最大的一匹黑马。

8.1.3 桥牌计算机博弈大赛

从 1997 年开始，由世界桥牌联合会每年举办一次世界计算机桥牌锦标赛，大赛网址：<https://bridgerobotchampionship.wordpress.com/>。2016 年 9 月，第 20 届世界计算机桥牌锦标赛在波兰西南部城市弗罗茨瓦夫举行，共有来自法国、丹麦、德国、日本、美国、荷兰和中国 7 个国家的 8 支队伍参加本次锦标赛。

中国新睿桥牌开发团队报名参加了 2016 年比赛，新睿桥牌成为第一个参加世界计算机桥牌锦标赛的中国软件，在循环赛结束，新睿桥牌位列第 6 名，表明中国桥牌 AI 已经在全方位的走向世界。

从 2001-2016 年的成绩来看，桥牌计算机博弈大赛的前 4 名被 6 个软件开发团队垄断。他们是荷兰 Jack、法国 WBridge、丹麦 Shark Bridge、日本 Micro Bridge、德国 Q-plus Bridge 和美国 Bridge Baron。

2017 年 8 月，第 21 届世界计算机桥牌锦标赛在法国里昂落幕，经过为期 6 天的激烈角逐，法国的 Wbridge5 蝉联冠军，中国的新睿桥牌(Synrey Bridge)获得亚军，获得并列第三名的是荷兰的 RoboBridge 和日本的 Micro Bridge。新睿桥牌是第二次参加本项大赛，表明中国在世界桥牌人工智能领域，已经走向世界的前列。

8.1.4 德州扑克人机与机机博弈大赛

(1)自 2006 年开始，计算机扑克程序比赛（ACPC，Annual Computer Poker Competition）每年举办一次，该比赛由 Alberta 大学与 Carnegie Mellon 大学联合人工智能领域内的两个顶级国际会议 AAAI 和 UCAI 共同举办，其中德州扑克一直是大赛的主要竞赛项目。

哈尔滨工业大学王轩教授团队多次参加这项比赛，2013 年首次参加 ACPC 比赛，在循环赛中就击败了冠军，最终获得决赛阶段多人德州扑克项目第 4 名；2014 年 ACPC 比赛，取得决赛阶段三人 Kuhn Poker 项目（Three player Kuhn poker）第 3 名，获得决赛阶段多人德州扑克项目第 4 名；2016 年 ACPC 比赛，二人非限制性德州扑克

(Hheads-up No-Limit Texas Hold'em)，获得决赛阶段第 4 名。

(2) 2015 年 4 月，在美国匹兹堡的河流赌场举行了德州扑克人机大战，卡耐基梅隆大学的 Claudico 在与包括当时世界排名第一的道格·波尔克 (Doug Polk) 在内的四名人类顶尖高手过招，最终初出茅庐的 Claudico 以失利告终。

(3) 2017 年 1 月，德州扑克人机大战在美国匹兹堡再次举行，卡耐基梅隆大学 Claudico 的升级版——冷扑大师 (Libratus) 在比赛中轮流击败了四名顶尖人类高手。

(4) 2017 年 4 月，由中国创新工场 CEO 李开复发起的人工智能与真人对打的德州扑克赛事——冷扑大师对弈中国龙之队表演赛在海南生态软件园开赛，冷扑大师完胜中国龙之队。

8.1.5 其它赛事与研究成果

(1) 国际计算机博弈比赛 (CO -Computer Olympiad) 由国际机器博弈协会 (ICGA -International Computer Games Association) 每年组织一届，已经有了 30 多年的历史，竞赛项目数已达 20 多种，是当前国际计算机博弈领域最具影响和权威的比赛，通过竞赛推动了计算机博弈在世界范围内的发展，大赛网址：<http://icga.leidenuniv.nl/>。

(2) 五子棋 AI 比赛由捷克布拉格查尔斯大学的学生于 2000 年创办，每年举行一次，五子棋 AI 大赛网址：<http://gomocup.org>。

(3) 2007 年美国科学杂志 (Science) 评出的人类 10 大科学突破中，包括加拿大阿尔波特大学科学家历时 18 年破解了国际跳棋 (64) 的研究成果，这是整个机器博弈发展史上的一个里程碑。

8.2 国内机器博弈赛事

8.2.1 中国象棋人机与机机博弈大赛

(1) 2005 年 9 月，由“岭南双雄”吕钦、许银川领军的征战全国象棋甲联赛的广东王老吉队 5 员战将，以总比分 1 胜 4 和 5 负不敌由电脑与 5 位业余高手组成的“人机合一”网络联队。

(2) 2006 年 8 月，由中国人工智能协会和浪潮集团主办的首届中国象棋人机大战在北京中国科技馆开战，有中国象棋第一人之称的中国象棋特级大师许银川，在北京与“浪潮天梭+棋天大圣”进行人机对决，双方通过 2 盘角逐，打成平局。

(3) 2016 年 5 月，2016 “楚河汉界杯”亚洲象棋人工智能对决邀请赛在中国河南省荥阳市举行，象棋名手、象棋旋风、象棋天启三款软件分别获得第一、二、三名。

8.2.2 中国计算机博弈大赛

从 2006 年开始，中国计算机博弈大赛每年组织一次，大赛创始人为东北大学徐心和教授，大赛网址：<http://computergames.caaai.cn/>。表 8-1 为历届中国计算机博弈大赛的参赛数据。中国计算机博弈比赛经历了二个阶段：

第一阶段：2006~2010，全国计算机博弈锦标赛阶段，由中国人工智能学会主办，这是比赛发展的初期。

第二阶段：2011~至今，全国大学生计算机博弈大赛暨全国计算机博弈锦标赛阶段，由中国人工智能学会和教育部高等学校计算机类专业教学指导委员会共同主办。而且 2015 年该比赛被国家体育总局棋牌运动管理中心纳入第三届全国智力运动会项目。表 8-1 为历届中国计算机博弈大赛的参赛数据。

目前比赛共设置 18 个项目，其中仅面向大学生的计算机博弈大赛项目包括：五子棋、六子棋、点格棋、苏拉卡尔塔棋、亚马逊棋、幻影围棋、不围棋、爱因斯坦棋、军棋、海克斯棋 10 种棋类；面向全社会的锦标赛项目包括：中国象棋、围棋、13 路围棋、9 路围棋、国际跳棋（100 格）、国际跳棋（64 格）、二打一扑克牌（斗地主）和桥牌 8 种棋牌类。

表 8-1 历届中国计算机博弈大赛的参赛数据

年 份	承办单位 或比赛地点	棋类项目数	参赛队数	参赛学校 (单位) 数
2006	北京中国科技馆	1	18	7
2007	重庆工学院	4	43	11
2008	北京 ICGA 国际赛事	13	77	14 个国家
2009	深圳大学城	4	32	8
2010	北京理工大学	8	53	18
2011	北京科技大学	8	93	24
2012	东北大学	13	163	25
2013	哈尔滨工程大学	13	168	24
2014	成都理工大学	14	203	35
2015	中国棋院	17	222	35
2016	沈阳航空航天大学	17	225	40
2017	重庆理工大学	18	245	42

通过竞赛不断提升了我国计算机博弈水平，一些棋种的冠军得主每年在更替，产生了北邮的“本手”和北工大的“深石”围棋、蒋志敏的“象棋名手”和东大的“棋天大圣”中国象棋等知名软件，促进了计算机博弈技术的普及与发展，也培养了大学生的科技创新精神。

第9章 结束语

2017年7月20日，国务院印发了《新一代人工智能发展规划》（国发[2017] 35号），规划指出：人工智能发展进入新阶段。经过60多年的演进，特别是在移动互联网、大数据、超级计算、传感网、脑科学等新理论新技术以及经济社会发展强烈需求的共同驱动下，人工智能加速发展，呈现出深度学习、跨界融合、人机协同、群智开放、自主操控等新特征。规划还指出：支持开展形式多样的人工智能科普活动，鼓励广大科技工作者投身人工智能的科普与推广，全面提高全社会对人工智能的整体认知和应用水平。支持开展人工智能竞赛，鼓励进行形式多样的人工智能科普创作。鼓励科学家参与人工智能科普。

本白皮书恰逢国家新一代人工智能发展规划发布之际形成，更具有时代意义，希望它能更好的发挥宣传和科普的作用。本书简要介绍了机器博弈的发展过程、现状、面临的挑战和发展趋势；分析了机器博弈的状态复杂度、博弈树复杂度和计算复杂度等；描述了机器博弈的常用搜索、评估和优化技术：穷尽搜索、裁剪搜索、启发式算法、迭代算法、最佳优化算法、随机搜索算法、遗传算法、并行计算、神经网络、机器学习等；论述了开发机器博弈平台系统的意义、分类、设计规范和相关技术；以六子棋、围棋、点格棋和爱因斯坦棋为代表，介绍了完备信息博弈的关键技术，以德州扑克、军棋、桥牌为代表，介绍了非完备信息博弈的关键技术；最后介绍了机器博弈相关国内外赛事，为从事机器博弈及其相关领域的科研和应用人员提供参考。

各种形式的计算机博弈竞赛，极大地促进了学术交流，检验了新技术，推动了机器博弈的快速研究与发展。我国计算机博弈发展历程虽然只有10余年时间，但是在近几年的全国比赛中，平均每年有220多支代表队参赛，而且竞赛项目已达到18项，这一规模绝对可以和

目前国际上的任何一个计算机博弈比赛的规模相媲美。当然，我们的水平还不是很高，尤其是在科学研究方面和国际先进水平还有较大的差距，但是我们相信这种差距会不断地缩小，我们不仅要成为计算机博弈大国，更要成为计算机博弈的强国。

伴随着人工智能科学发展的 60 多年，计算机博弈经历了起步、发展、成熟、飞跃四个阶段。当前完备信息博弈技术相对比较成熟，非完备信息博弈和随机类博弈技术还需进一步发展。深度学习算法在 AlphaGo 围棋计算机博弈中的成功应用，不仅学术意义巨大，而且实用性也很强，引发了世界范围内对人工智能技术的高度关注，调动了更多的专家学者开展深入研究的积极性。尽管在计算机博弈领域还存在着各种各样的问题，许多工作还需要向更广领域和更深层次推进，但是随着研究人员的不断增加以及计算机博弈技术在各个领域的广泛应用，将会产生越来越多的研究成果。

可以预见，计算机博弈作为人工智能一个颇有发展前途的研究领域，在未来将与其它领域的技术更广泛、更紧密地融合，推动人工智能技术与相关产业快速发展，将人工智能带上了一个新的台阶。

参考文献

- [1] 王骄, 徐心和. 计算机博弈. 人工智能的前沿领域——全国大学生计算机博弈大赛[J], 计算机教育, 2012, (7): 14-17.
- [2] 王亚杰, 邱虹坤, 尹航, 等. 计算机博弈竞赛与创新人才培养模式改革[J]. 实验技术与管理, 2016, 33(10):10-14.
- [3] Thomas H Cormen, Ronald L Rivest, Charles E Leiserson, Clifford Stein. Introduction to Algorithms[M], Cambridge: Mit Press, 2009.
- [4] Bench-Capon T. J. M., Dunne P. E.. Argumentation in artificial intelligence [J], Artificial Intelligence, 2007, 171(10-15): 619-641.
- [5] Takeuchi S, Kaneko T, Yamaguchi K. Evaluation of Game Tree Search Methods by Game Records[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2011, 2(4):288-302.
- [6] Marsland T A, Campbell M. Parallel Search of Strongly Ordered Game Trees[J]. Acm Computing Surveys, 1982, 14(4):533-551.
- [7] Lieberum J. An evaluation function for the game of amazons[J]. Theoretical Computer Science, 2005, 349(2):230-244.
- [8] Geoffery E. Hinton, Salakhutdinov RR. Reducing the Dimensionality of Data with Neural Networks [J]. Science. 2006 Jul 28;313(5786)504-507.
- [9] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[M]. MIT Press, 2006.
- [10] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search.[J]. Nature, 2016, 529(7587):484-489.

-
- [11] Schölkopf, B, Platt, J, Hofmann, T. Greedy Layer-Wise Training of Deep Networks[J]. Advances in Neural Information Processing Systems, 2007, 19:153-160.
- [12] 刘知青, 吴修竹. 解读 AlphaGo 背后的人工智能技术[J]. 控制理论与应用, 2016, 33(12):1685-1687.
- [13] 吕艳辉, 官瑞敏. 计算机博弈中估值算法与博弈训练的研究[J], 计算机工程, 2012, 38(11):163-166.
- [14] 徐长明. 基于连珠模式的六子棋机器博弈关键技术研究[D]. 沈阳: 东北大学, 2010.
- [15] 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.
- [16] 张小川, 陈光年, 张世强, 等. 六子棋博弈的评估函数[J], 重庆理工大学学报(自然科学版), 2010, 24(2):64-68.
- [17] 李学俊, 王小龙, 吴蕾, 等. 六子棋中基于局部“路”扫描方式的博弈树生成算法[J]. 智能系统学报, 2015(2):267-272.
- [18] 魏钦刚, 王骄, 徐心和, 等. 中国象棋计算机博弈开局库研究与设计[J]. 智能系统学报, 2007, 2(1):85-89.
- [19] 王骄, 孙英龙, 吕辉展, 等. 中国象棋中亚洲棋规循环问题的解决[J]. 东北大学学报(自然科学版), 2012, 33(4):476-481.
- [20] 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(自然科学版), 2005, 26(10):949-952.
- [21] 李淑琴, 李静波, 韩裕华, 等. 苏拉卡尔塔博弈系统中评估函数的研究[J], 北京信息科技大学学报(自然科学版), 2012, 27(6):42-45.
- [22] 张利群. 实现苏拉卡尔塔棋网络博弈平台的吃子算法[J]. 计算

-
- 机工程与应用, 2016, 52(7):62-66.
- [23] Qiu H, Zhang P, Wang Y, et al. Analysis of search algorithm in computer game of Amazons[C]// Control and Decision Conference. IEEE, 2014:3947-3950.
- [24] Quan J, Qiu H, Wang Y, et al. Application of UCT technologies for computer games of Amazon[C]// Control and Decision Conference. IEEE, 2016:6896-6899.
- [25] 郭琴琴, 李淑琴, 包华. 亚马逊棋机器博弈系统中评估函数的研究[J]. 计算机工程与应用, 2012, 48(34):50-54.
- [26] Li X, Wu L. A Multi-modal Searching Algorithm in Computer Go Based on Test[M]// Proceedings of the 2015 Chinese Intelligent Automation Conference. Springer Berlin Heidelberg, 2015:143-149.
- [27] Wang X, Li X, Wu L. The design and implementation of tibetan-go game software[C]// Chinese Control and Decision Conference. 2016:4266-4270.
- [28] 张加佳. 非完备信息机器博弈中风险及对手模型的研究[D]. 哈尔滨工业大学, 2014.
- [29] 马骁, 王轩, 王晓龙. 一类非完备信息博弈的信息模型[J]. 计算机研究与发展, 2010, 47(12):2100-2109.
- [30] Zhang J, Wang X. Building Opponent Model in Imperfect Information Board games[J]. Telkomnika Indonesian Journal of Electrical Engineering, 2014, 12(3).
- [31] 胡晓峰. 军事指挥信息系统中的机器智能:现状与趋势[J]. 人民论坛·学术前沿, 2016(15):22-34.
- [32] 朱丰, 胡晓峰. 基于深度学习的战场态势评估综述与研究展望[J]. 军事运筹与系统工程, 2016, 30(3):22-27.

-
- [33] 周超, 胡晓峰, 郑书奎,等. 战略战役兵棋演习系统兵力聚合问题研究[J]. 指挥与控制学报, 2017, 3(1):19-26.
- [34] 胡晓峰, 罗批. 兵棋推演:复杂系统管理的创新与实践[J]. 指挥控制与仿真, 2016, 38(6):1-5.
- [35] 郭圣明, 贺筱媛, 胡晓峰,等. 军用信息系统智能化的挑战与趋势[J]. 控制理论与应用, 2016, 33(12):1562-1571.
- [36] 王亚杰, 邱虹坤, 吴燕燕,等. 计算机博弈的研究与发展[J]. 智能系统学报, 2016, 11(6):788-798.
- [37] Herik V D, Jaap H, Uiterwijk, et al. Games solved: now and in the future[J]. Artificial Intelligence, 2002, 134(1):277-311.
- [38] Allis V. Searching for Solutions in Games and Artificial Intelligence[J]. Hosp Pract, 1994, 25(6).
- [39] Jonathan S, Neil B, Yngvi B, et al. Checkers is solved[J]. Science, 2007, 317(5844):1518-22.
- [40] Lieberum J. An Evaluation Function for the Game of Amazons[J]. Theoretical Computer Science, 2004, 349(2):230-244.
- [41] [Http://en.wikipedia.org/wiki/Surakarta_\(game\)](http://en.wikipedia.org/wiki/Surakarta_(game))
- [42] Fraenkel A S, Lichtenstein D. Computing a perfect strategy for $n \times n$, chess requires time exponential in n [J]. Journal of Combinatorial Theory, 1981, 31(2):199-214.
- [43] 李志敏, 罗里波, 李祥. 完全二叉树理论的计算复杂度[J]. 数学学报, 2008, 51(2):311-318.
- [44] Matsubara H, Iida H, Grimbergen R, et al. Chess, Shogi, Go, natural developments in game research[J]. ICCA Journal, 1997, 19:103-112.
- [45] Wu I C, Huang D Y, Chang H C. Connect6[J]. Icgga Journal, 2005, 28(4):235-242.

-
- [46] Elwyn Berlekamp. The dots-and boxes game: sophisticated child's play[J]. AK Peters, Ltd. 2000, 3-6.
- [47] Robson J M. N by N Checkers is Exptime Complete.[J]. Siam Journal on Computing, 2006, 13(2):252-267.
- [48] Stockmeyer L J, Chandra A K. Provably Difficult Combinatorial Games[J]. Siam Journal on Computing, 2006, 8(2):151-174.
- [49] Lichtenstein D, Sipser M. GO Is Polynomial-Space Hard[J]. Journal of the Acm, 1980, 27(2):393-401.
- [50] Reisch S. Gobang is PSPACE-complete[J]. Acta Informatica, 1980, 13(1):59-66.
- [51] Ming Y H, Tsai S C. On the fairness and complexity of generalized k -in-a-row games[M]. Elsevier Science Publishers Ltd. 2007, 385 (1): 88–100.
- [52] Iwata S, Kasai T. The Othello game on an $n \times n$ board is PSPACE-complete[J]. Theoretical Computer Science, 1994, 123(2):329-340.
- [53] Michael Sipser. Introduction to the Theory of Computation (Second Edition)[M]. China Machine Press, 2006: 265,272-289,314-323.
- [54] Hearn R A. Amazons is PSPACE-complete[J]. Computer Science, arXiv:cs.CC/0502013v1, 2005.
- [55] Herik H J V D, Uiterwijk J W H M, Rijswijck J V. Games solved: Now and in the future[J]. Artificial Intelligence, 2002, 134(1-2):277-311.
- [56] Fuller S H, Gaschnig J G, Gillogly J J. Analysis of the Alpha-Beta Pruning Algorithm[J]. Analysis of the Alpha-Beta Pruning Algorithm, 1973.

-
- [57] Lu H, Xia Z Y. AWT: Aspiration with Timer Search Algorithm in Siguó[C]// International Conference on Computers and Games. Springer-Verlag, 2008:264-274.
- [58] Zou J. Chinese Chess Algorithm Design and Optimize Based on MTD(f)[J]. Computer & Digital Engineering, 2008.
- [59] Leifker D B, Kanal L N. A Hybrid SSS*/Alpha-Beta Algorithm for Parallel Search of Game Trees.[C]// IJCAI. 1985:1044-1046.
- [60] Kaindl H, Shams R, Horacek H. Minimax Search Algorithms With and Without Aspiration Windows[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1992, 13(12):1225-1235.
- [61] 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(06):1-7.
- [62] 焦尚彬, 刘丁. 博弈树置换表启发式算法研究[J]. 计算机工程与应用, 2010, 46(06):42-45.
- [63] Donkers H H L M, Uiterwijk J W H M, Herik H J V D. Probabilistic opponent-model search[J]. Information Sciences, 2001, 135(3-4):123-149.
- [64] Schaeffer J. The History Heuristic and Alpha-Beta Search Enhancements in Practice[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1989, 11(11):1203-1212.
- [65] Sakuta M, Hashimoto T, Nagashima J, et al. Application of the killer-tree heuristic and the lambda-search method to lines of action[J]. Information Sciences—informatics & Computer Science An International Journal, 2003, 154(3-4):141-155.
- [66] Reinefeld A, Marsland T A. Enhanced iterative-deepening search[J]. Pattern Analysis & Machine Intelligence IEEE

-
- Transactions on, 1994, 16(7):701-710.
- [67] Marsland T A, Reinefeld A, Schaeffer J. Low overhead alternatives to SSS*[J]. Artificial Intelligence, 1987, 31(2):185-199.
 - [68] Plaat A, Schaeffer J, Pijls W, et al. SSS* = Alpha-Beta + TT[J]. Computer Science, 2014, 8(1):25.
 - [69] Berliner H. The B* tree search algorithm: A best-first proof procedure[J]. Artificial Intelligence, 1979, 12(1):23-40.
 - [70] Berliner H J, Mcconnell C. B * probability based search [J]. Artificial Intelligence, 1996, 86(1):97-156.
 - [71] Lorentz R. Using Evaluation Functions in Monte-Carlo Tree Search[J]. Theoretical Computer Science, 2016.
 - [72] Guez A, Silver D, Dayan P. Scalable and efficient bayes-adaptive reinforcement learning based on monte-carlo tree search[J]. Journal of Artificial Intelligence Research, 2013, 48(1):841-883.
 - [73] Browne C B, Powley E, Whitehouse D, et al. A Survey of Monte Carlo Tree Search Methods[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2012, 4:1(1):1-43.
 - [74] Baier H, Winands M H M. Time Management for Monte-Carlo Tree Search in Go[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2012, 7168:39-51.
 - [75] Raiko T, Peltonen J. Application of uct search to the connection games of hex, y,* star, and renkula![J]. AI and Machine Consciousness, 2008.
 - [76] Gelly S, Silver D. Combining online and offline knowledge in UCT[C]//Proceedings of the 24th international conference on Machine learning. ACM 2007: 273-280.

-
- [77] Zhang J, Wang X, Lin J, et al. UCT Algorithm in Imperfect Information Multi-Player Military Chess Game[C]//Proceedings of Joint Conference on Information Sciences. 2008:1-9.
- [78] Raiko T, Peltonen J. Application of uct search to the connection games of hex, y,* star, and renkula![J]. AI and Machine Consciousness, 2008.
- [79] Kato H, Takaya M, Yamamura A. Analysis of a Monte Carlo Tree Search in Knight- Amazons ☆ [J]. Procedia Computer Science, 2015, 62:31-38.
- [80] Hashimoto J, Kishimoto A, Yoshizoe K, et al. Accelerated UCT and Its Application to Two-Player Games[J]. Lecture Notes in Computer Science, 2012, 7168:1-12.
- [81] 陈国良, 孙广中, 徐云,等. 并行计算的一体化研究现状与发展趋势[J]. 科学通报, 2009(8):1043-1049.
- [82] 陈国良, 毛睿, 蔡晔. 高性能计算及其相关新兴技术[J]. 深圳大学学报(理工版), 2015(1):25-31.
- [83] 李之棠, 陈华民. 博弈树并行搜索算法[J]. 小型微型计算机系统, 1998(10):53-56.
- [84] 计永昶, 丁卫群, 陈国良,等. 一种实用的并行计算模型[J]. 计算机学报, 2001, 24(4):437-441.
- [85] 卢风顺, 宋君强, 银福康,等. CPU/GPU 协同并行计算研究综述 [J]. 计算机科学, 2011, 38(3):5-9.
- [86] 聂慧, 彭娇, 金晶,等. 基于多核系统的并行线性 RankSVM 算法[J]. 计算机应用研究, 2017, 34(1):46-51.
- [87] 曹婷婷. 基于多核处理器串行程序并行化改造和性能优化[D]. 西南交通大学, 2009.
- [88] 杨冰. 多 CPU 多核系统上的并行预处理算法[D]. 中国工程物

-
- 理研究院, 2014.
- [89] 王京辉, 乔卫民. 基于 PVM 的博弈树的网络并行搜索[J]. 计算机工程, 2005, 31(9):29-30.
- [90] 吉根林. 遗传算法研究综述[J]. 计算机应用与软件, 2004, 21(2):69-73.
- [91] 王书宇, 李龙澍, 汪群山. 改进的协同进化遗传算法在机器博弈中的应用[J]. 计算机技术与发展, 2008, 18(12):4-7.
- [92] Liaw C, Wang W H, Tsai C T, et al. Evolving a team in a First-Person Shooter game by using a genetic algorithm[J]. Applied Artificial Intelligence, 2013, 27(3):199-212.
- [93] A.I. EsparciaAlcázar, A. MartínezGarcía, A. Mora, et al. Controlling bots in a First Person Shooter game using genetic algorithms[J]. Evolutionary Computation:cec, 2010, 1210(1):1 - 8.
- [94] Kim H T, Kim K J. Hybrid of Rule-based Systems Using Genetic Algorithm to Improve Platform Game Performance[J]. Procedia Computer Science, 2013, 24:114-120.
- [95] Amos M, Coldridge J. A genetic algorithm for the Zen Puzzle Garden game[J]. Natural Computing, 2012, 11(3):353-359.
- [96] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6):82-97.
- [97] Salakhutdinov R, Hinton G. An efficient learning procedure for deep Boltzmann machines[J]. Neural Computation, 2012, 24(8):1967.
- [98] Hinton G E. Technical Perspective: A Better Way to Learn

-
- Features[J]. Communications of the Acm, 2011, 54(10):94.
- [99] Hinton G E. Training products of experts by minimizing contrastive divergence[J]. Neural Computation, 2014, 14(8):1771-1800.
- [100] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436-444.
- [101] G. E. Hinton, R. R. Salakhutdinov. Reducing the dimensionality of data with neural network[J]. Science, 2006, 313(5786): 504-507.
- [102] Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural Networks, 2015, 61:85.
- [103] 吕艳辉, 官瑞敏. 计算机博弈中估值算法与博弈训练的研究[J]. 计算机工程, 2012, 38(11):163-166.
- [104] 苏攀, 王熙照, 李艳. 基于不平衡学习的分类器博弈模型及其在中国象棋中的应用 [J]. 计算机研究与发展, 2011, 48(5):841-847.
- [105] A. N. Meltzoff, P. K. Kuhl, J. Movellan, T. J. Sejnowski. Foundations for a new science of learning. Science, 2009, 325(5938): 284-288.
- [106] T. Mitchell. Machine Learning, New York: McGraw-Hill, 1997.
- [107] Ryszard S. Michalski, Ivan Bratko. 机器学习与数据挖掘[M]. 电子工业出版社, 2004.
- [108] T. G. Dietterich. Machine learning research: Four current directions. AI Magazine, 1997, 18(4), 97-136.
- [109] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[J]. IEEE Transactions on Neural Networks, 2005, 16(1):285-286.
- [110] 郭潇道, 李程, 梅俏竹. 深度学习在游戏中的应用[J]. 自动化

-
- 学报, 2016, 42(5):676-684.
- [111] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540):529.
- [112] Zeng F, Mei X, Liu J, et al. Research on estimate system of computer game decision algorithm[C]// Control and Decision Conference. IEEE, 2016:6892-6895.
- [113] 梅险, 孙德强, 杜世锦,等. 非完备信息博弈游戏的实践研究 [C]// 中国信息通信研究新进展论文集. 2014.
- [114] Mei X, Zhang D, Liao Q, et al. Research on the platform system of military chess computer game[C]// Control and Decision Conference. IEEE, 2013:1754-1757.
- [115] Mei X, Zhang D, Sun D, et al. The Computer Game Model of Dou Dizhu[C]// Control and Decision Conference. IEEE, 2014:3951-3954.
- [116] Qiu H, Gao F, Wang Y, et al. Implementation of Common Platform for Two-Person Game[C]// Control and Decision Conference. IEEE, 2013:929-932.
- [117] Yin W, Qiu H, Wang Y, et al. Application of referee-player role model to the platform of computer games[C]// Control and Decision Conference. IEEE, 2015:5799-5802.
- [118] Quan J, Qiu H, Cao Y, et al. Research of test platform on computer games with distributed system[C]// Control and Decision Conference. IEEE, 2016:5032-5035.
- [119] Wu I-Chen, Huang Dei-Yen. A New Family of k-in-a-row Games [C]. In Proceedings of The 11th Advances in Computer Games Conference, 2005: 88-100.

-
- [120] Wágner J, Virag I. Solving Renju [J]. ICGA Journal, 2001, 24(1): 30-34.
- [121] Allis L V, Herik H.J van den, Huntjens M P H. Go-Moku Solved by New Search Techniques [J]. Journal of Computational Intelligence, 1995, 12(1): 7-24.
- [122] Neumann J V. Zur Theorie der Gesellschaftsspiele[J]. Mathematische Annalen, 1928, 100(1):295-320.
- [123] Berlekamp E. Introductory overview of mathematical Go endgames[J]. Proceedings of Symposia in Applied Mathematics, 1991,43: 73-100.
- [124] Mller M, Zrich I E. Game Theories and Computer Go[C]// In Processing of the Go and Computer Science Workshop (GCSW'93), INRIA, 1993.
- [125] Kocsis L, Szepesvári C. Bandit Based Monte-Carlo Planning[C]// European Conference on Machine Learning. Springer-Verlag, 2006:282-293.
- [126] Auer P, Cesa-Bianchi N, Fischer P. Finite-time Analysis of the Multiarmed Bandit Problem[J]. Machine Learning, 2002, 47(2-3):235-256.
- [127] Wang Y, Gelly S. Modifications of UCT and sequence-like simulations for Monte-Carlo Go[C]// Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on. IEEE, 2007:175-182.
- [128] Enzenberger M, Muller M, Arneson B, et al. Fuego—An Open-Source Framework for Board Games and Go Engine Based on Monte Carlo Tree Search[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2010, 2(4):259-270.

-
- [129] Baudiš P, Gailly J L. PACHI: State of the Art Open Source Go Program[C]// Advances in Computer Games. Springer Berlin Heidelberg, 2011:24-38.
- [130] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587):484-489.
- [131] Gelly S, Silver D. Achieving Master Level Play in 9 x 9 Computer Go[C]// AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, Usa, July. DBLP, 2008:1537-1540.
- [132] Banerjee B, Stone P. General game learning using knowledge transfer[C]// International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 2007:672-677.
- [133] Gelly S, Silver D. Combining online and offline knowledge in UCT[C]// International Conference on Machine Learning. ACM, 2007:273-280.
- [134] Guillaume M. Chaslot, Mark H.M. Winands and H. Jaap van den Herik. Parallel Monte-Carlo Tree Search[C]// Proceedings of the 6th International Conference on Computers and Games. Springer-Verlag, 2008:60-71.
- [135] Gelly S, Hoock J B, Rimmel A, et al. The parallelization of Monte-Carlo planning[J]. Eprint Arxiv, 2008, 26(4):282--293.
- [136] Wang J, Li S, Chen J, et al. 4*4-Pattern and Bayesian Learning in Monte-Carlo Go[M]// Advances in Computer Games. 2011:108-120.
- [137] Wang J, Xiao C, Zhu T, et al. Only-One-Victor Pattern Learning in Computer Go[J]. IEEE Transactions on Computational Intelligence & Ai in Games, 2017, 9(1):88-102.

-
- [138] Barker J K, Korf R E. Solving dots-and-boxes[C]// Twenty-Sixth AAAI Conference on Artificial Intelligence. AAAI Press, 2012:414-419.
- [139] 连莲, 徐心和, 张雪峰,等. 点点连格棋机器博弈系统关键技术分析[C]// 中国人工智能学会学术年会. 2009:p719-724.
- [140] Yu Yang, Shu-qinLi, YanZhao. Research on the chessboard representations of Dots-and-Boxes[C]// The 3rd International Conference on Information Technology and Computer Science (ITCS2011). 2011.7
- [141] Li D, Li S, Bao H. Research and Implementation of Dots-and-Boxes Chessboard Based on the Game Strings-and-Coins[M]// Advanced Research on Computer Education, Simulation and Modeling. Springer Berlin Heidelberg, 2011:14-20.
- [142] Zhuang Y, Li S, Peters T V, et al. [C]//Computational Intelligence and Games (CIG), 2015 IEEE Conference on. IEEE, 2015: 314-321.
- [143] Kocsis L, Szepesvári C. Bandit based monte-carlo planning[C]//European conference on machine learning. Springer Berlin Heidelberg, 2006: 282-293.
- [144] Moravčík M, Schmid M, Burch N, et al. DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker[J]. arXiv preprint arXiv:1701.01724, 2017.
- [145] Wikipedia. EinStein würfelt nicht! [EB/OL].
[https://en.wikipedia.org/wiki/EinStein würfelt nicht!_w%C3%BCrfelt_nicht!](https://en.wikipedia.org/wiki/EinStein_würfelt_nicht!_w%C3%BCrfelt_nicht!).
- [146] Li X, Yang G, Wu L, et al. An offensive and defensive expect

-
- minimax algorithm in EinStein Würfelt Nicht![C]// Control and Decision Conference. IEEE, 2015:5814-5817.
- [147] 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.
- [148] R.coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search.in Proceedings of the 5th International Conference on Computers and Games,ser.CG'06.Berlin, Heidelberg:Springer-Verlag,May 2006,4630:72-83
- [149] 和力, 吴丽贤. 斜方棋机器博弈程序的设计与实现[J]. 现代计算机:专业版, 2012(17):47-50.
- [150] 王骥, 孙建伶. 基于优化迭代的博弈树算法[J]. 计算机应用与软件, 2008, 25(2):228-230.
- [151] Wang X, Wang Y, Liu T, Research of EinStein würfelt nicht! würfelt nicht! Evaluation Algorithm. In: Cao H, Zhu X eds 25th Chinese Control and Decision Conference. Singapore: IEEE, 2013. 1751-1753.
- [152] Wang X, Zhang H, Wang Y. Research of Einstein würfelt nicht! Monte Carlo Algorithm Based on Optimized Evaluation.Control and Decision Conference. 2016 :5054-5057.
- [153] Blair J R S, Mutchler D, Liu C. Heuristic Search in One-Player Games With Hidden Information[M]. University of Tennessee, 1992.
- [154] Iolis B, Bontempi G. Comparison of Selection Strategies in Monte-Carlo Tree Search for Computer Poker[J]. 2010:1-4.
- [155] Broeck G, Driessens K, Ramon J. Monte-Carlo Tree Search in Poker Using Expected Reward Distributions[C]// Asian Conference on Machine Learning: Advances in Machine

-
- Learning. Springer-Verlag, 2009:367-381.
- [156] Filo L, Filipe S, Reis L, et al. Building a no limit texas hold'em poker agent based on game logs using supervised learning[C]// International Conference on Autonomous and Intelligent Systems. Springer-Verlag, 2011:73-82.
- [157] Moravčík M, Schmid M, Burch N, et al. DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker[J]. Science, 2017, 356(6337):508.
- [158] Zinkevich M, Johanson M, Bowling M H, et al. Regret Minimization in Games with Incomplete Information.[C]// Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December. DBLP, 2007:1729-1736.
- [159] Bowling M, Burch N, Johanson M, et al. Computer science. Heads-up limit hold'em poker is solved.[J]. Science, 2015, 347(6218):145-9.
- [160] Johanson M. Measuring the Size of Large No-Limit Poker Games[J]. Computer Science, 2013.
- [161] Burch N, Johanson M, Bowling M. Solving imperfect information games using decomposition[C]// Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI Press, 2014:602-608.
- [162] Schnizlein D, Bowling M, Szafron D. Probabilistic state translation in extensive games with large action sets[C]// Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence. 2009:278-284.
- [163] Gilpin A, Sandholm T, Sørensen T B. A heads-up no-limit Texas

-
- hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs[C]// International Joint Conference on Autonomous Agents and Multiagent Systems. DBLP, 2008:911-918.
- [164] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.
- [165] Cooper G F. The computational complexity of probabilistic inference using bayesian belief networks[J]. Artificial Intelligence, 1990, 42(2):393-405.
- [166] Pearl J. Fusion, propagation, and structuring in belief networks [J]. Artificial Intelligence, 1986, 29(3):241-288.
- [167] Beal D F. A generalised quiescence search algorithm [J]. Artificial Intelligence, 1990, 43(1):85-98.
- [168] 程克非, 张聪, 沈一栋. 计算机桥牌双明手解的 Hash 表改进[J]. 重庆大学学报:自然科学版, 2005, 28(12):45-47.
- [169] 张志刚. 双明手桥牌打牌辅助分析程序研究[J]. 现代计算机, 2015(23):47-50.