



中国人工智能学会
Chinese Association for Artificial Intelligence

中国人工智能学会系列白皮书 ——棋盘上的人工智能

中国人工智能学会
二〇二五年十一月



中国人工智能学会系列白皮书 ——棋盘上的人工智能

中国人工智能学会
二〇二五年十一月

《中国人工智能学会系列白皮书》编委会

主 任：戴琼海

执行主任：马华东

副 主 任：赵春江 何 友 王恩东 郑庆华 刘成林

周志华 孙富春 庄越挺 胡德文 杜军平

杨 强

委 员：陈松灿 董振江 付宜利 高新波 公茂果

古天龙 何 清 胡清华 黄河燕 季向阳

蒋田仔 林浩哲 梁吉业 刘奕群 潘 纲

石光明 孙茂松 孙长银 陶建华 王海峰

王熙照 王 轩 王蕴红 吴 飞 于 剑

余有成 张化光 张学工 章 毅 周鸿祎

周 杰 祝烈煌

本书编写组

张小川 徐 勇 王亚杰 李学俊 邱虹坤 李霞丽

周 珂 王静文 丁 濛 徐心和 李淑琴 杨放春

王 骄 徐长明 吴 愚 梅 险 高 强 常晓亚

全书统稿：张小川 徐 勇 常晓亚

前 言

计算机博弈是人工智能领域的重要应用，它以高对抗性的棋牌类游戏项目为研究对象，具有怡神益智、评判客观、挑战无穷的特点。近年来，随着人工智能、大模型等技术的飞速发展，计算机博弈模型能够自主学习复杂的策略和技能、处理更加复杂的博弈任务，成为衡量 AI 智能水平的重要领域之一。从棋类博弈到电子游戏，机器博弈不仅是技术进步的展示窗口，更是人类智慧与机器智能交互融合的舞台。未来，计算机博弈领域将继续快速发展，技术的融合和创新将推动该领域达到新的高度。

本书编写的指导思想是：通识技术和专项技术相结合，依托具体的棋牌类项目，介绍国内外相关技术的应用方法与进展。本书主要介绍了机器博弈的发展历程、国内外研究现状、复杂度分析、机器博弈的典型技术、平台技术，完备信息博弈和非完备信息博弈的关键技术、以及国内外赛事等，提出了机器博弈未来的发展趋势。期望该书的出版能促进全社会对机器博弈、以及人工智能的整体认知和应用水平。

本书的编写者全部是机器博弈领域资深的高校教师，而且都在不同时期指导学生参加全国计算机博弈大赛获重要奖项，具有较强的理论水平和实践经验。编写分工如下：第 1 章由东北大学徐心和、安徽财经大学徐勇编写；第 2 章和第 4 章由沈阳航空航天大学邱虹坤编写；第 3 章由沈阳大学高强编写；第 5 章由哈尔滨理工大学梅险编写；第 6.1 节由东北大学秦皇岛分校徐长明编写；第 6.2 和 7.1 节由东北大学

王骄编写；第 6.3 节由北京信息科技大学李淑琴编写；第 6.4 节由安徽大学李学俊编写；第 6.5 至 6.8 节、第 7.4 至 7.5 节、第 8 章由安徽财经大学徐勇编写；第 6.9 节由中央民族大学李霞丽编写；第 6.10 节由北京信息科技大学丁濛编写；第 6.11 节由北京科技大学周珂编写；第 6.12 节由安徽财经大学张炜编写；第 6.13 节由沈阳工业大学王静文编写；第 7.2 节由重庆三峡学院吴愚编写；第 7.3 节由北京邮电大学杨放春编写；第 9 章和第 10 章由沈阳航空航天大学王亚杰和安徽财经大学徐勇编写。全书统稿由张小川和徐勇完成。

本书在编写过程中得到很多专家的支持和指导，在此表示诚挚的感谢！书中难免有错误和有妥之处，恳请读者批评指正。

作 者

2024 年 11 月

目 录

中国机器博弈 2024 发展报告	1
第 1 章 引言	1
第 2 章 机器博弈的发展状况	8
2.1 机器博弈历史	8
2.2 机器博弈研究现状	13
2.2.1 国外的研究现状	13
2.2.2 国内的研究现状	20
2.3 机器博弈产业现状	26
2.4 面临的问题与展望	28
第 3 章 博弈的复杂度	37
3.1 概述	37
3.2 博弈问题的状态复杂度及其估算方法	39
3.2.1 博弈问题的状态复杂度定义	39
3.2.2 博弈问题的博弈树复杂度	42
3.3 博弈问题的计算复杂性	47
第 4 章 机器博弈的典型技术	52
4.1 概述	52
4.2 穷尽搜索	52
4.3 裁剪搜索	53
4.4 启发式算法	54
4.5 迭代深化	56
4.6 最佳优先算法	56
4.7 随机搜索算法	57
4.8 并行计算	60
4.9 遗传算法	61

4.10 路径寻迹算法	62
4.11 神经网络	64
4.12 机器学习	65
4.13 知识库	67
4.14 微分博弈	71
4.15 贝叶斯网络与概率推理	72
第 5 章 机器博弈的平台技术	76
5.1 计算机博弈平台概述	76
5.2 计算机博弈平台的分类	77
1.完备信息博弈平台和非完备信息博弈平台	77
2.单引擎博弈平台和多引擎博弈平台	77
3.单机博弈平台和网络博弈平台	77
4.程序级博弈平台和模块级博弈平台。	77
5.3 计算机博弈平台的设计规范	78
5.3.1 人机交互接口规范	78
1.显示输出规范	78
2.操作输入规范	78
3.其它约定规范	78
5.3.2 平台与引擎通信的协议规范	78
1.对局状态和过程的信息编码	78
2.引擎与博弈平台的通信协议	79
3.引擎示例代码	79
5.4 计算机博弈平台的相关技术	79
1.匿名管道通信技术	79
2.数据保存和加密技术	79
3.在线对局技术	80
5.5 计算机博弈平台应用实例	80

1. 哈尔滨理工大学军棋博弈平台系统	80
2. 哈尔滨理工大学二打一（斗地主）博弈平台系统	80
3. 北京理工大学苏拉卡尔塔棋博弈平台系统	81
4. 沈阳航空航天大学通用计算机博弈对战平台	81
第 6 章 完备信息机器博弈的专项技术	82
6.1 六子棋机器博弈	82
6.1.1 概述	82
6.1.2 六子棋机器博弈主要技术	83
6.2 围棋机器博弈	86
6.2.1 概述	86
6.2.2 围棋机器博弈主要方法	87
6.3 点格棋机器博弈	91
6.3.1 概述	91
6.3.2 点格棋机器博弈主要技术	91
6.4 爱恩斯坦棋机器博弈	94
6.4.1 概述	94
6.4.2 爱恩斯坦棋机器博弈主要技术	96
6.5 苏拉卡尔塔棋机器博弈	100
6.5.1 概述	100
6.5.2 苏拉卡尔塔棋规则	100
6.5.3 搜索算法	101
6.5.4 评估函数	103
6.6 海克斯棋机器博弈	103
6.6.1 概述	103
6.6.2 海克斯棋规则	104
6.6.3 算法改进	105
6.7 五子棋机器博弈	106

6.7.1 概述	106
6.7.2 五子棋规则	106
6.7.3 改进的五子棋博弈算法	108
6.7.4 机器人五子棋机器博弈	108
6.8 中国象棋机器博弈	110
6.8.1 概述	110
6.8.2 估值方法	111
6.8.3 搜索算法	111
6.9 藏族久棋博弈	112
6.9.1 藏族久棋简介	112
6.9.2 藏族久棋博弈方法	115
6.9.3 藏族久棋线上对弈平台	118
6.10 国际跳棋	120
6.10.1 概述	120
6.10.2 基本规则简介	120
6.10.3 国际跳棋机器博弈关键技术	121
6.11 不围棋	123
6.11.1 概述	123
6.11.2 不围棋机器博弈主要技术	123
6.12 幻影围棋	126
6.12.1 概述	126
6.12.2 幻影围棋博弈主要技术	127
6.13 亚马逊棋	129
6.13.1 概述	129
6.13.2 游戏规则	132
6.13.3 亚马逊棋博弈主要技术	133
第 7 章 非完备信息机器博弈的专项技术	137

7.1 德州扑克机器博弈	138
7.1.1 概述	138
7.1.2 非完备信息动态博弈解的主要方法	139
7.2 军棋机器博弈	142
7.2.1 概述	142
7.2.2 军棋机器博弈主要技术	143
1. 对手棋子可能性矩阵更新	143
2. 欺诈走法的选择和判定	146
3. 搜索技术的选择	146
7.3 桥牌机器博弈	147
7.3.1 概述	147
7.3.2 桥牌机器博弈的主要方法	149
7.4 国标（大众）麻将机器博弈	152
7.4.1 概述	152
7.4.2 国标（大众）麻将机器博弈规则	153
7.4.3 麻将博弈平台	155
7.5 二打一扑克牌机器博弈	156
7.5.1 概述	156
7.5.2 计分规则	156
7.5.3 二打一扑克牌机器博弈的特点	157
第 8 章 机器博弈的国内外赛事	158
8.1 国际机器博弈赛事	158
8.1.1 国际象棋人机博弈大赛	158
8.1.2 围棋人机与机机博弈大赛	160
8.1.3 桥牌计算机博弈大赛	162
8.1.4 德州扑克人机与机机博弈大赛	163
8.1.5 其它赛事与研究成果	164

8.2 国内机器博弈赛事	165
8.2.1 中国象棋人机与机机博弈大赛	165
8.2.2 中国计算机博弈大赛	166
第9章 结束语	169
参考文献	173

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。随着计算机博弈事业的发展，国内外学术界、工业界陆续涌现出多个计算机博弈相关的学术组织、企业，开展了一系列的计算机博弈相关研究与实践活动。如国际机器博弈协会（International Computer Games Association, ICGA 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。2024 年 10 月 19 日至 24 日，ICGA 在西班牙的圣地亚哥-德孔波斯特拉举办世界计算机国际象棋锦标赛，这也是该锦标赛的 50 周年纪念活动，该活动由欧洲人工智能会议主办，谷歌 DeepMind 赞助。2024 计算机和游戏大会将于 2024 年 11 月 26 日至 28 日在线举行。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。国内负责组织棋牌类博弈竞赛和学术交流的群众组织是中国人工智能学会下属的机器博弈专业委员会。由中国人工智能学会主办，浙江省机器人创新中心、重庆理工大学联合承办的 2024 年第三届高校计算机博弈教育论坛于 2024 年 4 月 20 日在浙江省宁波市余姚机器人小镇举办。2024 年 8 月 2 日下午，由中国人工智能学会（CAAI）主办，CAAI 机器博弈专委会组织、青海民族大学和重庆理工大学联合承办的“2024 中国机器博弈学术会议”，以线下线上相结合的方式

在青海民族大学博雅校区举行。

机器博弈的发展历程大致可以分为萌芽阶段，发展阶段和深度学习阶段。萌芽阶段，1928 年，冯·诺依曼提出了极大极小值定理，并在此基础上证明了博弈论的基本原理。1950 年，图灵基于几个最基本的国际象棋规则，运用简单的逻辑搜索两个步骤，完成了简单的计算机智能程序。1956 年，麦卡锡发明了 Alpha-Beta 搜索算法，两年后基于此算法的国际象棋程序问世，基本可以达到初学者的技术水平。发展阶段，随着研究的不断发展与深入，研究人员发现博弈程序的优劣很大程度上是由搜索的深度来决定的，所以人们开始致力于更加有效的搜索算法的研究。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，这一胜利使得计算机博弈的发展迎来了一个新的高峰。深度学习阶段，2015 年，著名的阿尔法狗（AlphaGo）诞生，并在 2016 年打败了世界围棋冠军李世石，这一事件也把计算机博弈的研究热潮推向了新的高度。2019 年，在暴雪娱乐公司打造的实时策略游戏《星际争霸 II》中，人工智能程序 AlphaStar 打败了职业玩家。2024 年诺贝尔物理学奖在瑞典皇家科学院揭晓，今年该奖项授予 2 位人工智能方面的专家：普林斯顿大学的 John J. Hopfield 和多伦多大学的 Geoffrey E. Hinton，他们“因在利用人工神经网络实现机器学习方面的基础性发现和发明而获奖”。在可以预见的未来，人工智能将改变人类的生产和生活方式。

中国机器博弈研究与应用虽然起步较晚，但发展迅速。作为中国机器博弈事业开拓者，东北大学徐心和教授自 2003 年便开始从事中

国象棋的计算机博弈的研究工作；并于 2005 年发起组织和领导了中国人工智能学会机器博弈专业委员会的成立，将国际象棋算法移植到中国象棋电脑程序中，并取得了显著成果。专委会的成立标志着中国在机器博弈领域的研究开始系统化和专业化。吸引了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。2008 年，专委会在北京举办了 ICGA 国际计算机博弈大赛，有 14 个国家的 71 支代表队参赛，这是中国在国际计算机博弈领域的重要展示。2010 年，专委会与教育部高等学校计算机类专业教学指导委员会联合主办了首届全国大学生计算机博弈大赛暨全国锦标赛，吸引了 18 所高校的 53 支代表队参赛，这标志着计算机博弈在中国高校中的普及和推广，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。2015 年，计算机

博弈被国家体育总局棋牌运动管理中心列入第三届全国智力运动会项目，进一步扩大了比赛的影响力和知名度。自 2006 年起，专委会每年都举办全国范围内的大型计算机博弈活动，取得了较大的社会影响。中国机器博弈竞赛项目从中国象棋 1 个项目，发展到 19 个比赛项目；参与高校从不足 10 所，发展到 70 多所高校 5000 余师生参与。特别需要提到的是，2024 年中国大学生计算机博弈大赛中增加了面向高校学生的机器人五子棋、坦克大战 2 个软硬件结合的博弈竞赛项目，标志着国内高校机器博弈活动向软硬件融合发展、产业化发展趋势的新探索。

盘点中国计算机博弈事业的发展历程，那最值得一提的学者还有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。

十余年来，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。形式上看，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，我们专委会主办了中国计算机博弈技术论坛、在中国控制与决策学术年会（CCDC）上开辟了“机器博弈”特邀专题论坛、在《重庆理工大学学报》（自科版）上开辟了“机器博弈”专栏，开展成果交流，提高我国在该领域的学术水平和学术影响。同时，专委会还组织了中国大学生计算机博弈大赛、中国计算机博弈锦标赛和中国大学生机器博弈创新设计赛等赛事，这些赛事不仅为大学生提供了实战平台，也推动了人工智能教育和研究的发展；与竞技世界（北京）网络技术有限公司、学大教育、浙江钱塘机器人及智能装备研究有限公司、重庆工能科技有限公司等企业开展校企合作。通过这些活动和平台，中国人工智能学会机器博弈专委会为机器博弈领域的研究人员、开发者和学生提供了丰富的资源和机会，推动了机器博弈研究的进步与应用的发展。

机器博弈的产业化是一个涉及人工智能、机器学习、博弈论等多个领域的复杂过程。AlphaGo 的成功，标志着人工智能进入了新的阶段，得益于深度学习、强化学习等人工智能技术的进步，机器博弈已经不再局限于游戏领域，而是开始向更广泛的行业渗透，如在数字经济、智慧医疗、智能交通、航空、航天、兵棋推演和战略、战役和战术博弈中等领域，机器博弈技术正在推动相关产业的发展。层出不穷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，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。因此，机器博弈的产业化是一个多元化、跨学科的过程，它不仅推动了人工智能技术的发展，

也为多个行业带来了创新和变革。随着技术的不断进步和应用场景的拓展，机器博弈的产业化前景非常广阔。

本《中国机器博弈 2024 发展报告》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，博弈棋种和比赛平台，然后结合相关棋种介绍了各种博弈技术，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，当然还包括目前流行的深度学习算法、最新的桥牌和德州扑克博弈算法等。

让计算机博弈活动在更多的学校中生根、开花、结果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。

近代机器博弈的研究，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机器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，这是机器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。

经过多年对机器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，机器博弈水平迅速上升。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2018年至2024年期间，国外机器博弈领域的研究显著推进，主要体现在理论模型创新、算法优化、应用场景拓展以及实验评估体系的完善上。2018年OpenAI Five在Dota2中击败人类队伍，OpenAI Five是OpenAI开发的Dota2 AI，在2018年TI8（国际邀请赛）的表演赛中击败了职业玩家队伍，展示了AI在复杂的多人对战游戏中的能力。DeepMind推出的AlphaZero在没有人类数据的情况下，通过自我对弈学习了国际象棋、将棋和围棋，并在短时间内超越了之前所有版本的AlphaGo，展示了AI自主学习的能力。2019年，Pluribus在德州扑克多人对战中获胜，AlphaStar在星际争霸II中表现卓越。2020年，大规模在线博弈研究展开，AI与全球棋手进行在线对战。2021年，AI辅助决策系统在商业博弈中得到应用，新一代AI在国际象棋中取得新突破。2022年，AI在电子竞技中广泛应用，对抗样本研究结合博弈论和机器学习。2023年，探索AI与人类协作的新模式，AI在国际关系模拟中作为决策辅助工具。2024年，AI伦理和博弈论的研究增多，AI在元宇宙中的博弈实验开始探索新的策略和行为模式。这些事件展示了机器博弈在多领域的广泛影响和应用潜力。

在中国国内，也有一些学者从事机器博弈方面的研究。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。

从 2006 年起，由东北大学徐心和教授发起成立了中国人工智能学会机器博弈专业委员会，在国内高校师生间推广与普及机器博弈知识，每年举办一届机器博弈大赛暨中国机器博弈锦标赛(王骄，孙英龙，吕辉展，& 杨辉，2012)。在沈阳航空航天大学王亚杰教授、重庆理工大学张小川教授、安徽大学李学俊教授等人的大力推动下，陆续在辽宁省和安徽省开展省级大学生机器博弈竞赛，吸引越来越多高校师生参与机器博弈相关研究中来，中国机器博弈进入了快速发展阶段(李飞，王亚杰，尹航，& 孙玉霞，2016)。

直到 2017 年，国内许多企业如腾讯、百度、联众、新睿、竞技世界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。

在 2018 至 2024 年期间，中国机器博弈领域经历了显著的发展。以 2018 年“天河二号”超级计算机的全球最快评选为起点，该领域的研究逐渐深入，不断融合深度学习、强化学习等先进技术，提升了机器博弈系统的智能决策能力。随后几年，全国范围内举办的机器博弈赛事进一步推动了技术交流与创新，并激发了年轻人的兴趣。2020 年后，出现了许多新的机器博弈技术，提升了机器博弈智能体的决策效率和准确性。到 2022 年，这些技术开始广泛应用于游戏、智能交通等实际场景。至 2024 年，中国机器博弈领域已展现出强大的创新活力和广阔的应用前景，预示着未来该领域的持续快速发展。此外，一些研究开始探索将机器博弈技术应用于社会治理领域，例如，通过模拟博弈过程来预测和应对社会冲突、制定更有效的政策等。至 2025 年，机器博弈研究呈现出从专用游戏算法向通用战略智能的范式转移。前沿探索集中于将大型语言模型的核心推理能力与机器博弈的自我对弈范式深度融合，旨在培养能适应复杂合作与竞争环境的通用决策智能体。研究表明，通过多轮战略游戏中进行的强化学习训练，智能体所获得的战略能力可有效泛化至未知任务乃至通用推理领域。这一进展标志着机器博弈的价值超越了传统的游戏胜负，正成为锻造

具备高级决策与协作能力的通用人工智能的关键实验平台与核心驱动力量。基于此类技术构建的智能决策支持系统，已在模拟推演、策略评估等实际场景中展现出指导价值。

目前，在机器博弈领域中，非完备算法扮演着至关重要的角色，它使得机器能够在信息不完整或不确定的环境中进行有效的决策，这对于提高机器在复杂博弈场景中的性能和适应性具有决定性的影响。它们在处理不完全信息或不确定性方面具有显著优势。这些算法在多个领域展现出其有效性。

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”。而机器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

在机器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，专家和学者们在机器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，产生了许多机器博弈技术，如极大极小搜索、负极大值搜索、Alpha-Beta 剪枝、并行搜索算法等 (Bench-Capon & Dunne, 2007; Cormen, Rivest, Leiserson, & C., 2009; Lieberum, 2005; T. A

Marsland & Campbell, 1982; Takeuchi, Kaneko, & Yamaguchi, 2010)。

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 (G. E. Hinton, Osindero, & Teh, 2006)，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 (Schölkopf & Platt, 2007; David Silver et al., 2016) 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机器博弈及相关领域中，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。

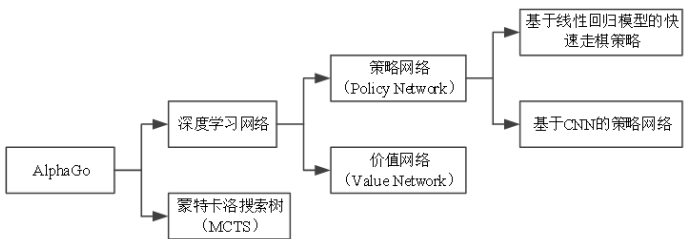


图 2-1 AlphaGo 算法组成

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(刘知青 & 吴修竹, 2016)。围棋程序 AlphaGo 的成功充分验证了深度学习与机器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。

在人工智能和博弈论的交叉领域, 博弈的类型成为塑造策略和决策模型的关键因素。博弈可以分为两大类: 完备信息博弈和非完备信息博弈。

完备信息博弈是指所有参与者在任何时间点都拥有关于博弈状态的完整知识和信息, 而参与者之间的策略选择直接影响博弈的发展和结果。这类博弈在许多经典的游戏中得到体现, 如印度女子工程学院 Sneha Garg 等学者设计了多带图灵机模型, 为井字棋游戏提供了新颖的计算决策方法(Garg & Songara, 2016); 孟加拉国东三角洲大学的 Muhammad Sakib Khan Inan 和 Rizwan Hasan 等人, 通过集成专家系统, 使用 XGBoost 算法和规则推理, 成功地在井字棋游戏中与极小化极大算法打成平局, 展示了监督机器学习在游戏 AI 中的潜力(Inan, Hasan, & Prama, 2021); 印度希里·巴格温·马法塔拉尔理工学院的 Mihir Rajesh Panchal 等人, 结合不同机器学习算法, 成功预测了象棋游戏状态(Panchal, Dodia, & Rathod, 2023); Sajjad Ghazanfari Shabankareh 等在伊朗和法国的研究机构, 通过结合遗传算法和神经网络改进了边缘跟踪方法, 显著提升了边缘检测的准确性和质量等(Shabankareh & Shabankareh, 2019)。这些都体现了完

备信息博弈在计算决策方法创新、监督机器学习与规则推理集成、机器学习算法应用和复杂计算任务优化等方面的广泛应用和深入研究。通过这些案例，完备信息博弈展示了其在推动人工智能技术发展和应用方面的重要作用，为策略制定、决策优化和复杂任务处理提供了坚实的理论基础和实践指导。

相比之下，非完备信息博弈则引入了信息的不确定性，其中参与者可能无法完全了解对手的策略、状态或意图，如 Ahmed H. Anwar 等在奥兰多中央佛罗里达大学开发了 MTD 框架，分析云提供商与攻击者在虚拟机迁移中的博弈，提出博弈论公式描述共置时机，验证 Nash 均衡，为首篇虚拟机迁移时机博弈研究，提供新防御视角 (Anwar, Atia, & Guirguis, 2021)；新南威尔士大学研究团队由 Shuo Yang 等人使用 PPO 算法在《Ticket to Ride》中训练 RL 代理，通过对抗和循环赛评估，发现 RL 代理优于启发式代理，RLSelfplayAI 表现最佳。论文证明了 RL 在复杂非完备信息博弈中的潜力，并提供了可推广的建模方案 (Yang et al., 2023)；捷克技术大学的沃伊捷赫·科瓦里克等人提出了不完美信息零和博弈的深度限制求解方法，并实验验证了其有效性，该方法对相关研究具有重要推动作用 (Kovářík & V, 2019)；来自英国帝国理工学院的 Francesco Belardinelli 等人提出了一种抽象细化框架，用于验证不完全信息下的多智能体系统战略属性 (Belardinelli, Ferrando, & Malvone, 2023)；日本高知工科大学的 Lucien Troillet 等则通过 DREAM 算法的智能体分析了不完全信息对 Geister 棋盘游戏策略的影响 (Troillet & Matsuzaki, 2024)；

瑞典皇家理工学院的 Dilian Gurov 等研究者探讨了多智能体团队在不完全信息下如何合成基于知识的策略 (Gurov, Goranko, & Lundberg, 2022); 日本东京电气通信大学研究者 Moyuru Kurita 和 Kunihiro Hoki 等则提出了一种构建麻将游戏中 AI 玩家的新方法, 通过抽象为马尔可夫决策过程来构建高效搜索树 (Kurita & Hoki, 2021); 麻省理工学院的 Dimitris Bertsimas 和哥伦比亚大学的 Alex Paskov 共同完成针对不完美信息游戏中的可解释性问题, 提出了新颖的特征表示方法和最优决策树, 通过与世界级扑克 AI 的对局实验验证了其有效性 (Bertsimas & Paskov, 2022)。利物浦大学的 George Christodoulou 等学者, 设计了贝叶斯博弈中资源分配问题的成本分担方法, 并提出了预算平衡在均衡的替代性要求, 改进了价格混乱度的界限 (Christodoulou, Leonardi, & Sgouritsa, 2016); 巴黎经济学院的 Andr  s Salamanca, 研究了不完全信息下具有可转移效用博弈中的价值概念, 并构建了货币转移作为额外附加支付的模型, 证明了相关值的推广和等价性 (Salamanca, 2020); 瑞典皇家理工学院的图通措格鲁, 费里敦研究了边缘计算中服务缓存和定价的问题, 提出了贝叶斯高斯过程多臂老虎机方法, 并证明了其算法的性能优于现有算法 (Mehdi, HoumairiAdam, Miriman, & BellafkihMostafa, 2024); 而摩洛哥国立邮电大学坎杜西, 埃尔迈赫迪研究了云安全中的挑战, 提出了一种基于贝叶斯博弈理论的动态防御机制, 并通过数值结果证明了其方案在提高云环境安全性方面的有效性 (T  t  nc  ođlu & D  n, 2024); 意大利特伦托大学的 Samuel Rota

Bulò 等提出了随机预测游戏框架，有效提升了分类器的安全性和鲁棒性(Bulò, Biggio, Pillai, Pelillo, & Roli, 2017); 塞尔丘克大学的 Mustafa A. Al-Asadi 等利用 FIFA 视频游戏数据成功预测了足球运动员的市场价值，为足球转会市场提供了有价值的参考(Al-Asadi & Tasdemir, 2022); 奥地利 MCI 因斯布鲁克大学的 Florian Merkle 等研究了对抗性机器学习的经济学问题，通过扩展游戏理论框架，分析了攻击者和防御者的经济激励和策略选择(Merkle, Samsinger, Schöttle, & Pevny, 2024)。与此同时，在理论基础层面，研究视野正从静态模型扩展到更符合现实世界的动态交互模型。例如，北京大学李阿明团队在《PNAS》上发表的关于“时序相互作用促进集体合作”的研究，突破了传统静态网络的限制，揭示了在动态变化的交互网络中，通过优化互动时序能更有效地促进群体合作，为构建多智能体协作系统提供了新颖的理论框架(Meng, McAvoy, & Li, 2025)。研究前沿正将大型语言模型与机器博弈范式深度结合。Yuan 等人提出的 MARS 框架，通过多轮合作与竞争性战略游戏中进行端到端强化学习自我对弈，其关键价值在于，该能力可泛化至游戏之外的通用推理任务，这强有力地验证了机器博弈作为培养高级决策模型的关键途径的深远价值(Yuan et al., 2025)。通过这些案例，非完备信息博弈的研究不仅推动了人工智能技术的发展，也为构建具备通用战略推理能力的智能体奠定了坚实基础。通过这些案例，非完备信息博弈展示了其在推动人工智能技术发展、复杂决策制定、多智能体

协作和对抗策略优化方面的重要作用，为相关领域的研究和应用提供了坚实的理论基础和实践指导。

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比较容易算出来的，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。

通过对这两个博弈类型的比较和研究，我们不仅能够更全面地理解博弈理论的复杂性，还能洞察人工智能在策略设计和决策支持中的

巨大潜力。

2.2.2 国内的研究现状

国内机器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，机器博弈氛围沉寂。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机器博弈技术培训与讲座，申请控制与决策国际会议（Chinese Control and Decision Conferen，简称 CCDC）机器博弈专题。近几年，重庆理工大学张小川教授在《重庆理工大学学报（自然科学版）》申请了机器博弈专题，并与国际机器博弈协会（International Computer Games Association，简称 ICGA）开展交流与合作。上述举措都极大地促进了国内机器博弈的研究与发展。

2010 年后，国内机器博弈领域不仅在理论研究上有所突破，还在实际应用中取得了诸多成果。机器博弈作为人工智能领域的一个重要分支，其研究涵盖了棋类博弈、牌类博弈、电子竞技等多个方面，涉及搜索算法、估值函数、强化学习等关键技术，特别是在深度学习和强化学习技术的应用上取得了显著突破。许多研究机构和企业正致力于将最新的机器学习和人工智能技术应用于机器博弈中，以实现更高效的策略搜索和更精准的局势评估。当前，博弈论与人工智能领域的交叉研究日益兴盛，其中完备信息博弈与非完备信息博弈作为核心议题，其研究现状备受瞩目，不仅在理论构建、算法创新上取得了显著进展，也在实际应用中面临着一系列独特而复杂的挑战。"

在完备信息博弈领域，随着计算技术的飞速发展和算法的不断优

化，机器博弈的研究已经取得了令人瞩目的成果。在这类博弈中，机器可以获取到全部的博弈信息，从而进行更为精确的策略分析和决策。目前，该领域的研究重点主要集中在如何设计更高效的算法、提升机器的自主学习能力，以及探索机器博弈在实际应用中的更广泛可能性。学者们结合棋牌各自特点，针对机器博弈搜索、评估与优化等方面开展了深入研究。例如，中央民族大学的李霞丽对藏族传统棋类游戏“久”棋的机器博弈算法与软件开发的研究。他们通过数据采集和棋型提取，设计了基于棋型的攻防策略，并开发了具有多种功能的“久”棋博弈软件(李霞丽，吴立成，& 李永集，2018)。对于局面估值问题，沈阳理工大学的吕艳辉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(吕艳辉 & 官瑞敏，2012)；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(徐长明，2010；徐长明，马宗民，& 徐心和，2009)，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(张小川，陈光年，张世强，孙可均，& 李祖枢，2010)，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(李学俊，王小龙，吴蕾，& 刘慧婷，2015)；针对中国象棋，东北大学王骄教授对机器博弈开局库(魏钦刚，王骄，徐心和，& 南晓斐，2007)、循环判定规则(王骄 et al.，2012)、评估函数参数组的自动调整和优化(王骄，王涛，罗艳红，& 徐心和，2005)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(李淑琴，李静波，韩裕华，&

陶思拓, 2012); 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(张利群, 2016); 针对亚马逊棋, 沈阳航空航天大学邱虹坤对搜索算法、评估与调参进行了分析研究(Hongkun, Peng, Yajie, Jiehong, & Fei, 2014; Jianning, Hongkun, Yajie, Fei, & Shengran, 2016; 邱虹坤, 王浩宇, & 王亚杰, 2022), 李淑琴教授根据特征权重值给出一个分阶段的评估函数(郭琴琴, 李淑琴, & 包华, 2012); 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(Jiao et al., 2017; Xinyang et al.)。完备信息博弈领域的研究已经取得了显著成果, 为机器博弈技术的发展奠定了坚实基础。

目前, 非完备信息博弈领域的研究正处于快速发展阶段, 逐渐成为人工智能和计算机科学领域前沿热点。非完备信息机器博弈策略的研究, 不断推动着机器智能决策和策略推理能力的提升, 尤其是在扑克比赛、无人机对抗、兵器推演等复杂决策场景中, 其重要性愈发凸显。

在算法理论层面, 多智能体强化学习与进化计算等前沿方向的融合持续深化。例如, 南京航空航天大学陈洪放等人提出了基于策略优化和表征搜索的改进多智能体进化强化学习方法(SORS), 该方法为处理多智能体非平稳性和部分可观测性难题提供了新颖的解决方案(陈洪放, 王秋红, 顾晶晶, & 张凯, 2025)。在高端应用领域, 机器博弈技术在国防智能决策中的价值日益凸显。南昌航空大学郑巍等人针对非均势空战中的策略趋同问题, 提出了基于非对称自博弈的智能

决策方法。有效提升了在武器失衡和数量失衡场景下的决策效能（郑巍，汤佳豪，熊小平，& 樊鑫，2025）。例如，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(Hongkun et al., 2014; 马骁，王轩，& 王晓龙，2010; 张加佳，2016)。中央民族大学吴立成等人提出了一种基于卷积神经网络的“拱猪”博弈算法，包括亮牌和出牌两个网络，在真实牌谱数据集上训练后，达到了较高的准确率，能够产生合理的博弈策略(吴立成，吴启飞，钟宏鸣，王世尧，& 李霞丽，2023)。中央民族大学李霞丽等人综述了麻将博弈 AI 的构建方法，分析了基于知识和基于数据的方法，探讨了当前最高水平的麻将 AI Suphx 的构建原理及其优缺点，指出了未来研究方向(吴立成 et al., 2023)。沈阳航空航天大学邱虹坤基于数据库和经验分析技术，优化了桥牌博弈模型并构建了混合策略打牌模型，显著提高了桥牌 AI 的综合能力和获胜几率(邱虹坤，郑晓东，& 王亚杰，2021)。中南林业科技大学柴化云提出了基于深度神经网络的斗地主博弈智能体设计方法，结合置换表技术和迭代深化搜索算法，提高了系统博弈水平，并克服了高维度状态空间下对手决策评估的难点(柴化云 & 王福成，2020)。来自深圳市腾讯计算机系统有限公司的徐方婧等人设计了一种基于卷积神经网络的“斗地主”策略，通过蒙特卡洛树自我博弈训练网络，实现了较高的胜率，验证了算法的有效性与可行性(徐方婧，魏鲲鹏，王以松，彭啟文，& 于小民，2020)。上海大学唐杰等人提出了一种面向智能博弈游戏的卷积神经网络估值方法，应用于德州扑克中，通

过训练模型学习人类大师经验，增强了博弈程序的牌力，取得了较高的胜率(唐杰，许华虎，& 谈广云，2020)。南昌大学雷捷维等人结合 Expectimax 搜索与 Double DQN 算法，提出了一种非完备信息博弈算法，应用于麻将游戏中，显著提高了博弈性能和胜率(雷捷维，王嘉旻，任航，闫天伟，& 黄伟，2021)。

在扑克比赛这一非完备信息博弈的经典场景中，算法研究同样取得了显著进展。特别是基于反事实后悔最小化(CFR)及其变体的算法，在扑克比赛中展现出了强大的求解能力。如 DeepStack、Libratus 和 Pluribus 等智能体，通过融合 CFR 与深度学习技术，成功战胜了顶级人类玩家，充分证明了 CFR 及其变体在非完备信息博弈中的广泛应用前景(王帅 & 雷跃明，2017)。

在无人机对抗领域，鉴于战场环境的复杂多变以及信息的不确定性，传统的完备信息博弈策略已难以满足实际需求。因此，探索非完备信息下的无人机博弈策略显得尤为迫切。南京航空航天大学王琳蒙在无人机非完备信息攻防博弈策略方面的研究成果。研究团队通过建立无人机攻防博弈模型，并提出一种基于反向学习的改进麻雀算法，解决了非完备信息下的无人机攻防决策问题(王琳蒙，王玉惠，陈谋，& 刘昊天，2022)。通过引入机器博弈等相关人工智能技术，军用无人机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(仝秉达，段海滨，& 魏晨，2024；朱丰 & 胡晓峰，2016)。

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(周超, 胡晓峰, 郑书奎, & 夏荣祥, 2017; 朱丰 & 胡晓峰, 2016); 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(郭圣明, 贺筱媛, 胡晓峰, 吴琳, & 欧微, 2016)。

国内在机器博弈领域的研究展现出了蓬勃的发展态势和深厚的创新能力。学者们不仅在传统棋类游戏的机器博弈搜索、评估与优化等方面取得了显著成果, 如提出结合时间差分算法和神经网络的局面估值算法、应用遗传算法优化评估函数等, 还成功将机器博弈技术拓展到电子竞技、自动驾驶、金融交易等多个新兴领域, 展现了技术的广泛应用前景。越来越多的高校、研究机构和企业投入到机器博弈的研究中, 形成了强大的研究力量。在这些研究中, 多种创新算法和技术的提出, 如干扰生成对抗强化学习(田大江 et al., 2024)、Expectimax 搜索与 Double DQN 算法的结合(雷捷维 et al., 2021)、基于卷积神经网络的博弈策略等(吴立成 et al., 2023), 不仅提高了博弈系统的性能和胜率, 还推动了相关领域的理论和技术进步。同时, 针对特定棋牌游戏的深入研究, 如六子棋、中国象棋、苏拉卡尔塔棋等, 为机器博弈技术的精细化发展提供了有力支撑。这些机构不仅在国内进行竞争与合作, 还积极参与国际上的机器博弈比赛和学术交流, 不断提升自身的研究水平和影响力。此外, 活跃的学术生态持续推动着知识的交融与创新。在 2025 年中国机器博弈学术会议上, 来自东北大学的王骄教授探讨了“复杂场景下的机器博弈方法与应用”, 中央民族大学的李霞丽教授则分享了“知识驱动的分层分

阶段藏族久棋神经网络模型”，这些工作鲜明地体现了国内研究正从通用算法探索走向针对特定文化和复杂问题的专业化、精细化探索。国内高校、研究机构和企业之间的紧密合作与竞争，以及政府的积极支持，共同构建了一个充满活力的机器博弈研究生态。这种生态不仅促进了技术的快速迭代和进步，还培养了大量专业人才，为机器博弈技术的长期发展奠定了坚实基础。

总之，国内机器博弈研究在理论创新、技术应用和生态建设等方面均取得了显著成就，展现出了强大的研究实力和广阔的发展前景。未来，随着技术的不断进步和应用领域的不断拓展，机器博弈技术将在更多领域发挥重要作用，为人类社会带来更多便利和福祉。

2.3 机器博弈产业现状

在深度学习和人工智能技术的推动下，机器博弈相关产业在近年来取得了显著的发展。特别是以 AlphaGo 为代表的完备信息博弈研究，通过深度学习获得棋感直觉，并结合蒙特卡洛搜索树技术，成功击败了人类围棋冠军。在不完备信息博弈领域，如德州扑克，Libratus 系统通过赛前纳什均衡近似、残局解算和持续自我强化等关键模块，展现了机器博弈技术的新高度。

机器博弈相关技术的发展，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。庞大的机器博弈产业吸引了众多公司争相跟进，学术界与产业界结合日趋紧密。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。

以游戏产业为例，人工智能不仅提高了游戏的趣味性，还使游戏变得更精致。技术创新是游戏产业发展的重要驱动力，而国家对电子竞技的大力支持（将电子竞技纳入第 99 个体育项目，并成为教育部增补专业），是游戏产业发展的重要保障。

2001 年至今，游戏行业的市场规模不断扩大。全球范围内游戏行业的竞争格局较为集中，全球游戏市场也保持着稳定的增长态势，2024 年全球游戏市场规模超过 2000 亿美元，玩家总数达 33 亿。相对于国外，我国游戏产业发展较快。2023 年底，中国国内游戏市场达到了 3029 亿元，同比增长 14%。2024 年上半年，中国游戏市场实际销售收入达到 1472.67 亿元，同比增长 2.08%，市场增长趋势平稳。其中，移动端游戏作为中国游戏市场的主力军，占整体市场的 75% 左右。此外，小游戏市场快速崛起，2024 年上半年收入破 166 亿，同比大增 60.5%。游戏用户规模达到 6.74 亿人，同比增长 0.88%，显示出国内游戏市场的广泛覆盖和深厚基础。腾讯和网易占据了中国市场较大的份额（腾讯约占 60%，网易约占 30% 多），尽管头部厂商占据优势，但一些新的游戏公司 and 工作室也不断涌现，给行业带来了新的活力。未来，随着 5G、VR/AR 等新技术的普及，技术创新和游戏体验将进一步优化，游戏产业将吸引更多用户，保持继续增长态势。

伴随机器博弈算法等技术的日趋成熟，相关的 AI 算力支撑硬件也在快速发展。在市场规模方面，中国的人工智能芯片市场价值达 70 亿美元，华为的昇腾芯片系列在 AI 算法方面可以与英伟达的 A100 相媲美，显示出中国在机器博弈领域的竞争力。华为昇腾 910B 芯片

在训练大规模语言模型时，相比英伟达的 A100，效率提升高达 80%，在具体测试性能上，超越 A100 AI GPU 20%。这表明中国在 AI 芯片领域正逐步减少对外部技术的依赖，并在全球市场中占据越来越重要的位置。

展望未来五到十年，随着机器博弈开发和应用的相关技术“从概念化到产业化”，机器博弈技术将进入实用阶段。该领域的产业发展呈现出更深刻的智能化与生态化趋势。在竞争模式上，领先企业正从单一产品竞争转向构建产业生态的“生态博弈”，通过开源关键数据集、组建产业联盟等方式确立行业标准。在核心技术层面，具身智能机器人已能理解并执行“铺床”等抽象任务指令，并在失败后自主纠错，标志着智能体对物理世界的干预能力迈上新台阶。更根本的突破在于，AI 已开始参与并优化强化学习算法本身的设计，实现了核心技术的自我进化。机器博弈与相关领域产学研相结合，将极大地激发科研工作者的创新热情，加速推进科技创新和技术进步。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，为我国智能化建设，引领机器博弈产业发展，发挥巨大的潜在应用价值。机器博弈产业将继续作为推动人工智能发展的关键力量，进一步强化其在国家科技进步中的关键引擎作用。

2.4 面临的问题与展望

在过去的几十年里，机器博弈研究成果对推动人工智能发展具有重要意义，但我国在该领域迈向全球领先水平的过程中，仍面临一系列核心挑战与“卡脖子”技术瓶颈。

（一） 面临的关键问题与“卡脖子”技术瓶颈

具体而言，当前的局限性主要体现在以下四个相互关联的层面：

1. 底层算力底座受制于人，高端训练生态存在差距。我国机器博弈研究，特别是面向通用智能体的大规模模型训练，严重依赖国际巨头的高性能 AI 芯片（如 GPU）及其封闭的软件生态。尽管国产 AI 芯片在特定场景已取得突破，但在支撑千亿参数级模型进行持续自我博弈所需的算力密度、互联带宽与软件栈成熟度上，与国际顶尖水平仍有差距，这直接制约了前沿探索的自主性与迭代效率。

2. 核心算法与框架创新不足，范式引领能力有待加强。当前主导性的算法范式（如 Alpha 系列、CFR 系列）多由国外机构定义。国内研究多为跟踪、改进与应用，在开创具有全球影响力的通用非完备信息求解器、超大规模博弈树搜索新方法、以及多智能体协作与对抗的理论模型等方面，原创性贡献仍显不足。这导致我们在学术话语权和未来产业标准制定中易处于被动。

3. 开源平台与基准数据集生态薄弱，协同创新门槛高。相较于国际上有 OpenSpiel、PettingZoo 等被广泛采纳的开源平台和标准基准，国内缺乏统一的、高质量的机器博弈开发与评测基础设施。这抬高了广大研究者，尤其是中小团队和初学者的技术门槛，不利于形成协同创新的开源社区，也使得国内优秀成果在进行国际对标时面临障碍。

4. 高水平复合型人才短缺，产学研深度融合不足。机器博弈深度交叉的特性，要求人才兼具深厚的博弈论、人工智能与系统工程能力。

目前，既精通核心算法又能解决产业实际问题的尖端人才和工程师队伍供给不足。同时，产学研链条存在脱节：一方面，企业难以获得前沿算法的持续支持；另一方面，学术界的研究成果缺乏充足的资金与真实的场景进行工程化淬炼与产业化落地。

5. 伦理规范与安全治理体系尚不完善。随着机器博弈技术在军事、金融、自动驾驶等高风险领域的深度应用，其伴生的伦理与安全挑战日益凸显。首先，存在技术滥用风险，博弈决策能力可能被用于开发更高效的自主攻击系统、进行市场操纵或设计诱导成瘾的交互界面，对社会安全与公平构成潜在威胁。其次，算法面临偏见与价值对齐的严峻挑战，智能体在自我博弈中可能涌现出人类难以预测甚至违背人类价值观的策略，例如为达胜利不择手段，因此如何确保其决策与人类社会的安全、公正、可信等伦理准则对齐是一项巨大挑战。此外，责任归属的模糊性也带来了治理难题，当基于博弈 AI 的自主系统（如无人机、交易系统）造成损害时，其决策过程的黑箱特性使得事故责任认定变得极其困难。

（二）发展机遇与突破路径

尽管挑战严峻，但国家层面的战略指引为机器博弈的发展注入了强大动力。2017 年国务院印发的《新一代人工智能发展规划》明确了包括智能决策理论在内的重点方向；2024 年政府工作报告提出“人工智能+”行动，标志着 AI 与产业融合已上升为国家战略。为此，我们必须把握机遇，规划清晰的突破路径：

1. 聚力软硬件协同创新，夯实自主算力基础。设立专项鼓励国产

AI 芯片厂商与顶尖研究团队合作，针对博弈负载进行芯片级优化，并共同构建成熟的国产软硬件一体化博弈算法栈，逐步在特定领域形成性能优势和应用生态。

2. 鼓励前沿探索与范式变革，力争算法引领。在国家重点研发计划中前瞻布局“通用博弈智能”等方向，鼓励高风险、高回报的原始创新。支持科研机构探索如大语言模型与博弈搜索融合等新路径，力争在核心算法和理论上实现从“0 到 1”的突破。

3. 构建开源开放平台，繁荣协同研发生态。由学会、头部企业联合高校，共同打造中国主导的机器博弈开源平台，集成主流项目接口、评估基准和算法库，并建设开放棋谱数据库。通过举办竞赛、设立基金等方式，快速吸引全球开发者，构建健康生态。

4. 深化产学研用融合，培养与汇聚高端人才。推动建立“企业出题、学界答题、成果共享”的联合实验室和创新联盟。以国家重大需求和应用场景为牵引，在实战中锤炼一支具备全链条能力的国家队，并依托竞赛和项目，大规模培养青年创新人才。

5. 构建负责任的创新体系，前瞻部署伦理与安全治理。在推动技术突破的同时，必须同步建立与之配套的伦理与安全框架。具体而言，应积极推动伦理准则研究，鼓励开展机器博弈的伦理学研究，明确技术开发的红线与底线，特别是在生命攸关和国计民生领域的应用规范。同时，需重点发展价值对齐与安全验证技术，在研究投入上支持安全、可靠、可控的博弈 AI 开发，致力于研发能够检测和纠正危险策略、确保系统行为符合设计预期的关键技术。此外，还应探索有

效的治理与监管机制，推动产学研各方共同参与，针对高风险应用场景建立审计、监督与问责机制，最终形成覆盖技术研发、部署与应用全周期的治理体系，确保技术创新始终行驶在健康发展的轨道上。

未来机器博弈将呈现多学科技术融合、产教学研相结合等趋势(徐长明 et al., 2009)。具体体现在：

(1) 复杂化

机器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。研究重点逐步由完备信息（完全信息、完美信息）博弈转到非完备信息（非完全信息、非完美信息）博弈。随着技术的进一步创新和应用场景的拓展，非完备信息博弈将展现出更为强大的实力和潜力，为人工智能领域的发展注入新的活力。

(2) 固件化

机器博弈软件与硬件的结合越来越密切，固化机器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：具有博弈思维能力的机器人、具有智能决策控制系统的无人驾驶汽车和具有群智博弈能力的无人机等。

(3) 集成化

为解决某类特定问题，机器博弈技术将与并行计算、大数据、大模型、知识工程、群智算法、人工神经网络、深度学习、类脑思维等人工智能技术将进一步集成融合。

(4) 智能化

机器博弈技术将呈现高度智能化趋势，合理拓展现有博弈技术，

深入研究更加智能的普适算法，使得机器博弈程序的类脑智能越来越高。构建具有强人工智能（甚至超人工智能）能力的通用机器博弈系统，将成为未来机器博弈研究的重点。

（5）工程化

机器博弈越来越注重实际工程应用，紧密地结合经济、娱乐、医疗、教育、智能交通、航空航天、国防等各个领域实际问题，在此基础上开展一系列人工智能领域的科学研究。例如虚拟现实仿真智能教育、人机交互沉浸式体感游戏、航空航天多学科协同综合设计等。

（6）产业化

机器博弈研究学术成果加速向产业化转变，学术界与产业界的结合日趋紧密，助力游戏开发、智能教育、智慧医疗、交通运输、航空航天、国防等相关领域的产业发展。

另外，随着各种新技术被越来越多地运用到机器博弈中，作为机器博弈技术推广、交流与检验的平台，中国人工智能学会机器博弈专业委员会组织的各类计算机博弈比赛将越来越被社会各界所认同。

可以预见，随着机器博弈技术的快速发展，我们不仅要追求机器智能在博弈中能力的新高度，探索人机智能在协同中融合的新深度，更要肩负起引导技术向善、确保安全可控的新责任。机器的胜利是科学发展的必然，而构建对人类透明、可信、可协作且符合伦理规范的博弈智能，则是未来更大的挑战与机遇。推动机器博弈技术的负责任创新，确保其发展始终服务于人类的整体利益与社会的和平进步，这必将成为人类创造力与智慧在人工智能时代更为辉煌和成熟的结晶。

2.5 国际对比与发展启示

要清晰定位我国在全球机器博弈格局中的坐标，明确自身优势与短板，需从基础研究、驱动模式、平台生态与战略布局等多个维度进行系统性的国际对比。

1. 基础理论与核心算法：引领者与快速跟随者。全球机器博弈的创新范式长期由少数国际顶尖机构定义。美国与英国的研究机构（如 DeepMind、OpenAI、卡耐基梅隆大学）扮演着“范式开创者”的角色，其成果如 AlphaGo、AlphaZero、Libratus 等，实现了深度学习与蒙特卡洛树搜索的结合、反事实遗憾最小化算法家族等基础性突破，旨在攻克星际争霸、Dota2 等超高复杂度游戏，其目标直指通用人工智能。相比之下，中国的研究力量（以高校为主体）更多地扮演着“卓越的改进者与应用者”。我们展现出强大的工程实现与算法优化能力，能够迅速吸收国际先进算法，并将其高效应用于六子棋、藏族久棋、麻将等具有文化特色的博弈项目中，且在特定棋种的竞赛水平上已达世界前列。然而，在提出全新的、具有全球影响力的通用博弈理论模型与核心算法方面，原始创新贡献仍有提升空间。

2. 研究驱动模式：产业前沿与学术社群双轨并行。国际上前沿研究的驱动模式呈现出“产业界尖端实验室主导，顶尖高校深度协同”的鲜明特征。企业以前瞻的 AGI 愿景为牵引，投入巨量的计算与工程资源，其研究具备高度的系统性和连续性。中国的研发生态则以活跃的学术社群和大规模竞赛为核心驱动力，通过国家自然科学基金等项目支持，形成了独特的人才培养与技术创新土壤。尽管出现了腾讯“绝

艺”、新睿桥牌等产业界成功案例，但具备国际影响力的、以机器博弈为核心探索路径的顶级产业研究实验室仍相对匮乏，产业界在前沿探索的投入强度与战略纵深上，与国际同行存在差距。

3. 平台生态与赛事体系：开放标准与本土繁荣。国际上已形成由巨头或社区主导的、标准化的开源平台生态（如 OpenAI Gym、DeepMind 的 OpenSpiel），这些平台为全球研究者提供了统一的“起跑线”，加速了创新循环。与之配套的 ICGA 计算机奥林匹克、ACPC 年度计算机扑克大赛等赛事，因其历史积淀与项目前沿性，成为全球顶尖 AI 的试金石。中国则依托庞大的高校群体，构建了世界上规模最大、最具活力的本土竞赛体系，并在博弈平台技术上积累了丰富的工程经验。然而，国内平台多为满足特定竞赛需求的“专用平台”，在开放性、标准化与计算资源支持上存在局限，可能导致技术发展在特定领域形成“内循环”。我国赛事在本土化和参与规模上优势明显，但在项目的国际前沿性及赛事的全球影响力方面，仍有提升空间。

4. 战略布局：宏观指引与精准投入。中国拥有《新一代人工智能发展规划》等强大的国家顶层战略指引，为领域发展注入了强劲的政策动能，这是我国独特的制度优势。相较之下，美国等国更多依靠市场资本和科研基金进行“自下而上”的资源配置。然而，将宏观战略优势转化为在机器博弈这一具体领域的持久领先，需要更具针对性的部署。对比国际同行在特定方向的长期、集中投入，我国在设立类似 DARPA 大挑战的、以机器博弈为核心攻关内容的国家级重大专项方面，尚有可为空间，以期实现对关键“卡脖子”技术的集中突破。

中国机器博弈事业已奠定规模化和工程化的坚实基础。未来的发展，要求在持续发挥自身优势的同时，更主动地融入全球创新网络，力争在基础理论的原始创新、产业研究生态的构建、开源平台标准的制定等方面实现突破，从“应用大国”迈向“创新强国”。

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。

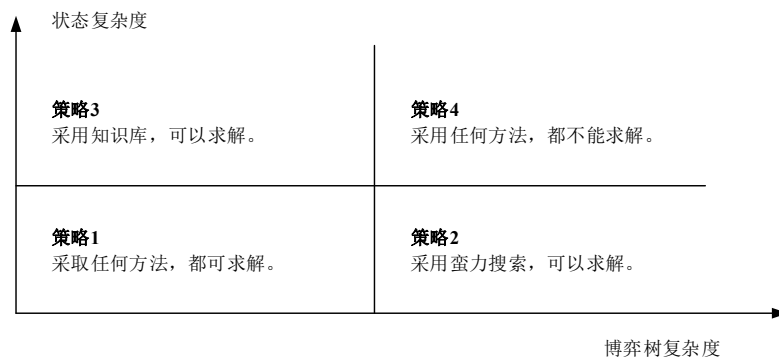


图 3-1 博弈问题的复杂度与求解策略之间的关系

文献(Herik, Uiterwijk, & Rijswijk, 2002)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(Herik et al., 2002)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(Herik et al., 2002)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(V. Allis, 1993)。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(Jonathan Schaeffer et al., 2007)，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(Lieberum, 2005)。

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。

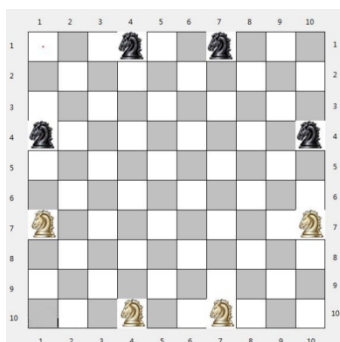


图 3-2 亚马逊棋棋盘

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。

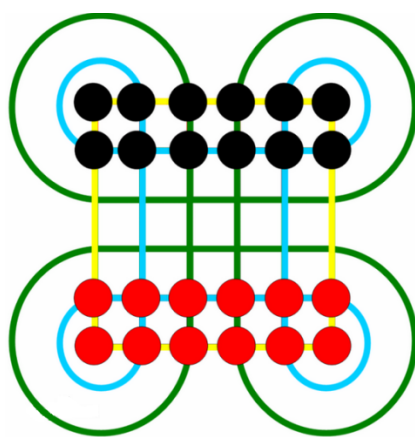


图 3-3 苏拉卡尔塔棋棋盘

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；

(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。

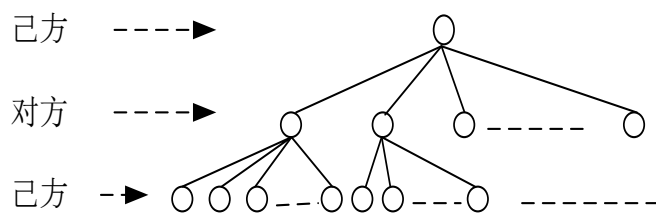


图 3-4 一个博弈树实例

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 (V. Allis, 1993)。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 (Fraenkel & Lichtenstein, 1981)）。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），一般采用平均分支因子 (李志敏, 罗里波, & 李祥, 2008) 的方法来估算博弈树复杂度。文献 (Matsubara, Iida, Grimbergen, & al., 1997) 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。

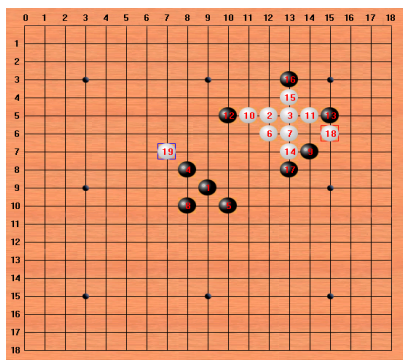


图 3-5 六子棋棋盘

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(I Chen Wu, Huang, & Chang, 2005)。

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(I Chen Wu et al., 2005)，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(Berlekamp, 2000)。

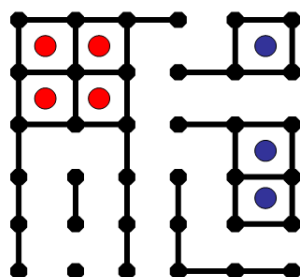


图 3-6 点格棋棋盘

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(Fraenkel & Lichtenstein, 1981)和西洋跳棋(Robson, 2006)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(Stockmeyer & Chandra, 2006)，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(Lichtenstein & Sipser, 1980)，五子棋(Reisch, 1980)、六子棋(Ming & Tsai, 2007)、奥赛罗棋

(Kasai, 1994)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(Sipser, 2006)）；亚马逊被证明属于 PSPACE-complete 问题(Hearn, 2005)，在证明过程中，它采用了一种公式博弈（Formula Game(Sipser, 2006)）。

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 $\text{Connect}(m,n,k,p,q)$ (Ming & Tsai, 2007)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(Kasai, 1994)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(Ming & Tsai, 2007)和文献(Kasai, 1994)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 (Ming & Tsai, 2007)），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(Ming & Tsai, 2007)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。

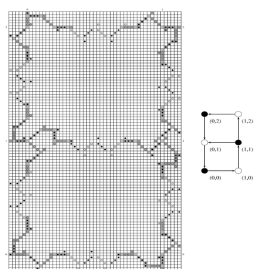


图 3-7 一个六子棋的特定模型

第4章 机器博弈的典型技术

4.1 概述

机器博弈系统中，典型的关键技术主要包括策略搜索、局面评估与优化、神经网络的学习与训练等技术，它们是决定博弈结果的关键因素。以国际跳棋、中国象棋、围棋等为代表的传统二人零和完备信息博弈，其博弈理论已经很成熟。

近几年来，随着计算机硬件算力提升，神经网络、机器学习、大数据、大模型等技术快速发展。特别是 GPU、TPU 并行计算技术的广泛应用，使得神经网络+深度学习变得更加便宜、快速、实用与有效，机器博弈系统的计算与逻辑思维能力也得到大幅提升。

尽管依靠加大搜索深度可以适当提升博弈系统的 AI，但同样也使得计算量大幅提升。在实际机器博弈项目中，通常有博弈实时博弈时间限制的要求，类似宽度优先搜索、穷尽搜索和盲目搜索一类的算法，时间和空间开销巨大，难以做到很深的搜索。因此，基本上不可能直接使用此类算法去解决相关问题，也不能单纯依靠加大搜索深度提高机器博弈能力。在机器博弈中，引入必要的相关知识会大幅提升 AI 能力，降低计算量。先验知识越丰富、越正确、对博弈局面评估得越准确，获胜的几率就越高。

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，它奠定了机器博弈的理论基础(Herik et al., 2002)。极小化极大算法通过构建一棵博弈树来寻找最优解。在博弈树中，每个节点代表一个游戏状态，而边

代表可能的走法。极小化极大算法会从叶子节点（游戏的最终状态）开始，向上评估每个节点的值，直到根节点，从而确定最优的走法。然而，这种方法在面对大型博弈树时计算量巨大，因此需要剪枝算法来减少不必要的计算。

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，是机器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 (Baudet & M., 1978) 和以此为基础改进与增强的算法，如渴望窗口搜索 (Aspiration search) (Lu & Xia, 2008)、MTD(f) (Memory-enhanced Test Driver with f and n) 搜索 (Jing, 2008) 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 (Baudet & M., 1978; Leifker & Kanal, 1985)

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。它通过引入两个参数 Alpha 和 Beta 来减少搜索的节点数，其中 Alpha 代表当前搜索分支中己方可能获得的最大值，Beta 代表对方可能获得的最小值。在搜索过程中，如果某个节点的值已经确定比 Alpha 小（对于己方）或比 Beta 大（对于对方），那么这个节点以下的分支就不需要继续搜索，因为它们不可能是最优

解。这种剪枝可以显著减少搜索空间，提高搜索效率。

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(Kaindl & Shams, 1991; Lu & Xia, 2008)

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(Jing, 2008)

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(张明亮 & 李凡长, 2009)。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 (Donkers, Uiterwijk, & Herik, 2001; 焦尚彬 & 刘丁, 2010)

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) (J. Schaeffer, 1989)

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) (Sakuta, Hashimoto, Nagashima, Uiterwijk, & Iida, 2003)

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 (Sakuta et al., 2003; J. Schaeffer, 1989)。

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 (Reinefeld & Marsland, 1994)。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 (T. A. Marsland, Reinefeld, & Schaeffer, 1987; Plaat, Schaeffer, Pijls, & Bruin, 1994) 和 DUAL* 算法，不采用极大极小方法取值的 B* (Berliner, 1979) 和 PB* (Hans et al., 1996) 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，在机器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

随机搜索算法主要通过对状态空间进行抽样来寻找近似最优解。其基本形式包括 拉斯维加斯算法与蒙特卡罗算法两类：前者在理论上保证能够找到最优解，但对采样次数无上限；后者在有限采样条件下逐渐逼近最优解，更适用于资源受限的决策场景。在机器博弈中，由于每步决策的时间与空间受到严格限制，且通常只需求得局部较优解，因此蒙特卡罗类算法更为常用。

1. 蒙特卡洛树搜索 (Monte Carlo Tree Search, MCTS) (Baier & Winands, 2011; Browne et al., 2012; Guez, Silver, & Dayan, 2013; Lorentz, 2016)

蒙特卡洛树搜索是一种用于某些决策过程中的启发式搜索算法，特别适用于机器博弈这类具有巨大状态空间的问题。它通过随机模拟来构建搜索树，而非依赖精确的估值函数，从而在复杂博弈中表现出色。

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 (Raiko & Peltonen, 2008)。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。

蒙特卡洛树搜索不仅在棋类游戏如围棋、国际象棋中表现出色，也被应用于其他需要决策的场景，如机器人路径规划、实时策略游戏等。它的一个关键优势是它不需要对游戏状态的精确评估，而是通过大量的模拟来近似最优解，这使得它在处理复杂和未知的游戏状态时特别有用。

MCTS 通过随机抽样来探索决策空间，并根据模拟结果构建搜索树，以寻找最优决策。这种算法最初是为了解决博弈问题而设计的，它通过大量的随机采样来估计未知量，特别适用于完全信息游戏、零

和游戏以及离散和确定性游戏。

MCTS 的每一次迭代都包含四个核心步骤（如图 4-1 所示）：

选择（Selection）：从根节点（当前局面）开始，递归地选择最优的子节点，直至到达一个可扩展的叶节点。选择策略通常基于 Upper Confidence Bound (UCB) 公式，以平衡“利用”已知高收益节点和“探索”访问次数少的节点。

扩展（Expansion）：当选择的叶节点不是终止状态时，为其创建一个或多个新的子节点。

模拟（Simulation）：从新扩展的节点开始，运行一段快速的随机对弈（即“走棋”），直至到达游戏终局，并得到一个胜负结果。

反向传播（Backpropagation）：将模拟得到的结局结果沿着之前选择的路径反向传递，更新路径上所有节点的统计信息（如访问次数、累计收益）。

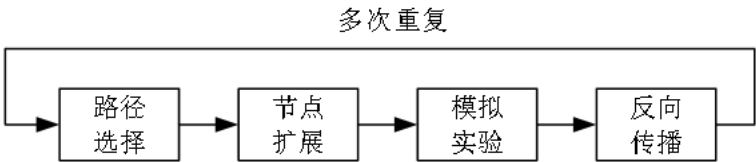


图 4-1 构造 MCTS 博弈树的过程

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 (David Silver et al., 2016)。

2. UCT 搜索算法 (Gelly & Silver, 2007b; Zhang, Wang, Lin, & Xu, 2008)

UCT 算法是将 UCB 公式应用于 MCTS 的典型代表。UCT 使得

MCTS 在超大规模博弈树的搜索过程中，表现出优异的时间和空间效率，目前已广泛应用于围棋、亚马逊棋、不围棋等完备信息博弈，以及幻影围棋、德州扑克等非完备信息博弈中，成为现代机器博弈的核心算法之一。

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 (Hashimoto, Kishimoto, Yoshizoe, & Ikeda, 2011; Kato, Takaya, & Yamamura, 2015; Raiko & Peltonen, 2008)。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 (T. A Marsland & Campbell, 1982) 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 (蔡晔, 2015; 陈国良, 孙广中, 徐云, & 龙柏, 2009; 计永昶, 丁卫群, 陈国良, & 安虹, 2001; 李之棠 & 陈华民, 1998)，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(曹婷婷，2009；卢风顺，宋君强，银福康，& 张理论，2011；聂慧，彭娇，金晶，& 李康顺，2017；王京辉 & 乔卫民，2005；杨冰，2015)。机器博弈中，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机器博弈，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(吉根林，2004)。

遗传算法是解决搜索问题的一种通用算法，在机器博弈中，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 (Amos & Coldridge, 2012; Esparcia-Alcázar, Martínez-García, Mora, Merelo, & García-Sánchez, 2010; Kim & Kim, 2013; Liaw, Wang, Tsai, Ko, & Hao, 2013; 吉根林, 2004; 王骄 et al., 2005; 王书宇, 李龙澍, & 汪群山, 2008)。

4.10 路径寻迹算法

路径寻迹算法是一类用于在图或网格中找到从起点到目标点的最优路径的算法。在机器博弈中，这些算法可以用来模拟或优化玩家或计算机在游戏中的移动策略。以下是几个常见路径寻迹算法在机器博弈中的应用解释：

1. Dijkstra 算法 (Dijkstra, 1959)

Dijkstra 算法主要用于找到从单一起点到所有其他点的最短路径，前提是每条边的权重非负。在机器博弈中，如果每个游戏动作（例如移动到相邻格子）的代价相等，Dijkstra 算法可以用来规划玩家的移动路径。例如，在战略游戏中，Dijkstra 算法可以帮助单位找

到最短路径去敌人或目标点。

2. A*算法 (Hart, Nilsson, & Raphael, 1972)

A*算法是一种广泛应用于游戏和机器人导航的路径寻迹算法。它结合了启发式搜索和标准的 Dijkstra 算法，通过引入启发函数（通常是估算从当前节点到目标节点的代价）来优化搜索效率。在机器博弈中，A*算法可以帮助 AI 找到到达目标的最优路径，适用于需要动态评估和优化路径的复杂游戏环境。

3. 广度优先搜索 (BFS) (Moore, 1959)

BFS 是一种无权图搜索算法，确保找到从起点到目标点的最短路径。在机器博弈中，BFS 适用于状态空间较小的游戏（如简单的棋类游戏），它不考虑权重，保证了路径的最小步数。BFS 简单易实现，但可能在状态空间较大的游戏中效率低下。

4. 深度优先搜索 (DFS) (None, 1986)

DFS 是一种深度优先的搜索算法，适用于状态空间很大但希望尽快找到可接受解的情况。在机器博弈中，DFS 可以用于不需要保证最短路径的复杂决策树，适用于探索策略空间深度。

5. 动态规划

动态规划通过存储和重用子问题的解来提高计算效率，适用于那些可以从子问题递推到整体问题的博弈。例如，象棋或围棋这类游戏中，动态规划可以帮助规划中期策略，优化多步决策。

路径寻迹算法在机器博弈中扮演着至关重要的角色，它们帮助 AI 在复杂的游戏环境中做出最优决策。不同的算法适用于不同类型

的游戏和优化目标,选择合适的路径寻迹算法可以显著提高机器博弈的性能和用户体验。

4.11 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(G. Hinton, 2000; Geoffrey Hinton, Deng, Yu, Dahl, & Kingsbury, 2012; G. E. Hinton, 2011; G. E. Hinton et al., 2006; Lecun, Bengio, & Hinton, 2015; Salakhutdinov & Ruslan, 2012)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。

人工神经网络在机器博弈中扮演着至关重要的角色, 通过结合深

度强化学习（如 AlphaGo 和 AlphaZero）、策略网络、价值网络和卷积神经网络（CNN），系统能够有效地学习游戏策略和状态评估。策略网络直接输出行动概率，而价值网络估计局面胜率，共同指导蒙特卡洛树搜索（MCTS）。自我对弈机制生成大量训练数据，使模型持续进化。多智能体系统中的协同学习进一步提升了策略优化。这些技术的综合应用使现代 AI 在复杂博弈中达到了超越人类的水平。

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 (Schmidhuber, 2015)，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 (David Silver et al., 2016; 吕艳辉 & 宫瑞敏, 2012; 苏攀, 王熙照, & 李艳, 2011)。

4.12 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 (Meltzoff, Kuhl, Movellan, & Sejnowski, 2009)。它利用经验来改善计算机系统自身性能 (Mitchell, 2003)，让计算机系统具有人类的学习能力，以便实现人工智能 (R. S., S., Ivan, & Miroslav, 2004)。

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（Dietterich, 1997）。

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(Sutton & Barto, 1998)。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。

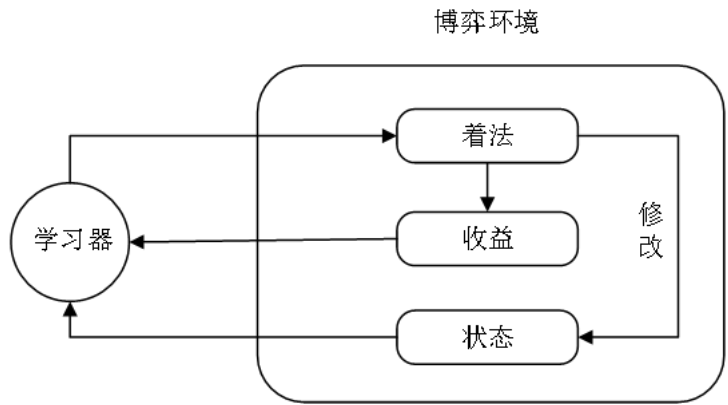


图 4-2 机器博弈强化学习设定

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 或 TPU 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 或 TPU 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(郭潇逍, 李程, & 梅俏竹, 2016), 例如采用基于深度学习和 Q-Learning 的 Deep Q-Network(Volodymyr, Koray, David, & al, 2015)技术的博弈系统已达到人类玩家水平, 而 AlphaGo(Schölkopf & Platt, 2007; David Silver et al., 2016)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, 提升机器博弈水平, 仍是今后研究的重点。

4.13 深度学习

深度学习作为机器学习的重要分支, 通过深层神经网络架构模拟人脑的层次化认知机制, 实现对数据中复杂模式的多层次抽象与表征(周雷, 尹奇跃, & 黄凯奇, 2022)。与依赖手工特征的传统机器学习方法相比, 深度学习具备从原始数据中自动学习高阶特征的能力(张千 et al., 2019), 这一特性使其在机器博弈领域展现出显著优势。通过端到端的训练范式, 深度学习与强化学习、蒙特卡洛树搜索(高锦涛, 胡志远, & 姜璐璐, 2025)等技术的深度融合, 已成为推动博弈智能体突

破性能瓶颈的核心驱动力。

在机器博弈的具体实践中，深度学习模型架构持续演进并展现出强大的适应性。卷积神经网络通过其局部连接和权值共享特性，在棋盘类游戏中有效捕捉空间拓扑关系(刘知青 & 吴修竹, 2016)，为策略评估提供结构化特征表示；循环神经网络及其长短期记忆变体，则在序列决策类博弈中展现出对历史状态依赖的建模能力(王璐瑶 & 吴蕾, 2025)。深度 Q 网络及其改进模型通过经验回放与目标网络稳定训练机制，在部分可观测环境中实现了超越传统方法的决策性能(刘剑锋, 普杰信, & 孙力帆, 2023)。值得关注的是，Transformer 架构凭借其自注意力机制，正逐渐应用于需要长程依赖建模的复杂博弈场景，为多步策略推理提供了新的技术路径(罗俊仁, 张万鹏, 苏炯铭, 王尧, & 陈璟, 2023)。

深度学习与强化学习的融合形成了深度强化学习这一重要范式，极大地推动了机器博弈技术的发展。这一范式通过价值网络与策略网络的协同优化，使智能体能够在高维状态空间中实现有效的策略搜索。其核心进展体现在多个方面：基于模型的强化学习通过构建环境动力学仿真来提升样本效率；分层强化学习将复杂任务分解为多个抽象层次，实现了策略的模块化学习；元强化学习则使智能体具备跨任务的快速适应能力(Huang et al., 2025)。自博弈机制的引入进一步加速了策略进化过程，通过持续自我对抗生成高质量训练数据，驱动智能体发现超越人类经验的纳什均衡策略。

近年来，深度学习在机器博弈中的技术演进呈现出若干重要趋

势。自监督与半监督学习方法有效缓解了对大规模标注数据的依赖，使智能体能够从有限的专家对局中提取有效知识。多模态融合技术整合视觉、文本等异构信息，为理解复杂博弈规则提供了更丰富的上下文。可解释性研究则致力于揭示深度决策模型的内在逻辑，通过注意力可视化、策略分解等方法增强智能体的透明度与可信度。大语言模型与博弈决策的交叉探索开辟了新的研究方向，其在战略推理、对手建模等方面的潜力正逐步显现。

尽管深度学习已推动机器博弈达到前所未有的水平，该领域仍面临诸多挑战。模型训练对计算资源的巨大需求限制了其普及应用，训练过程的稳定性与收敛性保证仍需进一步研究，如何在非平稳环境中维持策略的鲁棒性也是亟待解决的问题。未来，深度学习与机器博弈的融合将继续向更高效的训练算法、更通用的策略表示和更可信的决策机制方向发展，为实现通用博弈智能奠定坚实基础。

4.14 知识库

知识库技术通过结构化的方式存储领域内的专业知识、规则和案例等信息。这些信息可以来自各种来源，如书籍、论文、专家经验等，并经过整理、分类和编码后存储于知识库中。知识库技术使得计算机能够高效地访问和利用这些知识，从而支持复杂的决策和问题求解过程。

1. 知识库在机器博弈领域的应用(郭晓霞, 韩燮, & 赵融, 2018)

在机器博弈领域，知识库技术发挥着至关重要的作用，它负责存储和检索与博弈相关的丰富知识。这些知识覆盖了棋谱和开局库，其

中包含了历史上著名棋手的对弈记录以及各种常见开局的走法和应对策略，这些信息能够帮助机器在博弈的早期阶段就取得优势或避开潜在的陷阱。此外，知识库还记录了棋型和定式，即棋盘上特定形状的棋子分布及其相应的走法和策略，使得机器能够迅速识别当前棋局的优劣，并据此做出决策。最后，知识库中还包含了博弈的基本规则、高手的走法策略和战术思路等，这些构成了机器进行博弈的基础，同时也是机器学习和优化策略的重要来源。通过整合这些知识，机器博弈系统能够更深入地理解和模拟人类棋手的思维过程，从而在博弈中展现出更高的竞技水平。

2. 知识库技术在机器博弈中的优势

知识库技术在机器博弈中提供了显著的优势，它通过快速检索存储的信息，使得机器能够在极短的时间内做出高质量的决策，显著提升了博弈的效率。此外，机器可以利用知识库中积累的历史数据和统计信息，学习并模仿高手的走法策略和战术思路，通过不断的自我对弈和实践来优化自身的棋力。更为重要的是，知识库的内容可以根据最新的博弈理论和实践进行更新和扩展，这赋予了机器强大的适应性和灵活性，使其能够灵活应对多变的博弈场景和不同的对手，始终保持竞争力。

知识库技术在机器博弈领域发挥着重要作用，为机器提供了丰富的知识支持和决策依据。随着技术的不断发展，知识库技术将在机器博弈领域发挥越来越重要的作用。

4.15 微分博弈

微分博弈是指在时间连续的系统内，多个参与者进行持续的博弈，力图最优化各自独立、冲突的目标，最终获得各参与者随时间演变的策略并达到纳什均衡，即任何参与者都没有单独改变策略的意愿。微分博弈理论是求解协调控制问题的崭新思路(仝秉达 et al., 2024)。

微分博弈理论起源于20世纪50年代美国空军开展的军事对抗中双方追逃问题的研究，它是最优控制与博弈论的结合。随着博弈种类的拓展和解法的完善，微分博弈已被应用于经济学、管理学、环境科学等多个领域。在机器博弈领域，微分博弈的思想也逐渐被引入，为机器博弈提供了新的思路和方法。

微分博弈为机器博弈提供了坚实的理论基础，它使得机器能够在复杂的博弈环境中做出最优决策。微分博弈理论中的纳什均衡等概念，为机器博弈中的策略优化和决策制定提供了重要的理论指导。在算法应用方面，微分博弈中的动态规划、最优控制等方法被广泛应用于求解机器博弈中的最优策略。同时，微分博弈的连续时间模型和离散时间模型为机器博弈中的时间演化和策略更新提供了重要的参考。

尽管微分博弈在机器博弈中的应用还处于探索阶段，但它带来了新的挑战 and 机遇。微分博弈的复杂性和连续性要求机器具备更高的计算能力和智能水平，这对机器的硬件和软件都提出了更高的要求。同时，微分博弈也为机器博弈提供了新的研究方向和思路，有助于推动机器博弈领域的发展，特别是在追逃问题、兵棋推演智能化等方面展

现出巨大的潜力。因此，微分博弈不仅为机器博弈提供了理论支撑和算法应用，还带来了新的挑战 and 机遇，促进了人工智能技术的进步和应用的拓展。

随着人工智能技术的持续进步，机器博弈领域也迎来了显著的发展，其中微分博弈作为一种创新的思路和方法，展现出广阔的应用前景。未来，微分博弈有望在提升机器博弈的智能水平方面发挥关键作用，通过其思想和方法的引入，增强机器在面对复杂局面和对手策略时的决策能力，从而做出更优的决策。此外，微分博弈的应用范围有望进一步拓展，覆盖电子竞技、棋类博弈、策略游戏等多种类型的机器博弈，从而扩大机器博弈的应用领域。同时，微分博弈的研究和应用也将推动人工智能技术的创新和发展，为人工智能领域带来新的突破和进展，预示着微分博弈在智能决策和策略优化方面的巨大潜力。

4.16 贝叶斯网络与概率推理

贝叶斯网络是一种概率图模型，它能够通过图形化的方式表示变量之间的依赖关系，并利用贝叶斯定理进行概率推理。在机器博弈中，贝叶斯网络可以用于建模博弈过程中的不确定性，以及预测对手的可能动作和策略。如在扑克牌博弈中，玩家需要根据手中的牌和对手的行为来推断对手的牌型和可能的动作。贝叶斯网络可以用于建模这种不确定性，通过更新节点的概率分布来反映新获得的信息。玩家可以利用贝叶斯网络来计算不同牌型和动作的概率，从而制定最优的博弈策略。

概率推理在机器博弈中扮演着至关重要的角色。它允许机器在不

确定的环境中进行决策，通过计算不同策略的成功概率来优化其博弈行为。

在机器博弈中，概率推理可以用于以下几个方面：

对手建模：通过观察和分析对手的行为，利用概率推理来推断对手的策略、牌型或意图。这有助于机器制定更有针对性的应对策略。

风险评估：在博弈过程中，机器需要评估不同动作的风险和收益。概率推理可以帮助机器计算不同策略的成功概率和潜在损失，从而做出更明智的决策。

策略优化：通过概率推理，机器可以模拟不同策略的执行效果，并计算它们的预期收益。这有助于机器找到最优的博弈策略，以最大化其获胜的概率。

贝叶斯网络与概率推理在机器博弈中具有重要的应用价值，特别是在不完全信息博弈中。通过构建贝叶斯网络，定义条件概率表，更新信念和进行概率推理，机器可以在博弈过程中做出更优的决策。这种方法不仅适用于扑克游戏，还可以扩展到其他类型的博弈，帮助机器在复杂的不确定环境中进行有效的决策。

4.17 多智能体强化学习

多智能体强化学习 (Multi-Agent Reinforcement Learning, MARL) 研究在多个智能体同时学习并交互的环境中，如何通过学习和决策实现个体或团队收益的最大化。作为机器博弈研究的前沿阵地，它为解决桥牌、麻将、斗地主等多人棋牌游戏中的对抗、合作与混合型博弈问题提供了核心理论框架与方法论。

MARL 的核心挑战源于其环境的动态复杂性。非平稳性意味着每个智能体都在与持续进化的“对手”博弈，收敛难度剧增；信用分配难题要求在团队合作中，清晰界定每个成员动作对最终胜负的贡献；而部分可观测性则要求智能体必须在信息不完备的迷雾中做出决策。为应对这些挑战，研究者发展出多种行之有效的范式。其中，中心化训练与去中心化执行已成为最主流的范式之一，它通过在训练时利用全局信息学习一个强大的中心化评论家网络来指导智能体，而在执行时每个智能体仅依赖自身的局部观测进行独立决策，这完美契合了现实博弈中智能体无法实时通信的约束。此外，对手建模使智能体能够显式地推断并预测其他博弈方的策略与意图，从而做出更具针对性的反应；而在允许通信的场景下，智能体还可以通过学习通信协议来交换关键信息，实现更高效的团队协作。

近年来，大语言模型的崛起为 MARL 带来了革命性的赋能。LLM 强大的语义理解、上下文推理与知识泛化能力，使其能够作为一个高级的“策略大脑”嵌入多智能体系统。具体而言，LLM 能够深度赋能对手建模，通过分析历史行动序列，更精准地推测对手的风格、手牌范围甚至心理状态；在合作博弈中，LLM 驱动的智能体能够生成和理解更复杂的自然语言指令，实现超越固定协议的、更灵活和隐晦的战术配合，例如在桥牌中通过叫牌过程传递更丰富的战略信息。同时，基于 LLM 的生成式智能体能够高度模拟人类在博弈中的复杂行为模式与社会偏好（如公平、信任与报复），为在虚拟环境中研究群体合作机制的演化、社会规范的涌现等提供了前所未有的高保真仿真

平台。

MARL 与 LLM 等前沿技术的深度融合，正推动机器博弈智能体从擅长解决特定任务的“专家”，向具备通用战略理解、复杂社会交互能力的“通用决策体”演进。这一趋势不仅将持续攻克麻将、星际争霸等极具挑战性的博弈难题，更将为构建能够在开放环境中与人协同、具备高级战略思维的下一代人工智能系统奠定坚实的基础。

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(王亚杰，邱虹坤，吴燕燕，李飞，& 杨周凤，2016)。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(Fanfu, Xian, Jun, Wen, & Xiaobo, 2016)。

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(Mei et al., 2014)。

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(Xian et al., 2013)。

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(Mei, Zhang, Sun, Zhang, & Zhang, 2014)。

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(Jianning, Hongkun, Yu, Yajie, & Xueping, 2016; Qiu, Gao, Wang, Song, & Sun, 2013; Weihe et al., 2015)。

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 (Berlekamp, 2000; I. Chen Wu & Huang, 2006)，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(Wagner & Virág, 2001)和 Go-Moku(L. V. Allis, Herik, & Huntjens, 2010)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。

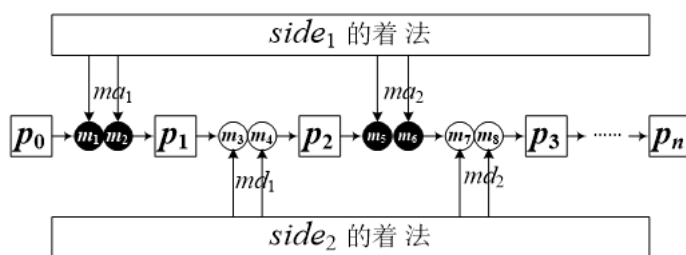


图 6-1 棋局演化过程

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(徐长明, 2010)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(I Chen Wu et al., 2005)首次定义了较为完备的模式类型；文献(徐长明, 2010)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。

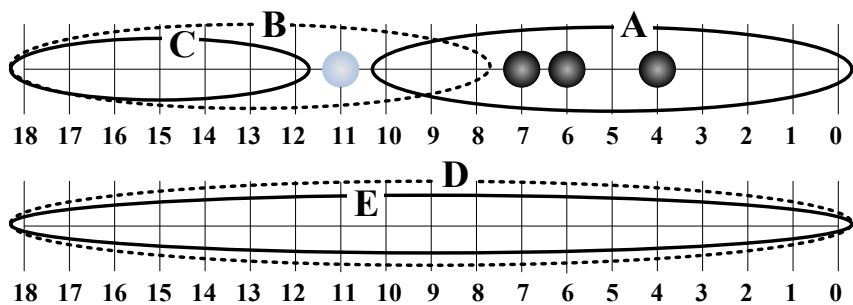


图 6-2 三进制线拆分成二进制模式的示例

全部棋形共计 1,048,512 个，在文献(徐长明, 2010)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。

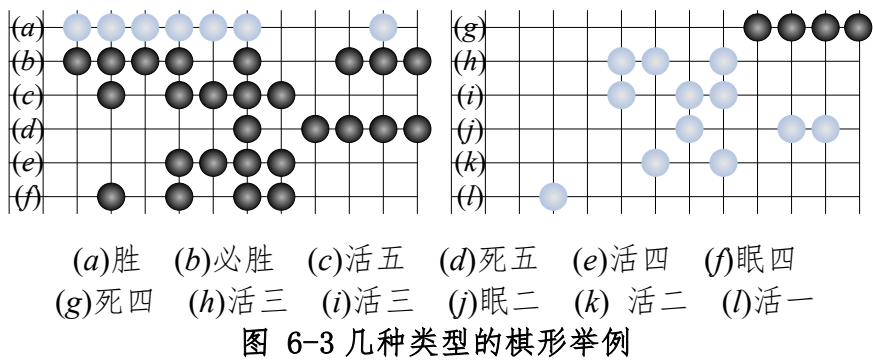


表 6-1 connect(19, 6, 2, 1) 中，不同类型模式的数目和比例

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(L. V. Allis et al., 2010)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(Volodymyr et al., 2015)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(Volodymyr et al., 2015)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(L. V. Allis et al., 2010)，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。在很长一段时期内，静态方法成为了研究的主流方向，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(Gelly & Silver, 2007a; Gelly, Wang, Munos, & Teytaud, 2006; Kocsis, 2006)彻底的改变。该算法在蒙特卡洛树中使用 UCB(Auer, Cesa, & Fischer, 2002)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(Y. Wang & Gelly, 2007)。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(Baudi & Gailly, 2011; Enzenberger, Müller, Arneson, & Segal, 2011)，直至 D. Silver 等利用深度学习对围棋领域知识进行学习(David Silver et al., 2016)。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 算法(Kocsis, 2006)，UCT 的全名为 UCB for Tree。UCB(Upper Confidence Bound)(Auer et al., 2002)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。

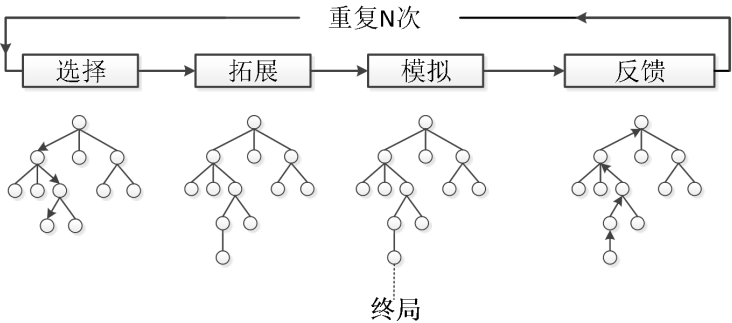


图 6-4 蒙特卡洛树搜索过程

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(Gelly & Silver, 2008)提出了模拟对局的 RAVE 增强算法，文献(Banerjee & Stone, 2007; Gelly & Silver, 2007b)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(Chaslot et al., 2008; S. Gelly, J.-B. Hoock, A. Rimmel, O. Teytaud, & Y. Kalemkarian, 2008; S. Gelly, J. B. Hoock, A. Rimmel, O. Teytaud, & Y. Kalemkarian, 2008)，王骄等提出了利用 4*4 的 Pattern 库提高模拟棋局的质量(J. Wang, Li, Chen, Wei, & Xu, 2011)，使用 OOV 算法进行特征学习(Jiao et al., 2017)。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）。

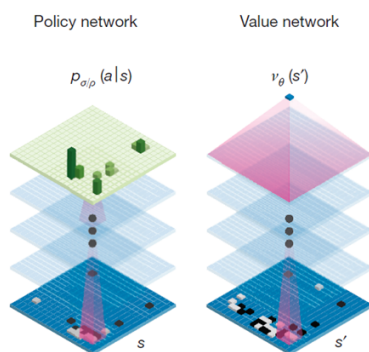


图 6-5 策略网络与估值网络

以 MCTS 为主干搜索算法（详见第 4.7 节），但对其关键组件进行了深度改造。不再使用纯粹的随机模拟，而是引入了通过监督学习和强化学习训练得到的策略网络（Policy Network）和价值网络（Value Network）。策略网络是用于在 MCTS 的选择和扩展步骤中提供高价值的落子先验概率，极大地缩小了搜索的宽度。价值网络，则用于在模拟步骤中直接评估叶节点的胜率，替代了大部分耗时的随机走棋，显著增加了搜索的深度。系统通过自我对弈生成海量高质量棋谱，用于持续迭代和优化策略网络与价值网络的参数，形成良性循环，使棋力不断进化。

更多细节详见 DeepMind 发表在 Nature 的论文(David Silver et al., 2016)，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。新一代的研究开始探索使用 Transformer 架构作为策略和价值网络的骨干。其自注意力机制能更有效地建模棋盘上任意两个落子点之间的长远依赖关系，对于理解大模样作战和全局性厚势的价值具有潜在优势，有望弥补 CNN 在全局感知上的不足。在构建围棋 AI 训练环境时，引入包含多个不同风格

AI 的种群进行大规模多智能体自博弈，可以有效避免策略退化，产生更加多样化和鲁棒的策略，防止 AI 陷入单一模式的局部最优。

6.3 点格棋机器博弈

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(Barker & Korf, 2012)，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(连莲，徐心和，张雪峰，& 颜宁, 2009)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。

2. 十字链表表示法

为方便点格棋棋局状态分析，文献(D. Li, Li, & Bao, 2011)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。

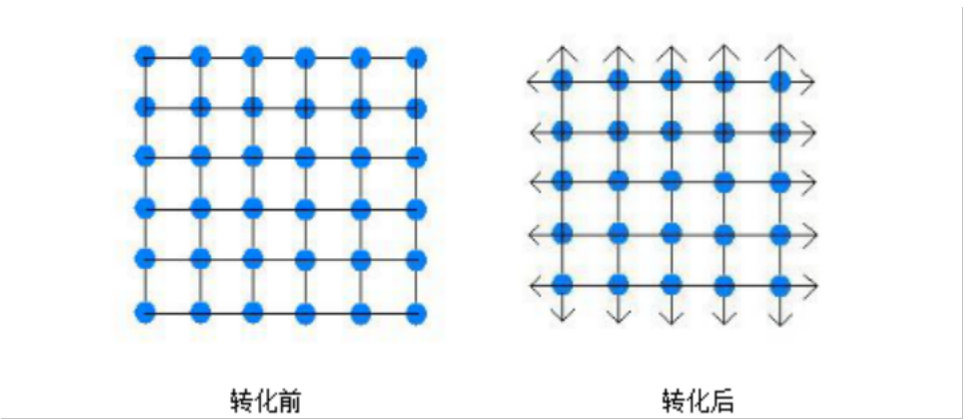


图 6-6 转化前与转化后的棋局

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。

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(Kocsis, 2006)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(David Silver et al., 2016)之后，集成深度学习(Moravík, Schmid, Burch, Lis, & Bowling, 2017)方法在计算机博弈领域得到了广泛关注。

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(Xuejim Li, Guang, Wu, & Zhang, 2015; 徐心和 et al., 2008)，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(Xuejim Li et al., 2015)，6 颗棋子必须全部置于出发区之内，如图 6-7(b)所示。

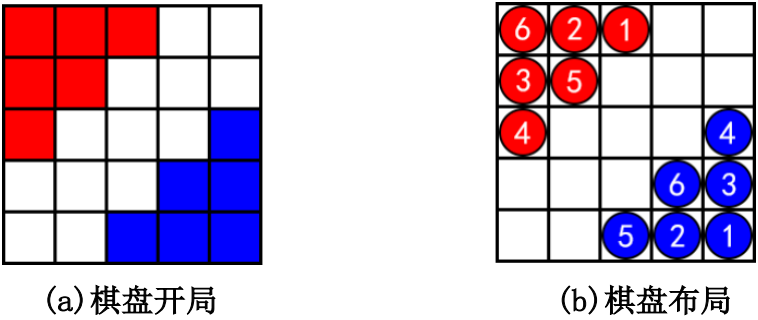
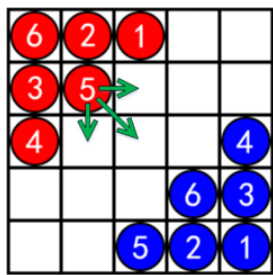


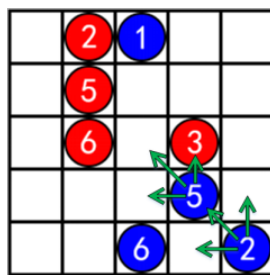
图 6-7 棋子开局与布局

(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(Xuejim Li et al., 2015)，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。

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

图 6-9 爱恩斯坦棋棋盘表示

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(Coulom, 2006; 和力 & 吴丽贤, 2012)。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(王骥 & 孙建伶, 2008)。

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(X. Wang, Wang, & Liu, 2013; X. Wang, Zhang, Wang, Qi, & Zhou, 2014; 吕艳辉 & 官瑞敏, 2012)。

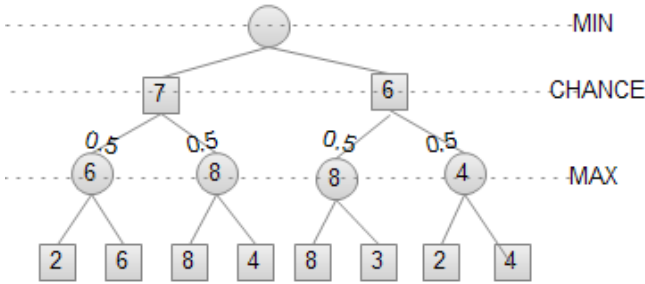


图 6-10 期望极大极小算法

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 Maximax Algorithm, ODEMA)(X. Wang et al., 2014)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。爱因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。

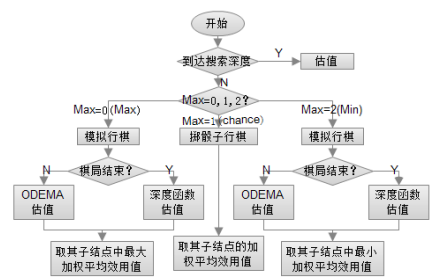


图 6-11 ODEMA 示意图

当然，在爱因斯坦机器博弈中，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。

6.5 苏拉卡尔塔棋机器博弈

6.5.1 概述

苏拉卡尔塔棋源自印尼爪哇岛的苏拉卡尔塔，是一种双人吃子游戏(李淑琴 et al., 2012)。棋盘是由横竖各 6 条边交叉构成的正方形棋盘，36 个交叉点为棋位，各边由 8 段圆弧连接，是一个高度对称的结构。双方各有 12 个棋子，棋子摆放初态如错误!未找到引用源。所示。双方轮流走棋，每个棋子均可沿着直线或对角线朝自己相邻的空位移动。本游戏的目的就是吃掉对方棋子，当一个棋子要吃掉对方棋子时，需要经过一条畅通无阻的直线及棋盘上任意一个或多个弧线，否则就不能进行吃子。吃掉对方所有棋子或进入循环，所剩棋子最多的一方取得胜利。

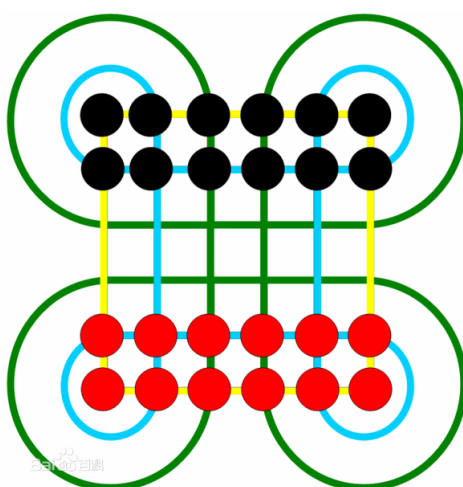


图 6-12 苏拉卡尔塔棋界面

6.5.2 苏拉卡尔塔棋规则

6.5.2.1 棋盘棋子

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。

6.5.2.2 行棋规则

- 1) 初始状态：棋子在各方底线排成 2 排。
- 2) 双方轮流走棋，每次走动一枚棋子；
- 3) 除了吃子之外，每枚棋子只能沿着垂直或对角方向走动一格，只能走向空位；
- 4) 吃对方子时必须经过至少一个完整的弧线。

6.5.2.3 胜负

- 1) 吃掉所有对方棋子一方获胜；
- 2) 进入循环，剩余棋子多的一方获胜。

6.5.3 搜索算法

6.5.3.1 主变量搜索算法 (Principal Variation search)

主变量搜索 (Principal Variation Search, 简称 PVS), 又被称作最小窗口搜索 (Minimal window Search), 或者 NegaScout 算法 (Rutko; Jonathan Schaeffer, 2000)。它基于这样的基本思想：在一个强有序的博弈树中，每个结点上第一个移动很有可能是最好，而且证明一个子树是低劣的 (inferior) 比求出该子树的博弈值所花费的时间少。

基于这个思想，最小窗口算法假设每个结点上第一个移动是最好的，并尝试不断证明接下来的移动是相对低劣的，直到证明这个假设是错误的为止。最小窗口算法对每个结点的第一个子树，使用完全的搜索窗口 $[\alpha, \beta]$ ，进行完全搜索 (exhaustively search)，得到该子树的博弈值 V ，对该结点的剩余子树，使用最小的搜索窗口 $[V, V + 1]$ 进行搜索，以快速找出该子树是否相对低劣，如果证明当前子树不是

低劣的,则需要重新搜索该子树,并修改博弈值 V 。这种最小的窗口又称为零窗口 (Null Window)。

6.5.3.2 历史启发算法(History Heuristic)

History Heuristic 是由 J. Schaeffer 提出的,它的基本思想是:维护每种不同移动的历史,记录移动对产生最佳博弈值方面有多成功,在搜索的时候,按照移动的历史来排位来对子树的搜索进行择序 (Kaindl & Shams, 1991)。

在搜索的过程中,每当找到一个好的走法,就将与该走法相对应的历史得分作为一个增量,一个多次搜索并确认为好的走法的历史记录就会很高,当搜索中间节点时,将走法根据其历史得分排列顺序,以获得较佳的排列顺序。这比采用基于棋类知识而节点排序的方法要容易得多。由于历史得分表随搜索而改变,对节点顺序的排列也会随之动态改变。

这一方法的实现有两个问题要解决:

首先如何将一个走法映射到历史得分的数组中? Schaeffer 指出可以用一个 12bit 数来表示走法在数组中的位置,其中 6bit 用来表示移动前的位置,6bit 用来表示移动后的位置。这样总共需要大约 2^{12} 个元素的数组来表达所有的组发。苏拉卡尔塔棋的棋盘为 $6*6$ 个格子,用一个 $36*36$ 个元素的二维数组,就可以表达所有的走法的起始及目标位置,我们可以使用起始和目标位置作为存取这个数组的两个下标。这样仅需要 1096B 即可。

另一个问题是如何确定历史得分的权值？当确定了一个好的走法时，要考虑：一是认为某分支的搜索层数越深，搜索得到的值也就越可靠；而是认为离根节点越近某一走法所对应的诸局面相似度越高，越往下分枝越多，相似程度越小。Schaeffer 给出了每发现一个好的走法就给出其历史得分增加 2^{depth} 的设计。

Schaeffer 同时也指出，历史启发对 Alpha-Beta 的增强，对于具体的棋类知识几乎无任何要求。任何棋类的类似搜索过程，只要能定义出合适的走法映射和增量因子就可以轻易的加入历史启发的增强。

6.5.4 评估函数

李淑琴等根据苏拉卡尔塔棋自身的特点，从棋子的数量、移动范围、攻击范围、子力攻击力、盘面分值和占弧价值 6 个方面对局面评估函数进行了研究，并对评估函数的参数进行了优化(李淑琴 et al., 2012)。

邱虹坤认为评估函数的准确度直接影响着算法的好坏，但是确定目前还没有有效的办法，主要依靠人工先验经验，结合实验测试与统计。在评估过程中，须要根据不同棋种的特点，有针对性地选择一些主要参数作为评估的依据，并加以适当的权重调整，得到相应的评估函数。其中，苏拉卡尔塔棋中较为重要的评估函数参数包括子力、机动性、控制力等(邱虹坤，2013)。

6.6 海克斯棋机器博弈

6.6.1 概述

海克斯棋 (Hex) 是一种二人对弈的战术棋盘游戏，它的起源可以追溯到 1942 年，由丹麦数学家 Piety Hein 发明；1948 年，该游

戏又被美国数学家 John Nash 重新独立发明，曾一度被称为“Nash Game”。海克斯棋的棋盘通常是一个六边形的网格，玩家通过放置棋子来尝试连接棋盘两对相对的边。游戏的目标是形成一条连接自己颜色边的不间断路径。由于海克斯棋的规则简单但策略深奥，它在计算复杂性理论中被认为是 PSPACE 完全的，这意味着它与某些已知的最难计算问题一样复杂。由于其独特的数学性质，海克斯棋也常被用于计算机科学和人工智能领域的研究，成为了测试和开发人工智能算法的理想平台。近年来，海克斯棋也成为了计算机博弈领域的研究对象，用过编写程序来模拟人类棋手的思维，进行人机对战，许多研究者和爱好者尝试使用不同的算法和技术来提升计算机在海克斯棋中的博弈能力。例如，基于深度强化学习的海克斯棋博弈算法研究，通过结合 Q Learning 与 CNN 卷积神经网络，使得计算机在博弈过程中能够选择更优的策略，并不断调整参数来提升海克斯棋的博弈能力。

海克斯棋的开源程序如 FutaHex2 也受到了广泛关注，这些程序通常基于蒙特卡洛树搜索算法，并提供了图形界面和核心库，使得玩家和开发者可以进行更深入的研究和学习。

6.6.2 海克斯棋规则

海克斯棋（Hex）是一种二人对弈的战术棋盘游戏，属于策略性游戏。游戏的目标是通过在棋盘上放置棋子来占领尽可能多的六边形点，以形成连接己方棋子的长链。海克斯棋的棋盘通常为 11×11 的方形网络（如图 6-13 所示），由红蓝双方交替落子，红方先手。游戏的胜负条件是看哪一方的棋子首先完成从棋盘一边到对边的连接。

- (1) 棋盘结构: 棋盘通常由 11×11 的六边形小格子组成。
- (2) 落子规则: 红蓝双方轮流在空位上放置棋子, 每次一方落子后, 对方在相邻的空位上落子, 直至棋盘被完全占领。
- (3) 胜负判定: 同种颜色的棋子相邻即认为它们相互连接。任意一方将自己的两条边界用自己颜色的棋子连接起来, 则对方获胜, 游戏结束。

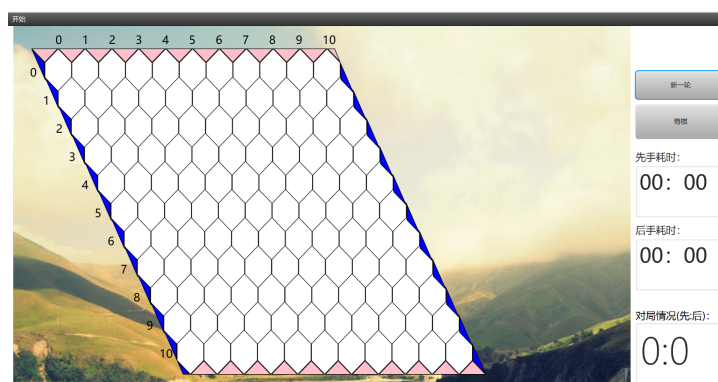


图 6-13 海克斯棋盘界面

海克斯棋是一个策略性很强的游戏, 其先手优势明显, 但并不意味着先手一定能胜。游戏的复杂性在于如何在棋盘上布局, 以及如何预测对手的下一步行动。因此, 掌握一定的策略和技巧是取胜的关键。

6.6.3 算法改进

张芃芃等以蒙特卡洛树搜索生成数据集训练卷积神经网络的方式, 使得模型能够在不断自我对弈的过程中, 修正自身选择动作的策略, 更新模型参数, 从而达到提升棋力的目的(张芃芃, 孟坤, & 杨震栋, 2020)。徐志凡等采用强化学习中 Q 学习的思路, 对开局库进行训练制作, 训练完成的 Q 表就是开局库, 面对开局局面只需选择当前局面下 Q 值最大的动作。相较于传统的人工经验和 MCTS 方法, Q 学习制作开局库的方式既节省了大量的计算时间, 又使得开局库中走

法更加精确(徐志凡, 李媛, 王静文, 李卓轩, & 曹一丁)。张志礼等针对海克斯棋自身的特点, 结合基尔霍夫电流定律, 设计并实现了基于电阻电路评估策略以及分阶段搜索的海克斯棋博弈算法(徐志凡 et al.)。

6.7 五子棋机器博弈

6.7.1 概述

五子棋是一种娱乐性强、规则简单易学、流行性广的计算机博弈棋类, 长期以来深受大家的喜爱(严小卫, 1999; 张明亮, 吴俊, & 李凡长, 2012)。

6.7.2 五子棋规则

6.7.2.1 棋盘与棋子

中国大学生计算机博弈大赛采用 15×15 围棋的棋盘, 黑白两种围棋棋子(中国人工智能学会机器博弈专业委员会, 2014a)。

6.7.2.2 行棋规则

① 先后手的确定。

由大赛裁判组在赛前通过抽签确定或对局前猜先的方式确定。

② 开局

中国大学生计算机博弈大赛采用指定开局模式。指定开局黑方决定了前三个棋子落于何处, 其中包括两个黑子和一个白子。黑方应同时给出第五手需要的打点数量。采用指定开局模式的比赛均采用斜指或直指开局(26种), 黑方第一子应落在天元处(黑1)。黑方还决定了白方的第一子的落点(白2)。黑方的第二子(黑3), 应落在围绕天元点5线 \times 5线而形成的以天元为正中的由交叉点组成的区域

内。同时在后续的对局中执行三手交换、五手 N 打及禁手规则。

③ 三手交换

在采用指定开局的对局中，在黑 3 之后，白方在应白 4 之前，可选择黑棋或白棋，每盘棋只有一次选择机会，如提出交换黑、白方，则黑方必须同意交换。

④ 五手 N 打

黑方在指定开局的同时要给出本局盘面黑 5 时所需的打点数量，此后无论对局者谁执黑棋，都需要在落第五手时按照要求的打点数量，在盘面上的空白交叉点上放置相应数量且位置不同形的黑子，白方只能在这些黑子中留下一个黑子作为黑 5。

⑤ 禁手

对局中如果使用三三禁手、四四禁手、长连禁手，将被判负。

6.7.2.3 终局胜负判定

① 最先在棋盘上形成五连的一方为胜。白棋长连视同五连。

② 黑方出现禁手，则判白方胜。如白方在黑方出现禁手后，未立即指出而又落下一白子，则黑方禁手不再成立。

③ 若黑方走出长连禁手，白方只要是在终局前指出此禁手，判白方胜。

④ 黑方五连与禁手同时形成，禁手失效，黑方胜。

⑤ 对局中，一方出现下列情况之一判负：比赛对局中移子或棋局散乱、超过规定时限、人为辅助计算、主动停止计时。

⑥ 对局中出现下列情况之一，判和棋：对局双方同一回合均放

弃行棋权、全盘下满，且无胜局出现、双方比赛同时超时。

6.7.3 改进的五子棋博弈算法

董慧颖等在 Alpha-Beta 剪枝算法中引入迭代加深以及局部搜索方法，提高程序棋技。在此基础上使用 Monte Carlo 方法和深度学习方法结合的方式来提高下棋技巧(董慧颖 & 王杨, 2017)。张明亮等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(张明亮 et al., 2012)。刘溜等针对传统蒙特卡洛树搜索算法存在“难以在节点的探索和利用之间做出平衡；难以聚焦重要搜索分支”等问题，提出使用策略价值网络完成棋局评估与落子着法生成，将策略价值网络与蒙特卡洛树搜索相结合。策略价值网络指导搜索树的展开，搜索结果用以持续更新网络参数，形成一种自博弈方法，在多轮自博弈中实现算法的迭代优化(刘溜 et al., 2022)。

6.7.4 机器人五子棋机器博弈

2024 年中国大学生计算机博弈大赛暨第十八届中国计算机博弈锦标赛新增机器人五子棋软、硬结合项目。机器人五子棋项目行棋规则同五子棋机器博弈项目，不同之处在于该博弈项目是基于机器人五子棋博弈平台而开展的，该平台提供了博弈 AI 算法模块的二次开发接口，如图 6-14 所示。



图 6-14 机器人五子棋博弈平台

机器人五子棋博弈平台采用两关节平面型桌面机器人结构，由机器人本体、运动控制、视觉识别和人机界面四个部分组成，配备自动送子机构，通过视觉识别精准棋盘及棋子位置，是一款集图像分析算法、AI 智能博弈和关节柔性控制于一身的人工智能对弈机器人。产品定位陪练、助教和竞赛三大功能，设计传承寓教于乐的棋类元素，支持人机对弈和双机对弈，同时设置有本地博弈算法和自定义博弈算法，提供 python 和 C++接口，可以通过训练对接高阶算法让机器人更加智能。

6.7.4.1 机器人五子棋博弈平台的特点

智能机器人五子棋博弈平台综合使用了嵌入式、视觉处理等先进技术开发而成，具有以下特点：

- ① **先进视觉识别系统。**能在不同光照环境下，识别棋盘及棋子位置。
- ② **AI 算法系统。**通过自主研发的神经网络算法，对棋局做出对弈决策。
- ③ **智能操作系统。**采用仿人柔性机械手臂,安全可靠；自动送子机构供子，无需人工频繁操作，支持双机对弈。

6.7.4.2 应用场景

机器人五子棋博弈平台可以应用于以下场景：

① **陪练**。陪伴棋手，自主实现对弈练习。

② **助教**。实现自主下棋，在棋盘上演示各种变化，配合教练完成各项教学任务。

③ **比赛**。根据竞赛的要求，配备标准化/模块化的系统接口，能够实现不同后台对弈策略的快速、稳定的系统连接。

6.8 中国象棋机器博弈

6.8.1 概述

中国象棋作为我国传统的棋类游戏(徐心和 & 王骄, 2006)，每个棋子都各有特殊的行棋规则，其空间复杂度是分析中国象棋博弈难度的重要指标(魏印福 & 李舟军, 2019)，难度不亚于国际象棋(付强 & 陈焕文, 2007)。此外，中国象棋也是一种完全知识博弈，即行棋双方在任意状态下都完全清楚每个棋子的现状。而且中国象棋对弈规则简单明了，胜、负、和的评定标准单一，属于二人零和博弈(即博弈双方的获利和损失相加和总是为零)，不会有任何偶然性的结局。其规则可参考中国大学生计算机博弈大赛官方网站(中国人工智能学会机器博弈专业委员会, 2014b)。

中国象棋计算机博弈可以分解为 4 个主要部分：搜索引擎、走法生成模块、评估函数和开局库(王. 王骄, 罗艳红, 徐心和, 2005)。近年来，其研究工作主要聚焦于提高博弈程序的搜索效率以及优化局面评估函数两个方面，从而提高计算机的象棋博弈水平(朱舟 & 闵华松)。

6.8.2 估值方法

周玮等针对静态估值方法不能体现双方各自实力的变化趋势,实现博弈者战略意图定向搜索的实际,提出了基于对弈局势的二次估值方法,建立了以“双方棋力合理消长”为目标的非线性 0-1 规划模型,并引用遗传算法求解出“局势因子”,得出了搜索路径随局势因子改变的结论(周玮, 张贻, & 周静怡, 2006)。

6.8.3 搜索算法

朱舟等针对蒙特卡洛树搜索算法(Monte Carlo Tree Search, 简称 MCTS)收敛速度过慢,且在博弈过程中关键节点会出现信息丢失等问题,以中国象棋为载体,构建适用于中国象棋博弈系统的策略价值网络,提出了一种基于统计数据的并行蒙特卡洛树搜索算法(Parallel Monte Carlo Tree Search Based on Statistics, 简称 SPMCTS)。将并行化的重点设置在 MCTS 四个步骤中最耗时的扩展和模拟步骤,有效避免了算法执行过程中的等待时差。并且引入一组新统计数据,这些数据用于在 MCTS 的选择步骤中修改节点的选择策略,保证在进行节点选择时获取和利用更多的可用信息,缓解信息丢失对精度造成的影响(朱舟 & 闵华松)。

苏攀等针对中国象棋搜索空间庞大,搜索效率低下问题,提出一种采用分类器模拟象棋策略的方法。该方法使用不平衡学习技术对专家棋谱转换成的训练集作处理,并使用处理后的训练集训练分类器。最后,在系统中使用“招法生成器+分类器”的方法实现对弈(苏攀 et al., 2011)。

6.9 藏族久棋博弈

6.9.1 藏族久棋简介

藏族久棋是一类规则复杂且独特的民间棋类运动，是在西北地区“方棋”的基础上，融合汉、藏、蒙古、回族等多民族文化创新发展而形成的，具有数十个变种，兼具竞技性与趣味性(刘强，2017；马少龙 & 尚涛，2017)。藏族久棋被列入青海省、四川省、西藏自治区非物质文化遗产以及国家级非物质文化遗产。

藏族久棋的常用棋盘为 14 路，棋盘中间的棋格有一条对角斜线。棋盘(图 6-15(a))由 14 条等距垂直相交的平行线组成，共 196 个交叉点，是棋子的落子点，棋子分为黑白两种颜色，各 98 枚。久棋采用猜子定先后以及空枰开局的方法。

猜子定先后：“久棋”开局先后，应以公平公正为宜，一般以裁判员任意抓起一把棋子，由双方猜单双的形式决定，猜子时不得双方同时猜单或双。猜准者优先选择行棋先后或棋子颜色。

空枰开局：对弈双方各执一色棋子。

藏族久棋对弈包括布局、走子、飞子阶段，每个阶段规则独特且相互影响。

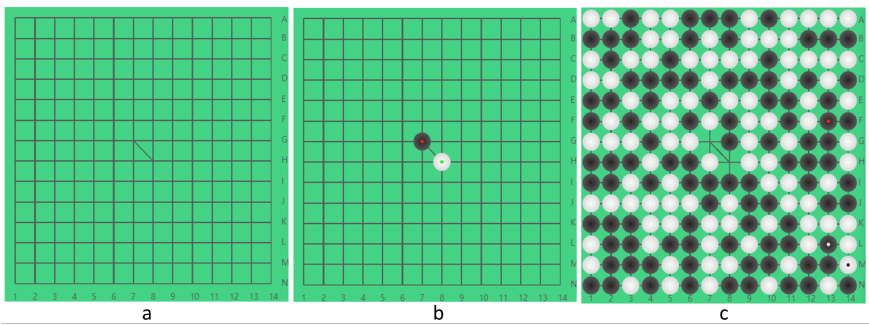


图 6-15 藏族久棋棋盘以及布局、走子开局

6.9.1.1 局阶段行棋规则

在布局阶段，按白先黑后的顺序交替放置棋子在棋盘上，每次轮流放一个棋子，且双方棋手的第一手棋子必须放在在对角线上某个端点位置上，如图 6-15(b)所示，直到棋盘上每个交叉点都布满棋子，布局阶段结束。

6.9.1.2 走子阶段行棋规则

将棋盘中央对角线两端的棋子移走，双方玩家交换游戏顺序，黑先白后，进入走子阶段，如图 6-15(c)所示。在走子阶段，有“移动”、“跳吃”和“成方吃子”三种走棋规则。

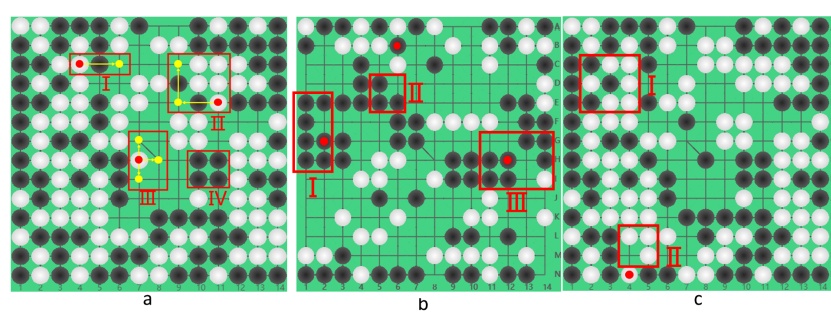


图 6-16 走子方式和藏族久棋阵型

(1)移动。将己方棋子移动到相邻的空交叉点。如图 6-16(a)中的III所示，白子 H7 在水平或垂直方向上移动。

(2)跳吃。己方棋子与敌方棋子相邻，且在相邻的同一水平线或者竖直线上敌方棋子与空的交叉点依次出现，则己方棋子跳到相邻的水平线或者竖直线上的某个空交叉点，并且吃掉路径上的敌方棋子。跳吃由单步跳吃和多步跳吃，图 6-16(a)中的 I 为单步跳吃，白子从 C4 跳到 C6，吃掉位于 C5 处的黑子。图 6-16(a) II 中白子 E11 连续跳两步，依次吃掉 E10 以及 D9 两处的黑子，落在 C9 处。

(3)成方吃子。在移动或跳吃之后，如果己方四个相邻的棋子形成一个正方形，也就是“方”，则己方可以吃掉对手棋盘中任何位置的一颗棋子。如图 6-16 (b) 中 II 所示，黑子 B6 跳到 D6，吃掉白子 C6 后，位于 D6、E6、E5、D5 处的四个黑子形成一个方，此时，黑方还能吃掉棋盘上任意位置的一个白子，叫方吃，方吃的威力很大，可以破坏低手的布局。

6.9.1.3 飞子阶段行棋规则

进入对弈后期，如果一方落后，棋子数等于或少于 14 枚时，少子方就进入飞子阶段，此时少子方移动棋子不受一步一格的限制，棋子可以落至任意空棋位上，阻止对方“成方”、“成阵型”或者走成“飞子阵型”，则可反败为胜。而没有成为飞子的一方则受到移动步数限制。飞子阶段双方均不允许只跳吃一个子，可以跳吃两子或两子以上。飞子阶段可以做成不同类型的久后，吃一颗子或多颗子。

6.9.1.4 胜负判定

(1)当某方的棋子只剩下 3 颗子时，判该方为负。

(2)当一方形成两个或两个以上稳定褡裢形状，而对方没有棋门时，判对方负。

(3)一方的棋子虽然较多，但都不能进行有效移动成方，则判负。

(4)超出比赛单方用时限制，判负。

6.9.1.5 藏族久棋中的褡裢棋型

褡裢是藏族久棋中的非常具有威力的阵型褡裢是由七枚棋子组成的一种棋型，又分为平口褡裢和斜口褡裢两种样式。

图 6-16(b) 中, I 为平口褡裢, G2 处的黑子可以移动到 F2 处成方, 吃掉白方任意一子。白子走一步棋后, 轮到黑子走棋时, 黑子可以从 F2 处移动到 G2 处成方, 继续吃掉白方任意一子。来回移动一子, 黑方可以通过成方吃掉白方任意一子, 威力很大。在图中 4×2 的矩形区域内, 将其按顺时针旋转可得到横向的平口褡裢。

图 6-16(b) 中, III 为横向的斜口褡裢, 分布在一个 3×4 的矩形区域内, 通过旋转可以得到纵向的斜口褡裢。此外, 还可以将图 6-16(b) 中的褡裢 I、III 进行镜像翻转, 得到褡裢阵型。

6.9.2 藏族久棋博弈方法

深度强化学习技术由 DeepMind 公司于 2013 年首次应用于 Atari 游戏 (Volodymyr et al., 2015), 广泛应用于完备或者非完备信息博弈游戏中, 训练的 AI 大部分达到甚至超过了人类顶尖高手 (J. Li, Koyamada, Ye, Liu, & Hon, 2020; David Silver et al., 2016; David Silver et al., 2018; D. Silver, Schrittwieser, Simonyan, & al, 2017; Zha et al., 2021)。目前先进的深度强化学习方法和技术在藏族久棋博弈中被广泛采用 (Funge & David, 2009; Xiali Li, Lv, Wu, Zhao, & Xu, 2020; Pomerol, 1997; 李霞丽, 陈彦东, 杨子熠, 张焱垠, & 吴立成, 2022; 梁凯, 2022; 吕征宇, 2020), 但是由于缺乏大量的高质量的数据集, 在普通实验室能够承受的算力资源条件下, 藏族久棋智能体的水平与人类高手的水平存在较大差距。藏族久棋博弈方法包括基于知识的方法 (Xiali Li, Wang, Lv, Li, & Wu, 2018; Su, Li, Li, Chen, & Fan, 2024; 李霞丽 et al., 2018; 沈

强望, 丁濛, 杜文涛, & 赵文龙, 2021; 王世尧, 2023; 张小川, 刘溜, 陈龙, & 涂飞, 2021)、小样本机器学习方法(邓颂庭, 2017)、时间差分算法(Time-Difference, TD)(Xiali Li, Lv, Wang, Wei, & Wu, 2019; 王松, 2018)、深度强化学习算法(X. Li et al., 2020; 李霞丽 et al., 2022; 梁凯, 2022; 吕征宇, 2020)等。

6.9.2.1 基于知识的方法

李霞丽等人(Xiali Li et al., 2018; 李霞丽 et al., 2018)从藏族久棋全国比赛中提取棋型,并通过基于矩阵的匹配方法进行模式识别,根据专家知识设计不同棋型的进攻和防御策略。但由于数据样本少以及专家知识匮乏,所开发的软件棋力仅略高于人类初学者的水平。

张小川等人(张小川 et al., 2021)构建了久棋博弈智能体的知识库,重点提出了规则库,开局库的构造方法.针对藏族久棋规则的特殊性,面向久棋特有的3个阶段,基于量化思想,建立了分段评估体系.以此方法开发久棋博弈程序,参加了全国计算机博弈大赛并获得优异成绩,验证了方法的可行性和有效性。

丁濛等人(沈强望 et al., 2021)采用基于棋型匹配的分段评估方法,针对藏族久棋布局、行棋、飞子三个阶段的行棋规则,构造了不同阶段的局面评估方法。

王世尧等人(王世尧, 2023)提出的基于专家知识的久棋博弈模型通过二维数组搜索遍历棋盘当前状态,针对走子阶段设计了偏攻击策略和偏防守策略两种不同的对弈策略,该模型参加了在西藏拉萨市

举办的2021年“岷山杯”藏棋全国公开赛，与对战两位一段棋手的胜率达到了74%和68%，而对战五段棋手的胜率仅有14%，综合来看其棋力高于人类一段棋手，但无法达到久棋高手的水平。

苏琛等人(Su et al., 2024)考虑到藏族久棋对弈的布局、走子、飞子阶段规则不同且空间复杂度不同，提出了久棋的三阶段嵌套博弈算法以更好地提升下棋水平。布局阶段采用基于棋型的 Alpha-Beta 剪枝算法。走子阶段细分为嵌套的三个子阶段，第一子阶段采用基于棋型评估的 Alpha-Beta 剪枝算法，第二子阶段采用复杂度更高的棋型评估，第三子阶段采用基于棋型的局部 MCTS 算法。飞子阶段采用棋型评估算法。使用该算法开发了 14 路久棋 AI，在 2023 年中国大学生计算机博弈大赛暨博弈锦标赛中取得优异成绩。

6.9.2.2 小样本机器学习方法

邓颂庭等人(邓颂庭, 2017)提出的小样本机器学习方法采用小样本数据推理能力强的贝叶斯网络，一定程度上解决了藏族久棋棋谱数据较少的问题。但由于基础棋阵的数量较少，且相互间的拓扑结构完全依赖专家知识，以及网络结点间的转换概率值设定缺少行棋间的线性关系，棋力受到了限制。

6.9.2.3 时间差分算法

王松等人(Xiali Li et al., 2019; 王松, 2018)将时间差分算法用于藏族久棋，通过优化上限置信区间算法(Upper Confidence Bound Apply to Tree, UCT)使其适应藏族久棋，并引用时间差分算法进行策略更新，解决了以专家知识为基础设计状态转移策略带来的

局限性。但因为模型的算法效率较低以及计算资源不足，藏族久棋博弈系统的棋力处于初级水平。

6.9.2.4 深度强化学习方法

吕征宇等人(X. Li et al., 2020; 吕征宇, 2020)提出结合Q-learning和Sarsa(λ)的混合通用深度强化学习模型,通过混合时间差分算法结合深度神经网络,从零开始进行自对弈训练。与仅使用Q-learning和仅使用Sarsa(λ)的程序进行对弈,混合深度强化学习模型的胜率分别达到了60%和70%。该算法减少了搜索次数,提高了学习效率。但由于缺乏训练数据以及硬件资源不足,棋力提升不明显。

陈彦东等人(李霞丽 et al., 2022)提出了藏族久棋的两阶段博弈算法,为藏族久棋的布局阶段设计了基于卷积神经网络和蒙特卡洛树搜索的自对弈算法,通过卷积神经网络指导蒙特卡洛树进行搜索,训练出最优模型并生成质量更高的着法;为战斗阶段设计了基于领域知识的 Alpha-Beta 剪枝算法。两阶段算法的博弈智能体程序在一定程度上具备了“学习”和“思考”的能力,棋力得到了提升。

梁凯(梁凯, 2022)设计了一种预测藏久棋着法和局势的策略-价值网络模型,通过自对弈产生数据并用于网络的训练。在不引入专家知识的前提下,无法完全依靠神经网络自对弈达到理想的效果。

6.9.3 藏族久棋线上对弈平台

见诸文献的藏族久棋线上对弈平台包括网络联机对弈平台(X. Li, Zhang, Wu, Chen, & Liu, 2023)、久棋攻略小程序(Chen, Wu, Yan, & al, 2023),久棋星宇平台(Xu, Li, & Zhang, 2024)等。

张焱垠等人开发的藏族久棋网络对弈平台(X. Li et al., 2023), 基于 Cocos Creator 游戏引擎进行开发, 采用场景搭配 TypeScript 脚本控制, 使用帧同步通讯方式配合华为游戏联机对战引擎相关 API, 使平台具有较高的鲁棒性。系统为 B/S 架构, 操作简单明了, 用户仅需鼠标或手指、键盘便可完成所有操作。该平台作为 2022 和 2023 年中国大学生计算机博弈大赛暨中国计算机博弈锦标赛藏族久棋线上比赛平台, 顺利完成了比赛。

陈彦东等人(Chen et al., 2023)开发的“久棋攻略”小程序, 能够在手机端实现久棋的线上对弈。“久棋攻略”小程序采用云开发技术和 JavaScript 语言开发, 实现了在线对弈、在线观战、棋谱采集、棋局复盘、高手棋谱展示、久棋资讯等功能, 截止到 2024 年 5 月份, 已经积累了 3724 名用户。“久棋攻略”目前小程序已经在西藏、青海、四川、北京等地使用, 根据用户反馈的问题和修改建议更新到了第五版(V3.0.1)。

徐淦等人(Xu et al., 2024)开发了久棋星宇 AI 自动化对弈平台, 具有 AI 对战、人人对战、自动生成并存储棋谱、胜负判断、棋局复原等功能。该平台作为 2023 年和 2024 年中国机器博弈锦标赛久棋项目的 AI 自动对战平台, 顺利完成了久棋 AI 比赛。久棋 AI 开发者只需要遵循一定的接口规范, 就可以方便地接入该平台, 实现 AI 程序的自动对弈。

6.10 国际跳棋

6.10.1 概述

国际跳棋(Draught)又名西洋跳棋,是一种比国际象棋历史还有悠久的双人棋盘类游戏。全球有 4000 万国际跳棋参与者,遍布 57 个国家、五大洲,是世界智力运动会的比赛项目之一。

6.10.2 基本规则简介

棋盘为正方形,由 10×10 共 100 个黑白(或深浅)格构成,棋子只能放在黑色格子中且每位棋手的左下角必须是黑格。对局时双方在棋盘的黑格内走动棋子,通过喂子和吃子,以最终吃掉另一方所有棋子或逼迫对方无法走棋为获胜条件。图 6-17 表示了一个开局时的棋盘状态。开局时所有棋子只能在深色格子中,双方左下角必须是深色格子。

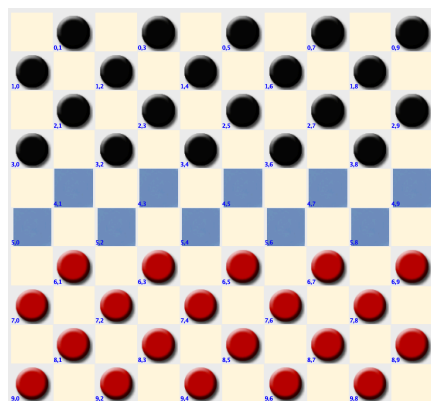


图 6-17 国际跳棋开局状态

6.10.2.1 行棋和升变

各方的棋子分为普通棋子和王棋两种。行棋时,双方轮流走棋,执白(红)方总是先行。每回合,普通棋子只能在其对角线方向、前方(指向对手的方向)的黑格内移动一格,不能后退。当某个己方棋子到达对方底线且停留在此位置时,会升变为王棋。王棋可以延对角线在

黑色格子中前进或后退任意距离。

6.10.2.2 吃子

行棋中，当对角线方向的相邻格中有敌方棋子，并且之后一格是空的，则玩家必须跳过敌方棋子并将其吃掉。能吃子时玩家必须吃子。当有多枚棋子可以吃时，玩家可以选择吃哪一个。如果选择的走法可以连续吃子，必须吃掉所有能连续吃掉的子，且必须选择能吃掉对方棋子最多的那种方式。被吃的子立即移出场外，并且玩家在一回合中不能跳过同一个棋子两次。当一方的王棋在相同的对角线上遇到相邻或不相邻的对方棋子时，且对方棋子后面有一个或多个空格，那么这枚王棋即可跳过对方棋子并占据它后面的任意一个空格。

6.10.2.3 胜利

当一方没有棋子或是不能移动时，对方获得胜利。若棋局进行到最后，双方均无法战胜对方，即为和棋。

6.10.3 国际跳棋机器博弈关键技术

一直以来有很多数学家、计算机学家试图解决国际跳棋的博弈问题。2007 年 7 月，加拿大计算机科学家宣布了对于 8×8 国际跳棋(64 格)的博弈程序 Chinook。该程序可以找到 64 格国际跳棋的最佳走法，且若双方都按这最佳走法下棋，那么棋局将以和局收场(Jonathan Schaeffer, 2010)。但对于 100 格的国际跳棋，还没有这样的程序出现。

由于规则相对简单，因此国际跳棋的博弈问题主要基于极大极小搜索算法及其优化 Alpha-Beta 剪枝算法。该类方法的本质就是通过深度优先的方式穷举某一棋局状态下所有的走法，从中找到最好的一

个。但由于状态空间成指数级别增长，搜索过程往往不可能达到终止状态，因此该类方法需要一个合理且准确的评估函数来提前结束搜索过程。

由此，一段时期内，国际跳棋的机器博弈问题主要优化方向有：

1) 如何提高搜索算法的运行速度。常用的技术有位棋盘，并行计算，特殊 Hash 函数的设计等；

2) 如何减少搜索的深度。主要的研究方向是，如何设计合理准确的评估函数，枚举不同等级(剩子数目)的残局库等。

DeepMind 公司在 2017 年提出了 Alpha-Zero 框架(D. Silver et al., 2017)。该框架用于训练一个不依赖于人类先验知识的围棋博弈程序，极大地推动了围棋技术的发生。考虑到同为对称知识的棋盘类游戏，可以使用 Alpha-Zero 框架进行国际跳棋 100 格的博弈问题训练。

Alpha-Zero 本质上是一个结合了蒙特卡洛搜索和卷积神经网络(CNN)的增强学习模型。采用该模型的国际跳棋 100 格的博弈方案如图 2 所示，主要分为自博弈阶段和 CNN 训练阶段。其中 CNN 部分利用自博弈阶段得到的棋谱，训练并更新价值和策略网络，其实质是一种评估函数的设计和提取。自博弈阶段，将 CNN 学习到的价值和策略网络结合到蒙特卡洛搜索中，通过不同版本的价值和策略网络进行程序与程序的自对弈，产生新的对弈棋谱，重复该过程完成博弈系统的训练。

6.11 不围棋

6.11.1 概述

不围棋 (NoGo) 是一种源自传统围棋的策略性棋类游戏, 但其规则和目标与围棋截然相反。玩家需要避免“吃”掉对方的棋子, 否则将立即判负。这种逆向思维的游戏方式为玩家带来了全新的挑战, 也引起了人工智能领域的关注。

不围棋使用 9×9 的围棋棋盘, 黑方先行, 双方轮流落子, 棋子颜色为黑白两色。游戏的核心规则是禁止吃子, 若某一步棋导致对方棋子被提走, 落子方将立即失败。玩家不能通过自杀或空手弃权进行回合, 且棋子一旦落下便不可移动。胜负的判定方式是, 如果一方无法在规则允许下落子则失败, 或因违规吃子而被判负。比赛中通常采用胜负制而不计算具体得分, 若在限定时间内未分出胜负, 则可能根据棋局复杂度或剩余空间进行裁定。

围棋侧重进攻和防守策略, 而不围棋则要求玩家谨慎布局, 避免触发违规。不围棋要求玩家采用逆向思维, 侧重于避免风险, 而非追求利益最大化。正因为这一独特性, 不围棋成为了研究新型人工智能算法的理想平台, 有助于在受限条件下探索最优策略。

6.11.2 不围棋机器博弈主要技术

6.11.2.1 蒙特卡洛树搜索

蒙特卡洛树搜索 (Monte Carlo Tree Search, MCTS) 是广泛应用于策略游戏的一种搜索算法, 其搜索过程如图 6-18 所示。特别适用于不围棋这种具有深度和宽度决策树的复杂博弈环境。在不围棋中, MCTS 通过模拟大量不同的棋局走向, 协助人工智能预测每个可

能棋步的后续局势(Guo & Chen, 2018)。采用 MCTS (详见第 4.7 节) 作为基础搜索框架。在算法的模拟阶段, 必须对每一步落子进行合法性检查, 严格规避任何会导致提走对方棋子的落点, 从而确保模拟对局符合不围棋的规则目标。利用遗传算法的全局优化能力, 结合模糊逻辑处理局面评估中的不确定性, 共同优化评估函数参数, 提升 AI 在模糊局面下的决策鲁棒性。结合卷积神经网络 (CNN) 进行局面特征提取, 并通过自我对弈的强化学习来提升策略质量, 使 AI 学会如何在不触发吃子的前提下最大化获胜机会。

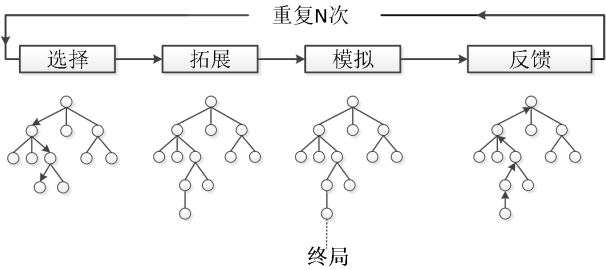


图 6-18 蒙特卡洛树搜索过程

在不围棋中, 由于目标是避免吃子, MCTS 需要特殊的调整以避免传统围棋中的进攻性策略。因此, 算法在模拟过程中必须特别关注棋盘中的“禁区”(即可能导致被吃的区域), 以规避不合规的落子(Guo & Chen, 2018)。MCTS 的主要优势在于其通用性强, 不需要预先定义大量规则或策略, 而是通过模拟和胜率评估来自动选择最优动作。然而, 由于不围棋规则的特殊性, MCTS 在这种环境下的计算复杂度较高, 往往需要与其他技术结合进行优化。

6.11.2.2 遗传算法与模糊逻辑结合

遗传算法 (Genetic Algorithm) 是一种基于生物进化理论的搜索与优化算法, 通过模拟自然选择和进化过程, 不断生成、评估和筛

选“个体”（在不围棋中是指不同的棋局策略），从而逐渐优化对局方案。在不围棋的博弈中，遗传算法常与模糊逻辑相结合，显著增强了人工智能在策略选择上的能力(Lee et al., 2012)。

遗传算法通过生成一组候选策略（称为“种群”），并基于“适应度”函数来评估每个策略的优劣。最优策略会被保留，并通过“交叉”和“变异”操作生成新的策略，从而不断提高对局能力。

模糊逻辑使人工智能能够在模糊或不确定的条件下进行决策，这与不围棋复杂多变的局面特性相匹配。通过动态调整模糊推理模型，遗传算法能够有效应对棋局的不同阶段，从而优化人工智能的判断和决策(Lee et al., 2012)。这种将遗传算法与模糊逻辑相结合的方式为不围棋 AI 的开发提供了一种新的视角，尤其适用于需要在高度复杂局面中做出实时决策的场景。

6.11.2.3 强化学习

强化学习（Reinforcement Learning, RL）是一种通过与环境互动自主学习策略的机器学习方法。在不围棋中，人工智能通过大量的自我对弈来积累经验，并不断改进其决策能力(Bo, Li, Geng, & Tong, 2020)。

强化学习的核心机制是基于试错过程进行逐步学习。在游戏的初期，人工智能可能随机选择棋步，但随着对局次数的增加，AI 能够逐渐识别出能够有效避免违规的策略，并不断优化(Bo et al., 2020)。

在不围棋的强化学习研究中，通常结合使用卷积神经网络（CNN）

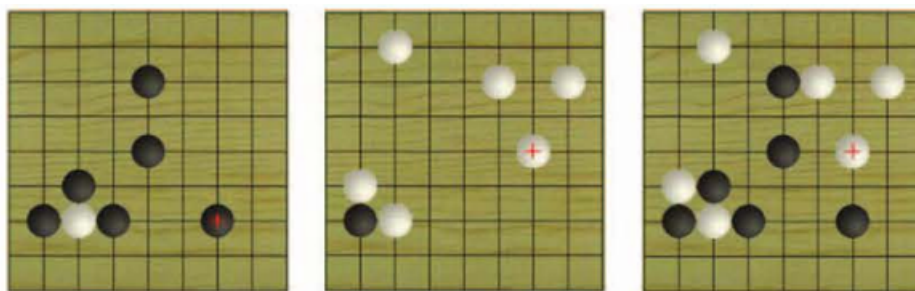
和蒙特卡洛树搜索（MCTS）。卷积神经网络用于评估棋盘的局势，提取棋局中的关键特征，而 MCTS 则利用这些特征进行更为精确的模拟和搜索 (Bo et al., 2020)。通过强化学习，AI 能够逐渐达到自我优化的状态，而不依赖于人类提供的大量棋谱或预设规则。这种自我对弈的方法使得 AI 不断提高其棋力，并能够灵活适应不同对手的策略。

6.12 幻影围棋

6.12.1 概述

幻影围棋（Phantom go）是一项欧洲的棋类游戏，其采用 9×9 的围棋棋盘和围棋规则，但是又在此基础上加入了信息不完全的限制，属于不完全信息博弈 (刘子建，胡德宏，赵文浩，& 吴佳明，2022)。国际计算机奥林匹克从 2007 年开始加入幻影围棋项目，由中国人工智能学会举办的中国计算机博弈锦标赛于 2012 年加入计算机幻影围棋比赛。

下幻影围棋时，双方均无法获得对方落子的位置，各方棋盘均为不完备信息的棋盘，只有裁判能看见双方落子，即拥有双方棋盘并集而成的完备信息对弈棋盘，进而给予双方操作是否合法或者是否提子的指令来进行比赛 (李飞 et al., 2016)。如图 6-19 所示，左边是黑方棋盘，中间是白方棋盘，右边是裁判棋盘，黑方棋盘中的白子是通过裁判的返回信息判断出的部分棋子落子位置，信息是不完全的。这种不完全的信息，既增加了对弈的难度，同时也增加了趣味性和机会性。



(a) 黑方棋盘 (b) 白方棋盘 (c) 裁判棋盘

图 6-19 幻影围棋的棋盘设置

6.12.2 幻影围棋博弈主要技术

博弈搜索是计算机博弈的重要组成部分，搜索相关算法也是计算机博弈系统智能化的重要方法。要实现较好的对战效果就必须加载智能化的机器走子搜索策略。尽管 Alpha-Beta 剪枝搜索、蒙特卡洛树搜索和 UCB 策略等围棋博弈中常用的搜索策略可以应用于幻影围棋，但是因为缺少足够的信息，传统搜索算法存在搜索精度太低、搜索时间太长的缺陷(张晓倩，黄鸿，& 程子豪，2019)。相比之下，将不同算法结合起来，根据棋盘状态不同（或者棋局阶段不同）采用不同的搜索算法，可以大幅增加棋局胜率。

6.12.2.1 信念状态-蒙特卡洛树搜索结合地域剪枝策略

信念状态-蒙特卡洛树的搜索算法包括 5 个过程：采样、选择、拓展、模拟和反馈。对标准 MCTS（详见第 4.7 节）进行关键改进，将其中的“状态”从单一棋盘扩展为“信念状态”，即所有可能真实局面的一个概率分布。搜索树中的每个节点都代表一个信念状态，通过采样、模拟和回溯来更新不同可能局面的概率，从而在信息不完全的情况下做出最优决策。在原始蒙特卡洛的规划过程中加入返回地域信息的过程，通过大量的模拟可以较准确的评估盘面的地域信息，最

后根据设定的条件确定剪枝范围，实现效率提升(张晓倩 et al., 2019)。由于掌握的对手棋子信息有限，在对局进行到中盘时才启用剪枝，效果更好。

6.12.2.2 蒙特卡洛结合 Alpha-Beta 剪枝算法

目前蒙特卡洛算法与 Alpha-Beta 剪枝算法有两种结合方式，第一种结合方式是博弈进行到中局，双方都通过裁判返回的命令获取了部分对方棋子的信息，若已知的对方棋子数与对方棋盘棋子总数的差值(DIFF)，小于等于某个阈值时，可以采用 Alpha-Beta 剪枝算法进行完全信博弈搜索。当大于该阈值时，系统又启用蒙特卡洛算法进行随机模拟来寻找最佳招法(胡强，谢显中，& 张小川，2014)。

第二种结合方式是将整个棋局分为开局、中盘和收尾三个阶段。每个阶段设计和使用不同的策略与应对方法，将算法与局势结合起来增大胜率。在开局阶段通过定式走法进行开局布局，抢占关键位置，占据有利局面，增加己方的优势，并且减少了时间的消耗。中盘阶段采用蒙特卡洛算法对棋局进行大量的模拟实验，可以利用并行技术来降低时间以换取更多的模拟次数。收尾阶段采用 Alpha-Beta 剪枝算法来选择棋子的落子位置，更大程度上得到分数(刘子建 et al., 2022)，具体的棋局招法搜索流程如图 6-2020 所示。

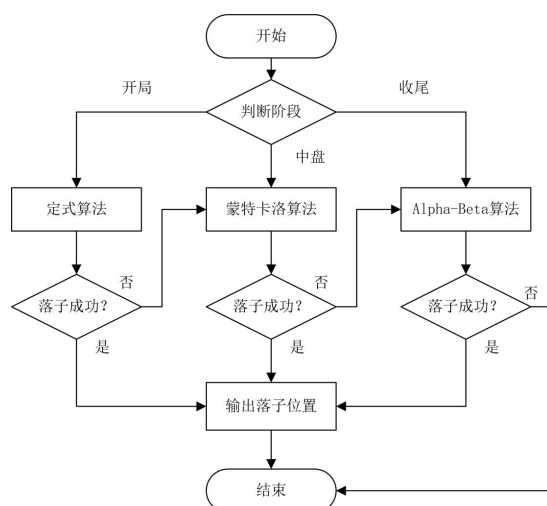


图 6-20 蒙特卡洛与 Alpha-Beta 算法结合的棋局招法搜索流程

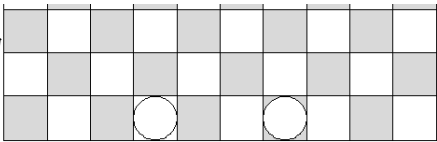
6.13 亚马逊棋

6.13.1 概述

亚马逊棋属于两人对弈确定性的棋盘类游戏，属于零和游戏(游戏双方在游戏结束时肯定会分出胜负)，于 1988 年由阿根廷的 Walter Zamkaskas 发明，于 1992 年在西班牙的游戏杂志上发布了相应的游戏规则，于 1993 年，由 Michael Keller 推荐被引入到名为“kNight Of The Square Table”的邮政游戏俱乐部，从此之后被逐步得到推广，1994 年由阿根廷和美国各出一个队进行比赛，比赛共为 6 场，比赛结果为 3 比 3 平手，在 1998 年，由日本的静冈大学的计算机博弈研究学院的 Hiroyuki Iida 发起了亚马逊棋的计算机挑战赛，当时的获胜软件名为 Yamazon，程序的设计者是 Hiroshi Yamashita，2000 年和 2001 年的国际计算机奥林匹克锦标赛都开展了亚马逊棋的比赛，亚马逊棋也由此推向全世界，亚马逊棋目前已成为国际计算机奥林匹克锦标赛常规比赛项目，自我国开展计算机博弈锦标赛以来，亚马逊棋一直是常规比赛项目，2011 年起开始的全国大学生计算机博弈大赛也将其列为比赛项目，也是比赛参与者较多的项目之一。

在国际计算机奥林匹克锦标赛上成绩较好的软件分别为美国加州大学 Northridge 分校的 Invader、瑞士巴塞尔大学的 Jens Lieberum 开发 Amazong 和加拿大埃尔伯塔大学 Michael Buro 开发的 Amsbot 等，各校的学者从不同的角度对亚马逊的搜索算法和估值函数进行大量研究，在 2000 年时，E. Berlekamp 的分析文章提出了亚马逊棋属于组合博弈的思想，搜索下一个位置的搜索量为 2^n ，而后，Michael Buro 证明了了亚马逊棋的解是类似完全 NP 问题，并且提出了复合概率剪枝的算法，复合概率剪枝算法对 $\alpha - \beta$ 剪枝算法进行了较大的改进，极大提高了剪枝效率，较大提高了搜索的深度，在目前的一些优秀的 AI(Artificial Intelligence, 人工智能) 博弈程序中，很多软件采用了该算法，而 Northridge 分校的 Invader 所采用的算法包含了多种技术，由 2003 年开发的采用简单的 MINIMAX 算法，到现在采用了蒙特卡洛算法、UCT 算法等，有效提高了搜索的效率与搜索的深度，同时在下棋过程中采用了可变深度搜索算法，在棋局的不同阶段采用了不同的搜索深度，搜索的深度通过搜索的时间进行的控制，且终局采用了终局数据库，有效提高了终局的搜索速度，是目前成绩最好的亚马逊棋软件。

亚马逊棋棋盘的大小规格为 10×10 ， 10×10 比赛用棋盘如图 6-21 所示。



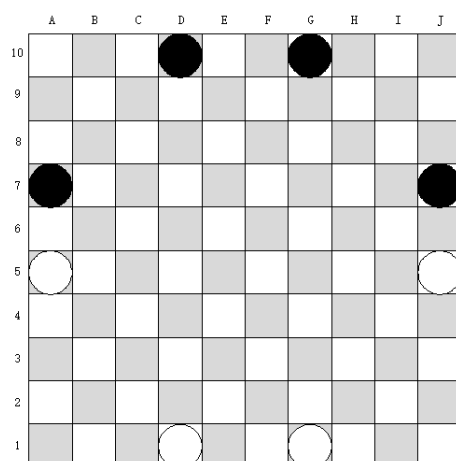


图 6-21 亚马逊棋 10×10 棋盘

亚马逊棋是一种高雅的并且规则非常简单的游戏，Berlekamp 从组合博弈理论出发提出了亚马逊棋具有的以下特点：

- 是一种双人游戏或人机对弈游戏。
- 可以下棋位置有限且有固定的规则。
- 对每一个游戏者来说，一个有限的可下位置引导出不同的下一步可下位置。

- 下棋双方轮流下棋。
- 下棋过程在有限的步数中完成。
- 属于信息完备博弈问题。
- 是一种无偏博弈。

在图 6-21 所示的亚马逊棋盘棋盘中，左侧采用的是数字标记，上方采用的是字母标记，这是在很多棋盘表示中常用的一种方法，即行用数字表示，列用字母表示，例如在顶部左侧的黑棋的位置可以表示为 D10，这种表示方法来表示棋子所在的位置非常直观，假设列也用数字表示，则原来的 D10 就改为 410，这种表示方法不能很好地体现出下棋的确切位置，如果要完整记录一盘棋的下棋过程就相当不方便。

便，目前大多数类似棋局的记录都是采用字母结合数字的表示方法，如围棋、国际象棋等都是采用这种方法来记录棋局，或采用以这种方法为基础的改进方法来记录棋局。

6.13.2 游戏规则

亚马逊棋的规则比较简单，棋子下棋的规则与设置障碍的规则相同，只要理解下棋的规则，那么，障碍设置的规则就迎刃而解，亚马逊棋规则如下：

1、在 10×10 的棋盘上红方(白方)在 A4, D1, G1 和 J4 位置上摆放白方四个皇后，蓝方(或黑方)在 A7, D10, G10 和 J7 位置上摆放四个皇后。

2、皇后可下棋的位置与国际象棋皇后下法的规则相同。

3、由红方(或白方)开始游戏，每轮下棋由两步组成：

A) 移动摆放皇后位置，规则和国际象棋皇后走棋的规则相同。

B) 落子后以当前皇后位置为基点设置障碍，障碍摆放点的位置和皇后可摆放点的位置相同(两者使用的规则相同)。

4、皇后和障碍设置的线路上不得有其他棋子或障碍。

5、可以完成最后一步的一方为赢家。

注：皇后下法规则为在无障碍条件下皇后可下在上、下、左、右、左上、左下、右上和右下的任何可到达的棋盘上的位置，设置障碍的方法与皇后的下法相同。

在全国大学生计算机博弈大赛中对比赛时间做了进一步的规定，目前采用的方法是包干计时，对弈各方用时不超过 20 分钟，超时判

负，这就对程序的搜索算法提出了更高的要求，对于亚马逊棋，一盘棋的下棋步数最多为 92 步，每方为 46 步，这样每步棋的平均用时为 26 秒，全盘考虑，每一步棋的计算使用时间最好不要超过 20 秒，否则很容易超时，在国际计算机奥林匹克锦标赛中也有相应的时间约束。

在第五条规则中规定了可以完成最后一步的一方为赢家，假如双方下棋过程中占领的格子的数目相同，那么，后手方总是占领最后一个格子，即完成最后一步，此时，后手方获胜，这样就在一定程度上消除了先手方的优势，使整个下棋过程更为公平，一些博弈爱好者在设计软件时将先手和后手下棋加以区别，当先手下棋时更注意进攻，而后手下棋时以防守为主，以争取取得更高的胜率。

6.13.3 亚马逊棋博弈主要技术

6.13.3.1 棋盘表示

亚马逊棋的棋盘为 $n \times n$ 的棋盘，目前比赛标准用的棋盘为 10×10 棋盘，棋盘处理分为两个部分，一部分为棋盘的内部数据，一部分为棋盘当前状态的显示，在该示例中用结构体来处理棋盘，棋盘上各个位置的状态可以用常量来表示，例如“0”表示棋盘位置为空，“1”表示白棋，“2”表示黑棋，“3”表示为障碍物等。

6.13.3.2 局面评估

亚马逊棋的估值是亚马逊棋博弈技术中最为复杂的技术，也是亚马逊棋软件博弈水平的核心。其估值函数主要由两方面组成：其一是对下棋过程中所占“领地”进行评估，其二是针对棋子所占位置进行评估，将两者结合就形成了亚马逊棋局面的估值，在有些软件中还将

棋子的灵活性评估加入到估值函数中，具体采用哪种估值方法需根据搜索方法与具体比赛规定的时间综合考虑。

一、领地的评估

亚马逊棋下棋过程中每一方一轮要下两步，一步是移动棋子，一步是释放障碍，其下棋的过程如图 6-22 所示。

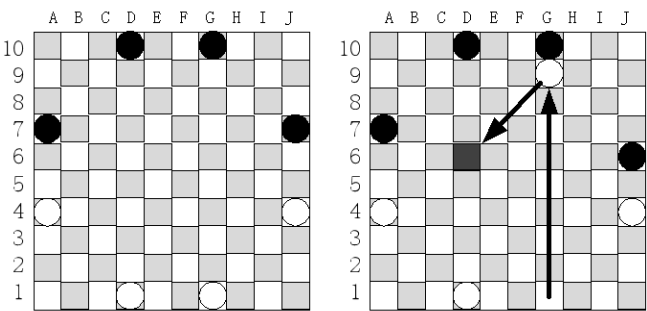


图 6-22 亚马逊棋下棋基本过程示意图

图 6-22 的左图为亚马逊棋皇后放置的起始位置，走棋和释放障碍的规则是按照国际象棋皇后的下棋规则来执行。游戏双方在后续过程中用 player $j(j \in (\text{Al-Asadi \& Tasdemir, 2022}))$ ，通常白方为 player1，黑方为 player2，图 6-13 的右图为白方第一轮下棋的过程，G1 的皇后首先移动到 G9 位置，然后由 G9 位置出发，释放障碍到 D6，完成白方的第一轮下棋，然后由黑方下棋，黑方下棋的过程和白方下棋的过程完全相同。

一个典型的中局和残局的情况如图 6-23 所示。

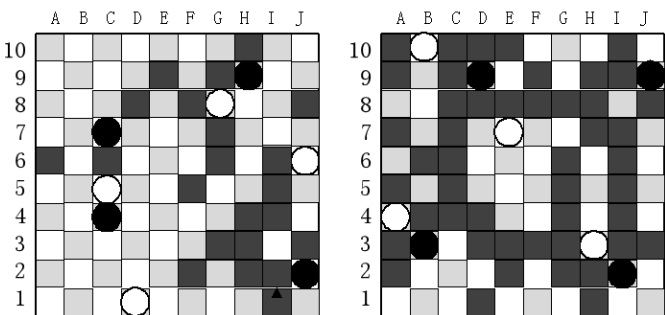


图 6-23 典型的中局和残局

在游戏的中局通常还有一定的“领地”未被划分，如图 6-23 中的左图，在图中尚未有某一领地被下棋的某一方完全占领，此时双方博弈的焦点是如何让己方能够占领更大的区域，同时尽可能限制对方占领更大的区域，这个过程也是亚马逊棋博弈的重点，随着游戏进入残局时，领地已被明确分割，棋盘被分割成一些各自独立的区域，如图 6-23 中的右图，此时实际上已分胜负，双方可能占领的领地区域已经明确区分，最终双方在各种的“领地”内填充障碍直至结束。

二、灵活度的评估

亚马逊棋的棋局评估的另外一个需要考虑的重要因素是棋子在棋盘中的灵活性，与其它一些计算机博弈游戏相比(如六子棋)，游戏双方在开局之后仅在一个较小的游戏范围内进行博弈，因此还需要考虑棋子在棋盘中的灵活性来对棋局估值函数进行进一步修正。

当棋局开始之后，一方的棋子就被限定在一定的范围内，如 player1 的一个棋子 A 被限定在一个有 n 点构成的范围内，此时棋盘被分割成两部分：控制领地的范围(内部范围)和不被控制的领地范围(外部范围)。在内部范围中可以直接计算出控制的领地 n ，对估值函数 t 的贡献值为 n 。在外部范围，player1 的另外一些活动的棋子将保护那些对 A 来说已经影响不到的位置，另外，走棋的另一方 player2 也会尽力圈住 player1 的 A 没有圈住的地盘，并且尽可能扩展自己的地盘，此时，A 已经无法影响到外部范围的估值 t ，这种估值方法通常在下棋过程进行到一定程度后才会较好地发挥作用。

考虑到灵活度的及时作用，引进了另外一个关联常数 m ，用 m 来

统计每个亚马逊棋子的灵活度。在估值函数 t 中，一些较为灵活的棋子可以对处于被动位置的棋子实施保护，为了能更快地计算 m ，可以考虑添加一个相关参数 $N(a)$ ，用它来表示棋盘空格中棋子从 a 点出发棋子一格就能移动到的格子数， $N(a)$ 一般在搜索过程中随着棋子位置的变化而需要及时更新，对于格子 a 上 player j 的一个棋子 A ，设：

$$\alpha_A = \sum_b 2^{-d_1(a,b)} N(b) \quad 2$$

此处，我们统计所有满足条件： $d_1(a,b) \leq 1$ 和 $D_1^{3-j}(b) < \infty$ 的格子 b ，如果 $\alpha_A = 0$ 则 A 是被圈住的。

图 6-24 所示为 $N(a)$ 和 α_A 的计算情况。

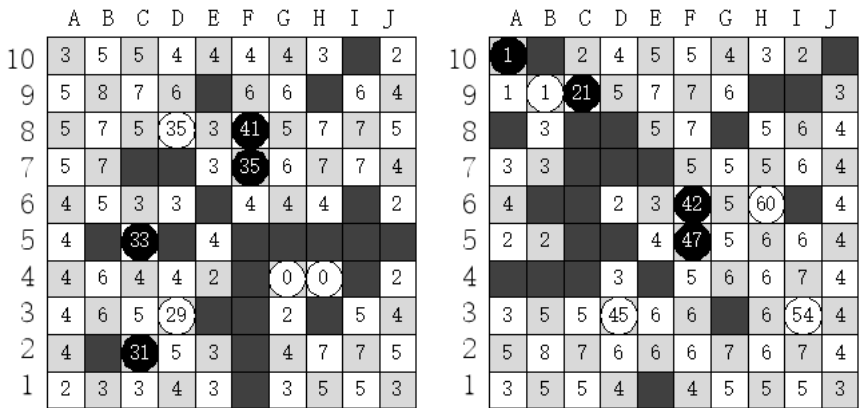


图 6-24 $N(a)$ 和 α_A 的计算情况

以 B9 点为例，因为 B9 周围的 8 个点均为未被占领的格子，故该点的 $N(a)$ 的值为 8，对于 D8 点为白棋占据， $b=1$ 的点有 C9、C8、D9、E8 和 E7， $b=2$ 的点有 B10、D10、B8 和 F6， $b=3$ 的点有 A8，计算 $\alpha_A = 7 + 6 + 5 + 3 + 3 + (5 + 4 + 7 + 4) / 2 + 5 / 4 = 35.25$ ，取整数为 35。

综合考虑领地和灵活度，可以获得亚马逊棋的估值方法。

6.13.3.3 搜索算法

亚马逊棋具有极大的分支因子，使用 10×10 的棋盘在开局时理论上分支因子就达到 2176，在前 10 步时平均分支因子超过 1000，因此在搜索过程中限制了搜索的深度，而搜索的深度会在很大程度上影响程序的棋力，与中国象棋或国际象棋不同，亚马逊棋在开局之后会逐渐降低搜索的分枝，由于在开局阶段不能进行全宽度的搜索，有的程序采用的方法是在开局阶段采用选择性搜索的方法来降低搜索的分支因子，提高搜索的深度，选择性的搜索一般是在开局后的 10 到 15 步左右的范围内进行，然后再扩展搜索的宽度，直至全宽度搜索，这种方法主要对搜索的总量进行控制以达到最佳的搜索。

亚马逊棋采用的主要搜索算法为 $\alpha - \beta$ 剪枝算法，近几年来，不少 AI 爱好者逐步采用 Multi-ProbCut 搜索算法，该方法是 $\alpha - \beta$ 剪枝算法的改进算法，也出现了剪枝算法结合 UCT 算法 (UCT 是 UCB Applied to Trees 的缩写，UCB 是 Upper Confidence Bound 的缩写，通常称为上确界或上限信心界)，较大提高了搜索效率，由美国加州大学 Northridge 分校开发的 Invader 就是采用 UCT 算法进行开发。

第7章 非完备信息机器博弈的专项技术

非完备信息博弈是机器博弈领域中最具挑战性的分支之一，其核心特征在于参与者无法获取决定博弈结果的全部信息。这类博弈的理论基础主要源于博弈论中的贝叶斯博弈与不完全信息动态博弈模型。在这些模型中，参与者拥有其私人信息（即“类型”），并需根据对他人类型和策略的信念，在动态交互中做出最优决策。其核心解概念

——贝叶斯纳什均衡与序贯均衡，要求参与者的策略在任意决策点上，基于当前的信息信念都是最优的。

解决非完备信息博弈的关键在于两大任务：信念管理（如何对未知信息进行建模与实时更新）与均衡策略求解（如何在信念不确定的情况下找到稳健的最优策略）。本章所介绍的德州扑克、军棋、桥牌等项目，均是这些理论模型的典型实例，后续各节将详细阐述如何将上述理论转化为具体的技术实践。

7.1 德州扑克机器博弈

7.1.1 概述

非完备信息机器博弈问题已被证明是一个 NP 难问题(Blair, Mutchler, & Liu, 1992)，德州扑克是不完全信息动态博弈的经典范例：每位玩家的手牌是其私人类型，公共牌和下注历史是公开的信息集。玩家需要在无法观测对手手牌的情况下，通过其行动序列来推断其手牌范围（即更新信念），并做出决策。该问题的求解目标是在如此庞大的博弈树上找到一个近似的序贯均衡策略，使得在任何信息集下，玩家都无法通过单方面改变策略来提升其期望收益。CFR 等一系列算法的提出，正是为了高效地逼近这一均衡解。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 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(Iolis & Bontempi, 2010); Johannes Heinrich 提出了一种 Kuhn poker 的近似纳什均衡策略(Broeck, Driessens, & Ramon, 2009); 2011 年，文献(Teófilo & Reis, 2011)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(Moravík et al., 2017)，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 (Zinkevich, Johanson, Bowling, & Piccione, 2007)。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。

$$R_i^T = \frac{1}{T} \max_{\sigma_i \in \Sigma_i} \sum_{t=1}^T (u_i(\sigma_i^*, \sigma_{-i}^t) - u_i(\sigma_i^t)) \quad 3$$

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 3。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。

$$R_i^T(I, a) = \frac{1}{T} \sum_{t=1}^T \pi_{-i}^{\sigma^t}(I) (u_i(\sigma_i^t|_{I \rightarrow a}, I) - u_i(\sigma^t, I)) \quad 4$$

对于信息集 I 中的每一个可选行动 a, 玩家 i 在时间 T 的反现实悔恨值如公式 4 所示。其中, $\pi_{-i}^{\sigma^t}(I)$ 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。

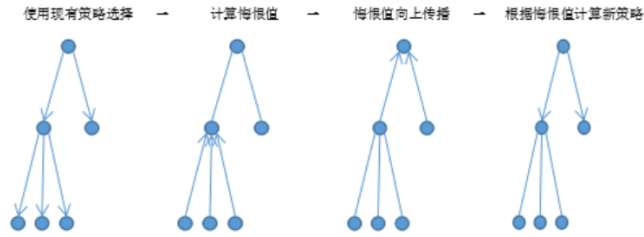


图 7-2 CFR 算法迭代过程

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 (Michael et al., 2015), 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(Moravík et al., 2017)。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(Johanson, 2013)。

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) (Burch, Johanson, & Bowling, 2013), 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) (Schnizlein, Bowling, & Szafron, 2009), 以及预测动作的受限集合(Gilpin, Sandholm, & Srensen, 2008)。

此外,DeepStack 还采用了深度神经网络 (Deep neural networks , DNNs) (Technicolor, Related, Technicolor, & Related)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。

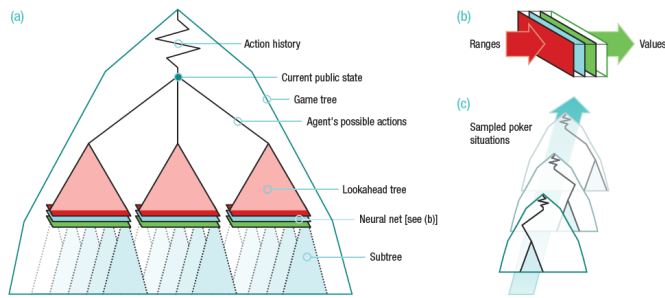


图 7-3DeepStack 算法概览

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图 7-3 (b) 计算。而图 7-3 (b) 中神经网络的训练样本为由图 7-3 (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。从理论上讲，军棋博弈可以建模为一个复杂的贝叶斯博弈。棋盘上每个未知棋子都是一个随机变量，其类型（军阶）服从一个初始概率分布（即先验信念）。每一次碰子结果都为所有棋子的联合概率分布提供了新的证据。因此，棋局演进的过程，本质上是一个持续的贝叶斯信念更新过程。后续介绍的可能性矩阵更新与欺诈走法，正是在这一理论框架下的具体技术实现。

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,	0.03, 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, 0.09, 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,	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,	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(Cooper, 1990)，直接使用精确推理方法效率很低。

信念传播算法(Pearl, 1986)是一种迭代求解概率图模型(Beal, 1990; 徐心和 et al., 2008)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、图 7-6 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。桥牌完美地融合了合作与对抗，并兼具不完全信息与动态特性。叫牌阶段是一个典型的信号博弈，玩家通过有限的叫品空间，向同伴传递关于自己手牌（私人类型）的信号，同时试图迷惑对手。打牌阶段则是一个在部分明牌条件下的不完全信息动态博弈，防守方需要通过出牌信号（如欢

迎张、张数信号) 进行受限的通信, 共同构建并更新关于庄家牌型的共同信念。因此, 桥牌 AI 的构建强烈依赖于对这类复杂交互模型的深刻理解。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 叫牌过程

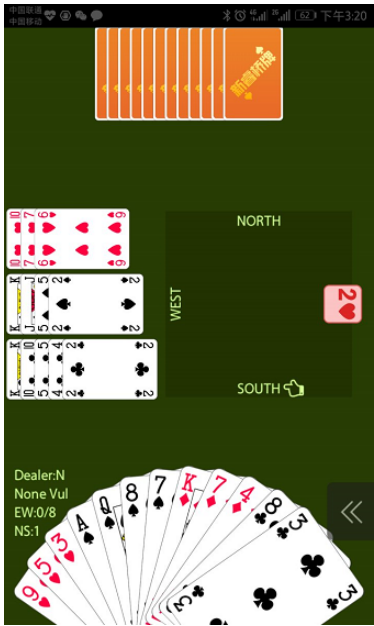


图 7-6 首攻后状态

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(程克非，张聪，& 沈一栋, 2005; 张志刚, 2015)。图 7-6 是一副牌双明手结果的呈现。

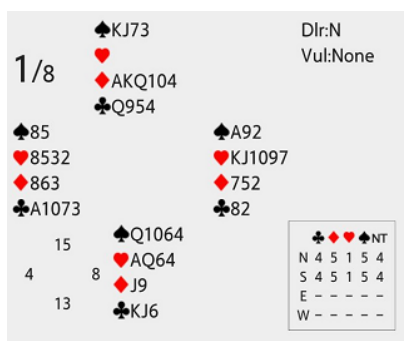


图 7-7 双明手结果图

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 7-8 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.Q652.OT.KJ7 T5.KT73.643.T832 -> HA(3/0/3-10/0/6)
```

图 7-8 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。

7.4 国标（大众）麻将机器博弈

7.4.1 概述

麻将将是人民群众喜闻乐见的娱乐项目，2020 年全国计算机博弈锦标赛正式将麻将纳入竞赛项目类别。麻将是一种 4 人非合作的游戏项目，其中每位玩家仅能看见自己手牌和堂子中玩家已打出的公开牌张，由于玩家可以在不同条件选择吃、碰、杠、胡等对弈行为，打乱玩家的对弈行为和出牌顺序，从而使得麻将博弈成为一个高度不确定的随机游戏，具有强实时性、高随机性和强不确定性，从而使决策难度增大。

麻将是一种非完备信息博弈活动，与其它非完备信息博弈活动相比，由于游戏过程存在吃、碰、杠等特殊行为影响出牌顺序，从而改变正常博弈活动的进程，打乱博弈活动顺序，进一步提高游戏难度(衣御寒 et al.，2024；赵海璐，2022)。

李霞丽等认为麻将游戏博弈算法主要包括基于知识和基于数据的两大类 AI 构建方法(李霞丽，王昭琦，刘博，& 吴立成，2023)。王亚杰等分别针对麻将博弈的弃牌模块、听牌模块、吃牌模块提出了弃牌优先级、听牌有效数、吃牌优先级的方法，完善了麻将 AI 的知识体系，设计了基础版博弈算法 Fanfou_ba 和优化版博弈算法 Fanfou_op(王亚杰，乔继林，梁凯，& 谢延延，2022)。衣御寒等针

对大众麻将中对手牌信息利用不充分的问题，提出了手牌估值方法，并设计了基础麻将程序（MJE），并使用深度强化学习方法设计了麻将 AI（MJE-RL）进一步提升麻将 AI 的博弈能力（衣御寒 et al., 2024）。

李淑芹等针对麻将博弈中状态空间巨大和隐藏信息过多等难点，提出利用局面信息缩减未知状态空间，并通过动态划分游戏状态提升牌型预测准确率的方法，认为局面信息利用方法和动态游戏划分方法不仅可以降低出牌的点炮风险，还可以获得更高的得分，实现通过缩减未知状态空间达到提升麻将博弈水平的目的（李淑琴 & 冯浩东，2022）。张小川等对出牌动作进行研究，以启发式快速出牌为思路，提出了面向敌方胡牌牌张的蒙特卡洛评估法，将启发式快速出牌方法和蒙特卡洛评估法相结合，对每张手牌进行估值计算，通过估值分数决定每轮出牌牌张。以历史出牌次数为分界点，以此分界将博弈过程时序化为前后 2 个决策时段，前段采用启发式快速出牌方法，后段采用蒙特卡洛评估法。通过前后时段法分层递进决策处理过程，给出最佳出牌着法，能有效减少出牌的决策时间并降低点炮率（张小川，严明珠，涂飞，陈俊宇，& 魏乐天，2024）。

7.4.2 国标（大众）麻将机器博弈规则

7.4.2.1 牌库

国标（大众）麻将项目牌库有万、条、筒三种花色，每种花色包含有分别标识为 1-9 的九张牌，每个点数的牌共 4 张，总共 108 张牌，如图 7-9 所示。庄家起手 14 张牌，其他 3 位起手 13 张牌。



图 7-9 国标（大众）麻将牌面示意图

7.4.2.2 打法

国标（大众）麻将项目行牌过程，可吃、碰、杠和报听，并都可获得直接收益。

7.4.2.3 对局

① 4 位玩家分东南西北入座，每人起手摸 13 张牌，由“庄家”位玩家起手按顺时针出牌，“庄家”位玩家起手多摸一张牌，共计 14 张。

② 行牌过程的优先级为：胡牌>杠>碰>吃。

③ 同一人存在多个响应时，比如可以同时响应胡和杠，则需要生成 2 个响应气泡以供选择，同时还需要生成 1 个放弃气泡，以供玩家选择过牌。

④ 如果存在多人响应，则每个人都需要生成相应的响应气泡，最终的执行结果根据行牌优先级判定。

⑤ 当有玩家成功胡牌，则牌局结束。

7.4.2.4 和胡牌

国标（大众）麻将项目胡牌方式包括三种：点炮，自摸，抢杠胡。

① 玩家点击胡牌和听牌自动胡牌皆可，胡牌后按番型计算积分。

② 允许一炮多响，即多个玩家响应一个点炮。

③ 如果牌摸完都无人胡牌则为流局。

7.4.3 麻将博弈平台

目前中国计算机博弈锦标赛国标（大众）麻将平台是由竞技世界（北京）网络技术有限公司开发，该平台分为裁判端、客户端和服务端三部分。裁判端担任游戏控制中心的角色，控制麻将游戏的创建与结束；客户端主要是 AI 智能体与服务器端建立通信的媒介，通过服务端生成的数字码，加入游戏对局。服务端相当于人类大脑，负责处理整个麻将博弈中的状态信息及牌信息，通过 HTTP 通信实现多个 AI 智能体的公共信息交互及指令收发。通过上述三部分，将 AI 智能体接入指定接口，即可实现四位 AI 玩家在线博弈，同时可自行设置游戏局数，完成 AI 程序的战力测试与评估。麻将 AI 是多智能体强化学习的典型应用场景。由于存在四个玩家且信息不完全，最先进的麻将 AI（如 Suphx）普遍采用中心化训练与去中心化执行的范式。通过全局虚拟评论家在训练时指导智能体学习，而对弈时 AI 仅依靠自己的手牌和公共信息进行决策。高级的麻将 AI 会通过 Transformer 等序列模型对每个对手的出牌、吃碰杠行为进行编码，隐式地推断他们的手牌范围和听牌状态，从而显著降低点炮风险并提高胡牌效率。通过这些技术，AI 学会了人类高手都难以稳定掌握的复杂策略，如“战略性弃胡”（在意识到难以获胜时主动降低损失）和“危险的立直”（在计算后认为收益大于风险时，即使有点炮风险也果断听牌）。

7.5 二打一扑克牌机器博弈

7.5.1 概述

斗地主作为一种传统的纸牌类游戏，广受群众喜爱。斗地主机器博弈属于非完备信息博弈，其游戏过程如图 7-10 所示。对于叫牌阶段的决策是基于不完全信息的，每个玩家仅掌握自己手中的 17 张牌，对剩余牌堆分布一无所知，必须根据当前手牌的牌力来决定是否抢当地主并选择相应分值。进入出牌阶段后，玩家身份已经明确：两位农民需要协同作战以对抗地主，但由于无法直接交流策略，增加了合作难题；而地主则需面对双重挑战，既要利用额外获取的 3 张底牌优化组合，又要应对两名对手可能采取的不同战术(王鑫，2024)。

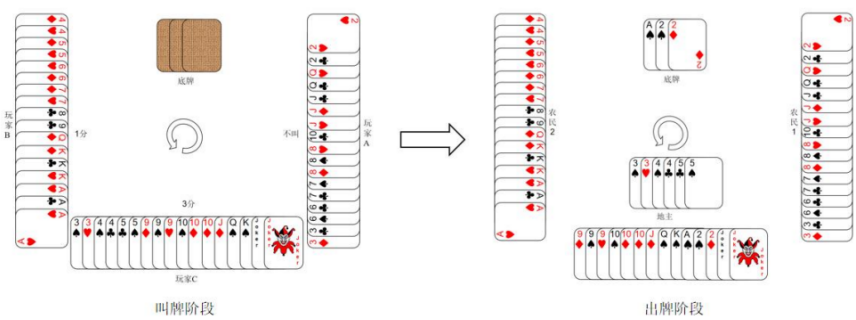


图 7-10 二打一扑克牌流程

7.5.2 计分规则

地主方该局实际抓牌分数定义为基础分 N （取值有 1、2、3 三种情况），地主赢了得 $2N$ 分，输了得 $-2N$ 分，农民方赢了得基础分 N 分，输了得 $-N$ 分。为了克服极端牌型造成一把定乾坤，打出了王炸、炸弹、春天、反春天牌型不倍乘分数，倍乘换成加法，假如农民赢了“农民该局的得分= $N+(\text{王炸}+\text{炸弹数}+\text{春天}+\text{反春天})\times N$ ”，“地主该局得分= $-2\times(N+(\text{王炸}+\text{炸弹数}+\text{春天}+\text{反春天})\times N)$ ”，假如地主赢了“地主该局得分= $2\times(N+(\text{王炸}+\text{炸弹数}+\text{春天}+\text{反春天})\times N)$ ”，“农民该局的

得分 $=-(N+(王炸+炸弹数+春天+反春天)*N)$ ”。

7.5.3 二打一扑克牌机器博弈的特点

斗地主计算机博弈作为非完美信息博弈的代表，因其在国内拥有庞大的玩家群体和高度复杂性而备受瞩目。游戏中的隐藏信息、分阶段决策、动作状态空间巨大以及竞争与合作并存等特点，为相关研究带来了巨大挑战(贺泽亚，2023；王鑫，2024)。

(1) 斗地主机器博弈即存在对抗又存在合作。斗地主需要三个玩家，其具有两种角色，一个玩家是地主，另外两个玩家是农民。斗地主的农民玩家既需要和地主对抗又要和另一个农民合作，而且两个农民之间不能有任何交流，只能看到自己的手牌和已经打出的牌。这样合作的难度大大增加。

(2) 斗地主机器博弈包含巨大的状态和动作空间。其信息集数高达 $10^{58} \sim 10^{83}$ ，动作空间高达 10^4 。

斗地主是合作（农民方）与对抗（地主 vs 农民）并存的混合型博弈。解决此问题的关键在于多智能体强化学习中的团队对抗算法。农民 AI 需要在没有显式通信的情况下，通过长期的合作博弈学习，形成默契的分工（如一人负责牵制，另一人负责跑牌）。AI 需要学习地主和农民两种截然不同的角色策略。先进的训练框架会让 AI 在种群自博弈中同时学习所有角色，从而更深刻地理解整个游戏的动态平衡，并学会针对不同角色采取最优应对。

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。

(16)2024 年国际象棋世界冠军赛由谷歌冠名赞助，该赛事将于 2024 年 11 月 23 日至 12 月 15 日在新加坡圣淘沙名胜世界举行。这是国际象棋世界冠军赛首次由一家全球互联网科技巨头赞助，标志着国际象棋运动的传播和影响力树立了新的标准。

(17) 2025 年 8 月于 Kaggle 平台举办的 AI 国际象棋锦标赛提供了全新的赛事范式。该赛事旨在纯粹测试大语言模型的内在推理能力，要求包括 OpenAI o3、xAI Grok 4、谷歌 Gemini 系列在内的 8 款

主流模型在不调用专用象棋引擎的情况下自主生成合法棋步。最终，OpenAI 的 o3 模型在决赛中以 4:0 的绝对优势战胜 xAI 的 Grok 4 夺冠，标志着大语言模型在复杂策略游戏的自主决策方面取得了显著进展。

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“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)2017 年 12 月，在东京举办的 2017 围棋 AI 龙星战(AI RYUSEI)决赛中，腾讯 AI Lab 围棋 AI“绝艺”以平稳表现相继战胜 DeepZenGo、MayoiGo、Raynz 和 AQ 等国际领先的 AI，夺得冠军。

(9)2024 世界人工智能围棋锦标赛由中国计算机学会、中国围棋协会、深圳市宝安区人民政府主办，比赛时间为 2024 年 11 月 7 日至 10 日，地点在中国广东省深圳市宝安区。

(10)2025 年,中国围棋 AI “绝艺” 在第 10 届 UEC 杯世界计算机围棋赛中表现出色，以 11 连胜的战绩夺得冠军。

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。

2022 年 3 月，法国 NukkAI 公司开发的桥牌 AI “NooK” 在巴黎举行的为期两天的桥牌锦标赛上击败了 8 名世界冠军，这一成就被认为是 AI 在不完全信息博弈中的一个重要里程碑。

2025 年，桥牌计算机博弈赛事呈现出线上人机协作的新趋势，世界桥联与北美桥联合作推出的首届“桥牌电竞杯”吸引了全球选手与 AI 搭档参赛，标志着该项智力运动在智能化与普及化方面迈出关键一步。

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。

(5)2022 年，中科院自动化所的兴军亮教授团队开发的轻量型德州扑克 AI 程序 AlphaHoldem 在 AAI 2022 上获得了卓越论文奖。

(6) 2025 年，世界扑克系列赛（World Series of Poker, WSOP）再创纪录，总参赛人次高达 246,960。在此次赛事中，中国选手罗曦湘成功登顶，勇夺象征最高荣誉的 WSOP 金手链，标志着中国选手在国际顶级扑克竞技舞台上取得了历史性突破。

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。

(4) 2025 年在中国安徽祁门举办的国家级智能博弈赛事，汇聚了全国 60 余所高校的 400 多支队伍，围绕五子棋、亚马逊棋、藏族久棋等 19 个项目展开角逐，并创新性地结合了“机器博弈+AI 围棋”的竞赛模式。在研究层面，挑战已从传统的完备信息博弈（如国际跳棋、围棋）延伸至更复杂的非完备信息与合作型博弈。

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

中国象棋人机与机机博弈大赛是中国象棋人工智能领域的重要赛事，它不仅展示了人工智能在象棋领域的最新进展，也为象棋爱好者提供了与人工智能对弈的机会。

(1) 2005 年 9 月，由“岭南双雄”吕钦、许银川领军的征战全国象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。

(4) 2021 年 7 月 6 日的一场“人机大战”中，中国象棋特级大师孙勇征与“小原”对弈，最终孙勇征认负。

(5) 2022 年 10 月 15 日，商汤科技的 AI 下棋机器人与中国象棋特级大师、世界冠军谢靖进行了一场人机大战，最终 AI 机器人获胜。

(6) 2025 年，国内象棋机器博弈领域持续活跃，以覆盖象棋等 23 个棋种、吸引全国 60 余所高校 400 余支队伍参赛的中国大学生计算机博弈大赛为代表的大型赛事蓬勃发展，同时“元萝卜杯”等高水平人机对决也多次上演并以和棋告终，标志着赛事体系日趋完善且人机水平趋于均衡。

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，大赛创始人东北大学徐心和教授，大赛网址：<http://computergames.caa.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三个阶段：

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：2011~2021，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。

第三个阶段：2021 年至今，中国大学生计算机博弈大赛暨中国计算机博弈锦标赛阶段，大赛由中国人工智能学会主办，2024 年赛事共包括为中国大学生计算机博弈大赛（10 项）、中国计算机博弈锦标赛（11 项）和中国大学生机器博弈创新设计赛。

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、爱恩斯坦棋、军棋、海克斯棋等 10 种棋类；面向全社会的锦标赛项目包括：中国象棋、围棋（19 路）、国际跳棋（100 格）、国际跳棋（64 格）、二打一扑克牌（斗地主）、桥牌、德州扑克、藏棋久棋和麻将等 9 种棋牌类。以及在 2024 年新增的 2 项软、硬件结合新项目：机器博弈五子棋、机器博弈坦克大战，其中，机器博弈五子棋项目由浙江省机器人创新中心（浙江钱塘机器人及智能装备研究有限公司）和重庆工能科技有限公司协助举办，机器博弈坦克大战项目由北京博创尚和科技有限公司协助举办。新增的 2 项软、硬结合新项目在 2024 年仅面向在校大学生开放。

中国大学生机器博弈创新设计大赛从 2024 年开始进行相应的竞赛赛制完善，此项赛事主要面向在校大学生。

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	棋类项目数	参赛队数	参赛学校
-----	------	-------	------	------

	或比赛地点			(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
2018	安徽大学	19	253	42
2019	中央民族大学	19	277	42
2020	重庆理工大学(在线)	19	332	49
2021	重庆三峡学院	19	417	60
2022	成都理工大学	19	433	69
2023	渤海大学	19	395	55
2024	青海民族大学	21	471	71
2025	安徽祁门	23	378	53

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），旨在推动我国人工智能技术的发展和應用，以促进经济和社会的全面进步。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。

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，成为衡量一个国家人工智能水平的重要指标之一。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。近年来，机器博弈领域发展表现出来的主要特点包括：

1. 智能化程度不断提升

机器博弈的智能化程度提升是人工智能领域的一个重要趋势，主要体现在以下几个方面：

(1) 深度学习与强化学习的结合

深度学习在感知和认知能力上的提升，与强化学习在决策和策略优化上的优势相结合，形成了深度强化学习方法。这种方法在围棋、扑克等游戏中取得了显著的成果，如 **AlphaGo** 和 **AlphaStar** 等系统就是典型代表。这些系统通过自我对弈和蒙特卡洛树搜索等技术，不断提升策略的鲁棒性和多样性，实现了在复杂博弈环境中的高效求解。

(2) 多智能体博弈的发展

随着人工智能技术的发展，多智能体博弈、学习与控制论等领域的交叉研究成果呈现出井喷式增长。这些研究不仅在理论上取得了突破，也在实际应用中展现出广泛的潜力，如在社会智能、机器智能、合作智能、**AI 安全** 和 **AI 伦理** 等新兴交叉研究领域扮演着越来越重要的角色。

(3) 不完全信息博弈的突破

在不完全信息博弈领域，如德州扑克等，人工智能系统如 **Libratus** 通过博弈论的方法，不依赖于机器学习和深度学习，实现了在大规模复杂博弈中的高效求解。这表明博弈论在人工智能中的应用潜力，尤其是在处理现实世界中的不完全信息问题时。

2. 理论研究与实践应用联系更加紧密

机器博弈领域的学术研究与产业界的结合越来越紧密，推动了智

能化技术的发展。企业与研究机构的合作，将科研成果转化为具有更高人工智能水平的产品，如智能游戏平台、智能决策控制系统等。计算机博弈技术正在向实际应用领域拓展，如智慧医疗、智能交通、航空、航天等。特别是在军事国防领域，机器博弈技术与人工智能的结合，推动了智能武器和系统的发展，如无人机的自主决策与控制，兵棋推演系统中的敌方行为模拟等。

3. 多学科领域融合不断扩大

机器博弈正在与博弈论、控制论、社会智能、机器智能、合作智能、AI 安全和 AI 伦理等多个学科进行深度交叉融合，开始从棋类游戏扩展到更广泛的领域，如金融、交通、军事等，这种融合不仅推动了传统学科壁垒的打破、促进了新兴交叉研究领域的发展，这种融合领域中的决策问题还具有实际的社会和经济价值。

近年来，交叉研究成果正呈现出井喷式增长的态势，表明了多学科融合在机器博弈领域的重要性和潜力。

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，将人工智能带上一个新的台阶。机器博弈未来发展趋势主要体现在：

1. 算法创新：随着深度学习、强化学习等技术的不断发展，我们可以期待更加高效和智能的博弈算法问世，能够更好地处理复杂和动态的博弈环境。

2. 应用场景拓展：人工智能博弈技术将广泛应用于金融、交通、

军事等领域，通过构建智能交易系统、优化交通流量管理、辅助指挥员进行战术规划和决策支持等。

3. 协同与对抗共存：未来的 AI 系统不仅需要与人类合作，还需要与其他 AI 系统进行博弈，催生出一系列新的研究问题。

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。本报告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、设计规范和关键技术；以六子棋、围棋、点格棋和爱因斯坦棋等为代表，介绍了完备信息博弈的关键技术，以德州扑克、军棋、桥牌等为代表，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，为从事机器博弈及相关领域科研和应用人员提供参考(中国人工智能学会机器博弈专业委员会, 2014a)。

参考文献

- Al-Asadi, M. A., & Tasdemir, S. (2022). Predict the Value of Football Players Using FIFA Video Game Data and Machine Learning Techniques. *IEEE Access*, 10, 22631–22645. doi:10.1109/ACCESS.2022.3154767
- Allis, L. V., Herik, H. J. V. D., & Huntjens, M. P. H. (2010). Go-Moku Solved by New Search Techniques. *Computational Intelligence*, 12(1), 7–23. doi:10.1111/j.1467-8640.1996.tb00250.x
- Allis, V. (1993). *Searching for Solutions in Games and Artificial Intelligence*. (Ph.D). University of Limburg,
- Amos, M., & Coldridge, J. (2012). A genetic algorithm for the Zen Puzzle Garden game. *Natural Computing*, 11(3), 353–359. doi:10.1007/s11047-011-9284-7
- Anwar, A. H., Atia, G., & Guirguis, M. (2021). A Game-Theoretic Framework for the Virtual Machines Migration Timing Problem. *IEEE Transactions on Cloud Computing*, 9(3), 854–867. doi:10.1109/TCC.2019.2905605
- Auer, P., Cesa, N., & Fischer, P. (2002). Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2-3), 235–256.
- Baier, H., & Winands, M. H. M. (2011). *Time Management for Monte-Carlo Tree Search in Go*. Paper presented at the Advances in Computer Games.

- Banerjee, B., & Stone, P. (2007). *General game learning using knowledge transfer*. Paper presented at the IJCAI'07: Proceedings of the 20th international joint conference on Artificial intelligence, Hyderabad, India.
- Barker, J. K., & Korf, R. E. (2012). *Solving dots-and-boxes*. Paper presented at the AAAI'12: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence.
- Baudet, & M., G. (1978). An Analysis of the Full Alpha-Beta Pruning Algorithm. *ACM*, 296–313. doi:10.1145/800133.804359
- Baudi, P., & Gailly, J. L. (2011). *PACHI: State of the Art Open Source Go Program*. Paper presented at the Advances in Computer Games, Springer, Berlin, Heidelberg.
- Beal, D. F. (1990). A generalised quiescence search algorithm. *Artificial Intelligence*, 43(1), 85–98. doi:10.1016/0004-3702(90)90072-8
- Belardinelli, F., Ferrando, A., & Malvone, V. (2023). An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information. *Artificial Intelligence*, 316, 103847. doi:10.1016/j.artint.2022.103847
- Bench-Capon, T. J. M., & Dunne, P. E. (2007). Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15), 619–641. doi:10.1016/j.artint.2007.05.001

- Berlekamp, E. R. (2000). *The dots-and-boxes game - sophisticated child's play*. Natick, MA: A K Peters Ltd.
- Berliner, H. (1979). The B* tree search algorithm: A best-first proof procedure. *Artificial Intelligence*, 12(1), 23–40.
- Bertsimas, D., & Paskov, A. (2022). World-class interpretable poker. *Machine Learning*, 111(8), 3063–3083. doi:10.1007/s10994-022-06179-8
- Blair, J., Mutchler, D., & Liu, C. (1992). *Heuristic Search in One-Player Games With Hidden Information*. Retrieved from
- Bo, J., Li, S., Geng, R., & Tong, J. (2020). *Design of No-Go Game Algorithm Based on Reinforcement Learning*. Paper presented at the 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Shenyang, China.
- Broeck, G. V. D., Driessens, K., & Ramon, J. (2009). *Monte-Carlo tree search in poker using expected reward distributions*. Paper presented at the Proceedings of the 21st Benelux Conference on Artificial Intelligence.
- Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., . . . Colton, S. (2012). A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational*

- Intelligence & Ai in Games*, 4(1), 1–43.
doi:10.1109/TCIAIG.2012.2186810
- Bulò, S. R., Biggio, B., Pillai, I., Pelillo, M., & Roli, F. (2017). Randomized Prediction Games for Adversarial Machine Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11), 2466–2478. doi:10.1109/TNNLS.2016.2593488
- Burch, N., Johanson, M., & Bowling, M. (2013). *Solving Imperfect Information Games Using Decomposition*. Paper presented at the AAAI'14: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence.
- Chaslot, GMJB, Winands, MHM, van, den, . . . HJ. (2008). *Parallel Monte-Carlo Tree Search*. Paper presented at the 2008 International Conference on Computers and Games, Springer, Berlin, Heidelberg.
- Chen, Y., Wu, L., Yan, J., & al, e. (2023). *Tibetan Jiu Chess Intelligent Game Platform*. Paper presented at the 2023 International Conference on Cognitive Computation and Systems, Singapore.
- Christodoulou, G., Leonardi, S., & Sgouritsa, A. (2016). *Designing Cost-Sharing Methods for Bayesian Games*. Paper presented at the Algorithmic Game Theory, Berlin, Heidelberg.
- Cooper, G. (1990). Computational Complexity of probabilistic inference using Bayesianbelief networks (research note). *Artificial*

- Intelligence*, 42(2-3), 393–405.
doi:10.1016/0004-3702(90)90060-D
- Cormen, T. H., Rivest, R. L., Leiserson, C. E., & C., S. (2009). *Introduction to Algorithms*. Cambridge: MIT Press.
- Coulom, R. (2006). *Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search*. Paper presented at the Computers and Games. CG 2006.
- Dietterich, T. G. (1997). Machine Learning Research: Four Current Directions. *Ai Magazine*, 18(4), 97–136.
doi:10.1177/105971239700500310
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1). doi:10.1007/BF01386390
- Donkers, H. H. L. M., Uiterwijk, J. W. H. M., & Herik, H. J. V. D. (2001). Probabilistic opponent-model search. *Information Sciences*, 135(3–4), 123–149. doi:10.1016/S0020-0255(01)00133-5
- Enzenberger, M., Mu?ller, M., Arneson, B., & Segal, R. (2011). Fuego—An Open-Source Framework for Board Games and Go Engine Based on Monte Carlo Tree Search. *IEEE Transactions on Computational Intelligence & Ai in Games*, 2(4), 259–270.
doi:10.1109/TCIAIG.2010.2083662
- Esparcia-Alcázar, A. I., Martínez-García, A., Mora, A., Merelo, J. J., & García-Sánchez, P. (2010, 18–23 July 2010). *Controlling bots in a*

- First Person Shooter game using genetic algorithms*. Paper presented at the IEEE Congress on Evolutionary Computation.
- Fanfu, Z., Xian, M., Jun, L., Wen, X., & Xiaobo, L. (2016). *Research on estimate system of computer game decision algorithm*. Paper presented at the Chinese Control and Decision Conference(CCDC).
- Fraenkel, A. S., & Lichtenstein, D. (1981). Computing a perfect strategy for $n \times n$ chess requires time exponential in n . *Journal of Combinatorial Theory, Series A*, 31(2), 199–214.
doi:10.1007/3-540-10843-2_23
- Funge, & David, J. (2009). Artificial Intelligence for Computer Games: An Introduction. *International Journal of Computer Games Technology*, 2009(1), 95–99. doi:10.1155/2009/251652
- Garg, S., & Songara, D. (2016). *The winner decision model of tic tac toe game by using multi-tape turing machine*. Paper presented at the Advances in Computing and Communications.
- Gelly, S., Hoock, J.-B., Rimmel, A., Teytaud, O., & Kalemkarian, Y. (2008). *On the Parallelization of Monte-Carlo planning*. Paper presented at the International Conference on Informatics in Control, Automation and Robotics.
- Gelly, S., Hoock, J. B., Rimmel, A., Teytaud, O., & Kalemkarian, Y. (2008). *On the Parallelization of Monte-Carlo planning*. Paper presented at the Eprint Arxiv.

- Gelly, S., & Silver, D. (2007a). *Combining Online and Offline Knowledge in UCT*. Paper presented at the International Conference on Machine Learning.
- Gelly, S., & Silver, D. (2007b, 20 July, 2007). *Combining Online and Offline Knowledge in UCT*. Paper presented at the International Conference on Machine Learning, Corvalis, Oregon, USA.
- Gelly, S., & Silver, D. (2008, July 13–17, 2008). *Achieving Master Level Play in 9 x 9 Computer Go*. Paper presented at the the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA.
- Gelly, S., Wang, Y., Munos, R., & Teytaud, O. (2006). Modification of UCT with Patterns in Monte-Carlo Go.
- Gilpin, A., Sandholm, T., & Srensen, T. B. (2008). *A heads-up no-limit Texas Hold'em poker player: discretized betting models and automatically generated equilibrium-finding programs*. Paper presented at the Adaptive Agents and Multi-Agents Systems.
- Guez, Silver, & Dayan. (2013). Scalable and Efficient Bayes-Adaptive Reinforcement Learning Based on Monte-Carlo Tree Search. *J ARTIF INTELL RES*, 48(1), 841–883.
- Guo, Q., & Chen, Y. (2018). *Artificial Intelligence Algorithm for the Game of NoGo based on Value Evaluation*. Paper presented at the

- International Conference on Information Systems and Computer Aided Education, Changchun, China.
- Gurov, D., Goranko, V., & Lundberg, E. (2022). Knowledge-based strategies for multi-agent teams playing against Nature. *Artificial Intelligence*, 309(8), 103728. doi:10.1016/j.artint.2022.103728
- Hans, J., Berliner, and, Chris, & McConnell. (1996). B* probability based search. *Artificial Intelligence*, 86(1), 97–156. doi:10.1016/0004-3702(95)00092-5
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1972). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science & Cybernetics*, 4(2), 28–29. doi:10.1145/1056777.1056779
- Hashimoto, J., Kishimoto, A., Yoshizoe, K., & Ikeda, K. (2011). *Accelerated UCT and Its Application to Two-Player Games*. Paper presented at the Advances in Computer Games.
- Hearn, R. A. (2005). Amazons is PSPACE-complete. *ArXiv*, abs/cs/0502013, 1–7. doi:10.48550/arXiv.cs/0502013
- Herik, H. J. v. d., Uiterwijk, J. W. H. M., & Rijswijk, J. v. (2002). Games solved: Now and in the future. *Artificial Intelligence*, 134(1), 277–311. doi:10.1016/S0004-3702(01)00152-7
- Hinton, G. (2000). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8), 1711–1800.

- Hinton, G., Deng, L., Yu, D., Dahl, G. E., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6), 82–97. doi:10.1109/MSP.2012.2205597
- Hinton, G. E. (2011). Technical Perspective A Better Way to Learn Features. *Communications of the Acm*, 54(10), 94–94. doi:10.1145/2001269.2001294
- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527–1554. doi:10.1162/neco.2006.18.7.1527
- Hongkun, Q., Peng, Z., Yajie, W., Jiehong, W., & Fei, L. (2014). *Analysis of search algorithm in computer game of Amazons*. Paper presented at the 2014 26th Chinese Control And Decision Conference (CCDC).
- Huang, G., Xu, T., Wang, M., Yi, Q., Gong, X., Li, S., . . . Zhou, B. (2025). Low-probability Tokens Sustain Exploration in Reinforcement Learning with Verifiable Reward. *arXiv preprint arXiv:2510.03222*.
- Inan, M. S. K., Hasan, R., & Prama, T. T. (2021, 1–4 Dec. 2021). *An Integrated Expert System with a Supervised Machine Learning based Probabilistic Approach to Play Tic-Tac-Toe*. Paper presented

- at the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON).
- Iolis, B., & Bontempi, G. (2010). *Comparison of Selection Strategies in Monte-Carlo Tree Search for Computer Poker*. Paper presented at the Proceedings of the 19th Machine Learning conference of Belgium and The Netherlands.
- Jianning, Q., Hongkun, Q., Yajie, W., Fei, L., & Shengran, Q. (2016). *Application of UCT technologies for computer games of Amazon*. Paper presented at the 2016 Chinese Control and Decision Conference (CCDC).
- Jianning, Q., Hongkun, Q., Yu, C., Yajie, W., & Xueping, L. (2016). *Research of test platform on computer games with distributed system*. Paper presented at the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China.
- Jiao, Wang, Chenjun, Xiao, Tan, Zhu, . . . Tseng. (2017). Only-One-Victor Pattern Learning in Computer Go. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(1), 88–102. doi:10.1109/TCIAIG.2015.2504108
- Jing, Z. (2008). Chinese Chess Algorithm Design and Optimize Based on MTD(f). *Computer & Digital Engineering*.
- Johanson, M. (2013). Measuring the Size of Large No-Limit Poker Games. *Computer Science*. doi:10.48550/arXiv.1302.7008

- Kaindl, H., & Shams, R. (1991). Minimax search algorithms with and without aspiration windows. *IEEE Trans on Pattern Analysis & Machine Intelligence*, 13(12), 1225–1235. doi:10.1109/34.106996
- Kasai, I. T. (1994). The Othello game on an $n \times n$ board is PSPACE-complete. *Theoretical Computer Science*, 123(2), 329–340. doi:10.1016/0304-3975(94)90131-7
- Kato, H., Takaya, M., & Yamamura, A. (2015). Analysis of a Monte Carlo Tree Search in Knight-Amazons. *Procedia Computer Science*, 62, 31–38. doi:10.1016/j.procs.2015.08.406
- Kim, H. T., & Kim, K. J. (2013). Hybrid of Rule-based Systems Using Genetic Algorithm to Improve Platform Game Performance. *Procedia Computer Science*, 24, 114–120. doi:10.1016/j.procs.2013.10.033
- Kocsis, L. (2006, September 18–22, 2006). *Bandit Based Monte-Carlo Planning*. Paper presented at the 17th European Conference on Machine Learning, Berlin, Germany.
- Kováčik, V., & V. L. (2019). Value Functions for Depth-Limited Solving in Zero-Sum Imperfect-Information Games. *CoRR*, abs/1906.06412. doi:10.48550/arXiv.1906.06412
- Kurita, M., & Hoki, K. (2021). Method for Constructing Artificial Intelligence Player With Abstractions to Markov Decision

- Processes in Multiplayer Game of Mahjong. *IEEE Transactions on Games*, 13(1), 99–110. doi:10.1109/TG.2020.3036471
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. doi:10.1038/nature14539
- Lee, C. S., Wang, M. H., Hagraas, H., Wu, M. J., Teytaud, O., & Chen, Y. J. (2012). Genetic fuzzy markup language for game of NoGo. *Knowledge-based systems*(34), 64–80. doi:10.1016/j.knosys.2012.02.006
- Leifker, D. B., & Kanal, L. N. (1985). A Hybrid SSS*/Alpha-Beta Algorithm for Parallel Search of Game Trees. *IJCAI*, 1044–1046.
- Li, D., Li, S., & Bao, H. (2011). *Research and Implementation of Dots-and-Boxes Chessboard Based on the Game Strings-and-Coins*. Paper presented at the Advanced Research on Computer Education, Simulation and Modeling. CESM 2011.
- Li, J., Koyamada, S., Ye, Q., Liu, G., & Hon, H. W. (2020). Suphx: Mastering Mahjong with Deep Reinforcement Learning. *ArXiv, abs/2003.13590*. doi:10.48550/arXiv.2003.13590
- Li, X., Guang, Y., Wu, L., & Zhang, Y. (2015, 23–25 May 2015). *An offensive and defensive expect minimax algorithm in EinStein Würfelt Nicht!* Paper presented at the The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China.

- Li, X., Lv, Z., Wang, S., Wei, Z., & Wu, L. (2019). A Reinforcement Learning Model Based On Temporal Difference Algorithm. *IEEE Access*(7), 121922–121930. doi:10.1109/ACCESS.2019.2938240
- Li, X., Lv, Z., Wu, L., Zhao, Y., & Xu, X. (2020). Hybrid Online and Offline Reinforcement Learning for Tibetan Jiu Chess. *Complexity*, 2020(2), 1–11. doi:10.1155/2020/4708075
- Li, X., Wang, S., Lv, Z., Li, Y., & Wu, L. (2018). Strategy research based on chess shapes for Tibetan JIU computer game. *ICGA journal*, 40(2), 1–11. doi:10.3233/ICG-180058
- Li, X., Zhang, Y., Wu, L., Chen, Y., & Liu, B. (2023). *An Online Game Platform for Intangible Cultural Heritage Tibetan Jiu Chess*. Paper presented at the 2023 Chinese Intelligent Automation Conference, CIAC 2023, Springer, Singapore.
- Liaw, C., Wang, W. H., Tsai, C. T., Ko, C. H., & Hao, G. (2013). EVOLVING A TEAM IN A FIRST-PERSON SHOOTER GAME BY USING A GENETIC ALGORITHM. *Applied Artificial Intelligence*, 27(1-4), 199–212. doi:10.1080/08839514.2013.768883
- Lichtenstein, D., & Sipser, M. (1980). GO Is Polynomial-Space Hard. *Journal of the Acm*, 27(2), 393–401. doi:10.1145/322186.322201
- Lieberum, J. (2005). *An evaluation function for the game of amazons*. Paper presented at the Advances in Computer Games.

- Lorentz, R. (2016). Using evaluation functions in Monte-Carlo Tree Search. *Theoretical Computer Science*, 644, 106–113.
doi:<https://doi.org/10.1016/j.tcs.2016.06.026>
- Lu, H., & Xia, Z. (2008, 29–9–2008). *AWT: Aspiration with Timer Search Algorithm in Siguu*. Paper presented at the International Conference on Computers and Games.
- Marsland, T. A., & Campbell, M. (1982). Parallel Search of Strongly Ordered Game Trees. *ACM Computing Surveys*, 14(4), 533–551.
doi:10.1145/356893.356895
- Marsland, T. A., Reinefeld, A., & Schaeffer, J. (1987). Low overhead alternatives to SSS. *Artificial Intelligence*, 31(2), 185–199.
doi:10.1016/0004-3702(87)90019-1
- Matsubara, H., Iida, H., Grimbergen, R., & al., e. (1997). Chess, Shogi, Go, natural developments in game research. *Icca Journal*(19), 103–112.
- Mehdi, K. E., HoumairiAdam, Miriman, E., & BellafkihMostafa. (2024). Enhancing cloud security: harnessing bayesian game theory for a dynamic defense mechanism. *Cluster Computing*, 27(9), 12509 – 12526. doi:10.1007/s10586-024-04604-
- Mei, X., Zhang, D., Sun, D., Zhang, B., & Zhang, H. (2014). *The computer game model of Dou Dizhu*. Paper presented at the 2014 26th Chinese Control And Decision Conference (CCDC).

- Meltzoff, A. N., Kuhl, P. K., Movellan, J., & Sejnowski, T. J. (2009). Foundations for a New Science of Learning. *Journal of Distance Education*, 325(5938), 284–288. doi:10.1126/science.1175626
- Meng, Y., McAvoy, A., & Li, A. (2025). Promoting collective cooperation through temporal interactions. *Proc Natl Acad Sci U S A*, 122(26), e2509575122. doi:10.1073/pnas.2509575122
- Merkle, F., Samsinger, M., Schöttle, P., & Pevny, T. (2024). On the Economics of Adversarial Machine Learning. *IEEE Transactions on Information Forensics and Security*, 19, 4670–4685. doi:10.1109/TIFS.2024.3379829
- Michael, Bowling, Neil, Burch, Johanson, Oskari, & Tammelin. (2015). Computer science. Heads-up limit hold'em poker is solved. *Science*, 347(6218), 145–149. doi:10.1126/science.1259433
- Ming, Y. H., & Tsai, S. C. (2007). *On the fairness and complexity of generalized k -in-a-row games* (Vol. 385): Elsevier Science Publishers Ltd.
- Mitchell, T. M. (2003). *Machine Learning*. New York: McGraw-Hill.
- Moore, E. F. (1959). *The Shortest Path Through a Maze*. Paper presented at the Proc. of the International Symposium on the Theory of Switching.

- Moravík, M., Schmid, M., Burch, N., Lis, V., & Bowling, M. (2017). DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. *Science*, 356(6337), 508. doi:10.1126/science.aam6960
- None. (1986). Depth-first iterative-deepening: An optimal admissible tree search: R.E. Korf [Artificial Intelligence 27 (1) (1985) 97–109]. *Artificial Intelligence*, 28(1), 123. doi:10.1016/0004-3702(86)90035-4
- Panchal, M. R., Dodia, C. J., & Rathod, P. D. (2023). *Game Machine and Algorithm towards Trends in Game States using Machine Learning and Deep Learning*. Paper presented at the 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India.
- Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3), 241–288. doi:10.1016/0004-3702(86)90072-X
- Plaat, A., Schaeffer, J., Pijls, W., & Bruin, A. (1994). $SSS^* = \text{Alpha-Beta} + TT$. Retrieved from
- Pomerol, J. C. (1997). Artificial intelligence and human decision making. *European Journal of Operational Research*, 99(1), 3–25. doi:10.1016/S0377-2217(96)00378-5
- Qiu, H., Gao, F., Wang, Y., Song, X., & Sun, Y. (2013, 25–27 May 2013). *Implementation of common platform for two-person game*. Paper

- presented at the 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, China.
- R.S., 美. 米., S., 美. M. R., Ivan, 美. B., & Miroslav, 美. K. (2004). *机器学习与数据挖掘: 方法和应用*: 电子工业出版社.
- Raiko, T., & Peltonen, J. (2008, 01/01). *Application of UCT Search to the Connection Games of Hex, Y, *Star, and Renkula*. Paper presented at the Proc. of the Finnish Artificial Intelligence Conference (STeP 2008), Espoo, Finland.
- Reinefeld, A., & Marsland, T. A. (1994). Enhanced Iterative-Deepening Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7), 701–710. doi:10.1109/34.297950
- Reisch, S. (1980). Gobang is PSPACE-complete. *Acta Informatica*, 13(1), 59–66. doi:10.1007/BF00288536
- Robson, J. M. (2006). N by N Checkers is Exptime Complete. *SIAM Journal on Computing*, 13(2), 252–267. doi:10.1137/0213018
- Rutko, D. Fuzzified Algorithm for Game Tree Search with Statistical and Analytical Evaluation. *Computer Science and Information Technologies*, 770, 90–111.
- Sakuta, M., Hashimoto, T., Nagashima, J., Uiterwijk, J. W. H. M., & Iida, H. (2003). Application of the killer-tree heuristic and the lambda-search method to lines of action. *Information*

- Sciences—informatics & Computer Science An International Journal*, 154(3-4), 141–155. doi:10.1016/S0020-0255(03)00047-1
- Salakhutdinov, & Ruslan. (2012). An Efficient Learning Procedure for Deep Boltzmann Machines. *Neural Computation*, 24(8), 1967–2006. doi:10.1162/NECO_a_00311
- Salamanca, A. (2020). On the values of Bayesian cooperative games with sidepayments. *Mathematical Social Sciences*, 108, 38–49. doi:<https://doi.org/10.1016/j.mathsocsci.2020.09.002>
- Schaeffer, J. (1989). The history heuristic and alpha-beta search enhancements in practice. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 11(11), 1203–1212. doi:10.1109/34.42858
- Schaeffer, J. (2000). The games computers (and people) play. *Advances in Computers*, 52(00), 189–266. doi:10.1016/S0065-2458(00)80019-4
- Schaeffer, J. (2010). Game Over: Black to Play and Draw in Checkers. *ICGA journal*, 30(4), 187–197. doi:10.1007/s10664-007-9047-3
- Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Mueller, M., Lake, R., . . . Sutphen, S. (2007). Checkers Is Solved. *Science*, 317(5844), 1518–1522. doi:10.1126/science.1144079
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. doi:10.1016/j.neunet.2014.09.003
- Schnizlein, D., Bowling, M. H., & Szafron, D. (2009). *Probabilistic State Translation in Extensive Games with Large Action Sets*. Paper

presented at the International Joint Conference on Artificial Intelligence.

Schölkopf, B., & Platt, J., Hofmann, T. (2007). Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*(19), 153–160.

Shabankareh, S. G., & Shabankareh, S. G. (2019). *Improvement of Edge-Tracking Methods using Genetic Algorithm and Neural Network*. Paper presented at the 2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Shahrood, Iran.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Driessche, G. V. D., . . . Lanctot, M. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489. doi:10.1038/nature16961

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., . . . Graepel, T. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419), 1140–1144. doi:10.1126/science.aar6404

Silver, D., Schrittwieser, J., Simonyan, K., & al, e. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676), 354–359. doi:10.1038/nature24270

Sipser, M. (2006). *Introduction to the Theory of Computation(Second Edition)*: China Machine Press.

Stockmeyer, L. J., & Chandra, A. K. (2006). Provably Difficult Combinatorial Games : SIAM Journal on Computing: Vol. 8, No. 2 (Society for Industrial and Applied Mathematics). *Society for Industrial & Applied Mathematics*, 8(2), 151–174.

Su, C., Li, Y., Li, X., Chen, Y., & Fan, X. (2024). *A Nested Three-Stage Game Algorithm Based on Chess Shape Evaluation for Tibetan Jiu Chess*. Paper presented at the 2024 IEEE 48th Annual Computers,Software and Applications Conference(COMPSAC), Osaka,Japan.

<https://doi.org/10.1109/COMPSAC61105.2024.00104>

Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks*, 9(5), 1054. doi:10.1109/TNN.1998.712192

Takeuchi, S., Kaneko, T., & Yamaguchi, K. (2010). Evaluation of Game Tree Search Methods by Game Records. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2(4), p.288–302. doi:10.1109/tciaig.2010.2102022

Technicolor, T. S., Related, S. O. R., Technicolor, T. S., & Related, S. O. R. ImageNet Classification with Deep Convolutional Neural Networks [50].

- Teófilo, L. F., & Reis, L. P. (2011). *Building a no limit texas hold'em poker agent based on game logs using supervised learning*. Paper presented at the Autonomous and Intelligent Systems, Springer, Berlin, Heidelberg.
- Troillet, L., & Matsuzaki, K. (2024). Evaluating the Influence of Imperfect Information in Geister Using DREAM Trained Agents. *IEEE Transactions on Games*, 16(3), 598–610. doi:10.1109/TG.2023.3324737
- Tütüncüoğlu, F., & Dán, G. (2024). Optimal Service Caching and Pricing in Edge Computing: A Bayesian Gaussian Process Bandit Approach. *IEEE Transactions on Mobile Computing*, 23(1), 705–718. doi:10.1109/TMC.2022.3221465
- Volodymyr, M., Koray, K., David, S., & al, e. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. doi:10.1038/nature14236
- Wagner, J., & Virág, I. (2001). Solving Renju. *J. Int. Comput. Games Assoc.*, 24, 30–35. doi:10.3233/ICG-2001-24104
- Wang, J., Li, S., Chen, J., Wei, X., & Xu, X. (2011). *4*4-Pattern and Bayesian Learning in Monte-Carlo Go*. Paper presented at the Advances in Computer Games.

- Wang, X., Wang, Y., & Liu, T. (2013). *Research of einstein würfelt nicht! evaluation algorithm*. Paper presented at the 中国控制与决策会议.
- Wang, X., Zhang, H., Wang, Y., Qi, J., & Zhou, B. (2014). *Research of Einstein w ü rfelt nicht! Monte Carlo Algorithm Based on Optimized Evaluation*. Paper presented at the 中国控制与决策会议.
- Wang, Y., & Gelly, S. (2007). *Modifications of UCT and sequence-like simulations for Monte-Carlo Go*. Paper presented at the IEEE Symposium on Computational Intelligence & Games.
- Weihe, Y., Hongkun, Q., Yajie, W., Jianning, Q., Fei, L., & Yuxia, S. (2015). *Application of referee-player role model to the platform of computer games*. Paper presented at the 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China.
- Wu, I. C., & Huang, D.-Y. (2006, 2006). *A New Family of k-in-a-Row Games*. Paper presented at the Advances in Computer Games, Berlin, Heidelberg.
- Wu, I. C., Huang, D. Y., & Chang, H. C. (2005). Connect6. *ICGA journal*, 28(4), 235–242. doi:10.1007/s00453-006-0070-3
- Xian, M., Dianlong, Z., Qijun, L., Yanxin, J., Hongbo, Z., & Nana, Z. (2013). *Research on the platform system of military chess*

- computer game. *IEEE*, 1754–1757.
doi:10.1109/CCDC.2013.6561216
- Xinyang, Wang, Xiali, Li, Li, cheng, & Wu. (2016). *The Design and Implementation of Tibetan-go Game Software*. Paper presented at the 第 28 届中国控制与决策会议.
- Xu, G., Li, X. L., & Zhang, Y. Y. (2024). *Online gaming platform for Tibetan Jiu chess: AIvs-AI and human-vs-human*. Paper presented at the 2024 IEEE 48th Annual Computers,Software and Applications Conference(COMPSAC), Osaka,Japan.
- Yang, S., Barlow, M., Townsend, T., Liu, X., Samarasinghe, D., Lakshika, E., . . . Turnbull, B. (2023). Reinforcement Learning Agents Playing Ticket to Ride—A Complex Imperfect Information Board Game With Delayed Rewards. *IEEE Access*, 11, 60737–60757.
doi:10.1109/ACCESS.2023.3287100
- Yuan, H., Xu, Z., Tan, Z., Yi, X., Guang, M., Long, K., . . . Zhao, B. (2025). MARS: Reinforcing Multi-Agent Reasoning of LLMs through Self-Play in Strategic Games. *arXiv preprint arXiv:2510.15414*. doi:10.48550/arXiv.2510.15414
- Zha, D. C., Xie, J. R., Ma, W. Y., Zhang, S., Lian, X. R., Hu, X., . . . al, e. (2021). *Douzero: Mastering doudizhu with self-play deep reinforcement learning*. Paper presented at the the 38th International Conference on Machine Learning(PMLR2021).

- Zhang, J., Wang, X., Lin, J., & Xu, Z. (2008). *UCT Algorithm in Imperfect Information Multi-Player Military Chess Game*. Paper presented at the Proceedings of the 11th Joint Conference on Information Sciences (JCIS 2008).
- Zinkevich, M., Johanson, M., Bowling, M., & Piccione, C. (2007). *Regret minimization in games with incomplete information*. Paper presented at the Proceedings of the Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada.
- 蔡晔, 陈. 毛. (2015). 高性能计算及其相关新兴技术. *深圳大学学报: 理工版*, 32(1), 25 - 31. doi:10.3724/SP.J.1249.2015.01025
- 曹婷婷. (2009). 基于多核处理器串行程序并行化改造和性能优化. 西南交通大学,
- 柴化云, & 王福成. (2020). 基于深度网络的多人机器博弈算法研究. *信息技术与信息化*(12), 213 - 216.
- 陈国良, 孙广中, 徐云, & 龙柏. (2009). 并行计算的一体化研究现状与发展趋势. *科学通报* (8), 1043 - 1049. doi:CNKI:SUN:KXTB.0.2009-08-010
- 陈洪放, 王秋红, 顾晶晶, & 张凯. (2025). 基于策略优化和表征搜索的改进多智能体进化强化学习. *计算机系统应用*, 1 - 13. doi:10.15888/j.cnki.csa.010014

- 程克非, 张聪, & 沈一栋. (2005). 计算机桥牌双明手解的 Hash 表改进. *重庆大学学报*, 28(12), 45 - 47. doi:10.3969/j.issn.1000-582X.2005.12.013
- 邓颂庭. (2017). *久棋博弈原型系统及多媒体课件设计与实现*. (Master). 中央民族大学, 北京.
- 董慧颖, & 王杨. (2017). 多种搜索算法的五子棋博弈算法研究. *沈阳理工大学学报*, 36(02), 39 - 43+83.
- 付强, & 陈焕文. (2007). 中国象棋人机对弈的自学习方法研究. *计算机技术与发展*(12), 76 - 79.
- 高锦涛, 胡志远, & 姜璐璐. (2025). 基于蒙特卡洛树搜索的参数调优方法. *计算机应用*, 1 - 16. Retrieved from <https://link.cnki.net/urlid/51.1307.TP.20251015.1427.004>
- 郭琴琴, 李淑琴, & 包华. (2012). 亚马逊棋机器博弈系统中评估函数的研究. *计算机工程与应用*, 48(34), 50 - 54. doi:CNKI:SUN:JSGG.0.2012-34-010
- 郭圣明, 贺筱媛, 胡晓峰, 吴琳, & 欧微. (2016). 军用信息系统智能化的挑战与趋势. *控制理论与应用*, 33(12), 1562 - 1571. doi:10.7641/CTA.2016.60470
- 郭潇逍, 李程, & 梅俏竹. (2016). 深度学习在游戏中的应用. *自动化学报*, 42(5), 676 - 684. doi:CNKI:SUN:MOTO.0.2016-05-007

- 郭晓霞, 韩燮, & 赵融. (2018). 基于知识库的象棋机器博弈搜索算法研究. *中国科技论文*, 13(20), 2394 - 2400. doi:CNKI:SUN:ZKZX.0.2018-20-018
- 和力, & 吴丽贤. (2012). 斜方棋机器博弈程序的设计与实现. *现代计算机 (普及版)* (17), 47 - 50.
- 贺泽亚. (2023). 斗地主计算机博弈叫牌及农民角色博弈关键算法的研究. (硕士). Available from Cnki
- 胡强, 谢显中, & 张小川. (2014). 一个幻影围棋计算机博弈系统的设计与实现. *数字通信*(1), 6.
- 吉根林. (2004). 遗传算法研究综述. *计算机应用与软件*, 21(2), 69 - 73. doi:10.3969/j.issn.1000-386X.2004.02.032
- 计永昶, 丁卫群, 陈国良, & 安虹. (2001). 一种实用的并行计算模型. *计算机学报*, 24(4), 437 - 441. doi:10.3321/j.issn:0254-4164.2001.04.016
- 焦尚彬, & 刘丁. (2010). 博弈树置换表启发式算法研究. *计算机工程与应用*, 46(6), 42 - 45. doi:10.3778/j.issn.1002-8331.2010.06.012
- 雷捷维, 王嘉旻, 任航, 闫天伟, & 黄伟. (2021). 基于 Expectimax 搜索与 Double DQN 的非完备信息博弈算法. *计算机工程*, 47(3), 304 - 310+320. doi:10.19678/j.issn.1000-3428.0057309
- 李飞, 王亚杰, 尹航, & 孙玉霞. (2016). 基于幻影围棋的 C 语言课程教学案例设计. *计算机教育* (10), 117-119. doi:10.3969/j.issn.1672-5913.2016.10.035

- 李淑琴, & 冯浩东. (2022). 牌型预测与蒙特卡洛模拟结合的麻将博弈策略. *重庆理工大学学报(自然科学)*, 36(12), 148 - 154.
- 李淑琴, 李静波, 韩裕华, & 陶思拓. (2012). 苏拉卡尔塔博弈系统中评估函数的研究. *北京信息科技大学学报(自然科学版)*, 27(06), 42 - 45+61. doi:10.16508/j.cnki.11-5866/n.2012.06.008
- 李霞丽, 陈彦东, 杨子熠, 张焱垠, & 吴立成. (2022). 藏族久棋的一种两阶段计算机博弈算法. *重庆理工大学学报: 自然科学*, 36(12), 110 - 120.
- 李霞丽, 王昭琦, 刘博, & 吴立成. (2023). 麻将博弈 AI 构建方法综述. *智能系统学报*, 18(06), 1143 - 1155.
- 李霞丽, 吴立成, & 李永集. (2018). 基于棋型的藏族"久"棋计算机博弈研究. *智能系统学报*, 13(4), 577 - 583. doi:10.11992/tis.201609023
- 李学俊, 王小龙, 吴蕾, & 刘慧婷. (2015). 六子棋中基于局部“路”扫描方式的博弈树生成算法. *智能系统学报*, 10(2), 267 - 272. doi:10.3969/j.issn.1673-4785.201401022
- 李之棠, & 陈华民. (1998). 博弈树并行搜索算法. *小型微型计算机系统*, 19(10), 53 - 56.
- 李志敏, 罗里波, & 李祥. (2008). 完全二叉树理论的计算复杂度. *数学学报: 中文版*, 51(2), 311 - 318. doi:CNKI:SUN:SXXB.0.2008-02-013

- 连莲, 徐心和, 张雪峰, & 颜宁. (2009). 点点连格棋机器博弈系统关键技术分析. Paper presented at the 中国人工智能学会第十三届学术年会, 北京.
- 梁凯. (2022). 基于改进 UCT 的藏久棋博弈技术研究. (Master). 沈阳航空航天大学, 沈阳.
- 刘剑锋, 普杰信, & 孙力帆. (2023). 融合对比预测编码的深度双 Q 网络. 计算机工程与应用, 59(06), 162 - 170. Retrieved from <https://link.cnki.net/urlid/11.2127.TP.20211125.1152.006>
- 刘溜, 张小川, 彭丽蓉, 田震, 万家强, & 任越. (2022). 一种结合策略价值网络的五子棋自博弈方法研究. 重庆理工大学学报(自然科学), 36(12), 129 - 135+120.
- 刘强. (2017). 藏棋——"久棋"源考. 西藏研究(6), 105 - 109. doi:10.3969/j.issn.1000-0003.2017.06.014
- 刘知青, & 吴修竹. (2016). 解读 AlphaGo 背后的人工智能技术. 控制理论与应用, 33(12), 1685 - 1687. doi:10.7641/CTA.2016.60526
- 刘子建, 胡德宏, 赵文浩, & 吴佳明. (2022). 基于幻影围棋的不完全信息博弈系统研究. 电脑与信息技术, 30(6), 25 - 29.
- 卢风顺, 宋君强, 银福康, & 张理论. (2011). CPU/GPU 协同并行计算研究综述. 计算机科学, 38(3), 5 - 9. doi:10.3969/j.issn.1002-137X.2011.03.002

- 罗俊仁, 张万鹏, 苏炯铭, 王尧, & 陈璟. (2023). 面向智能博弈的决策 Transformer 方法综述. *指挥与控制学报*, 9(01), 9 - 22. Retrieved from https://kns.cnki.net/kcms2/article/abstract?v=iSoVlldxB1O-v_5r8jDycTy8DNcjLwsHX6SRpBKe1fdYJLpA7ct-OiTjp153UTjPDOfKXcoK_gELwUGAnTW-Yg5pMAw1C1An0LtkUfAgfQ2_FaRsw77uIyGDCK3f-rGSaqqeTIW-ruw8erZQXN7mCd1oAmIVG4r__E3P801Z3yVnBHZ0J6WhZWzYM9dnl2K&uniplatform=NZKPT&language=CHS
- 吕艳辉, & 宫瑞敏. (2012). 计算机博弈中估值算法与博弈训练的研究. *计算机工程*, 38(11), 163 - 166. doi:10.3969/j.issn.1000-3428.2012.11.050
- 吕征宇. (2020). *基于深度强化学习的棋类博弈研究*. 中央民族大学, 马少龙, & 尚涛. (2017). 藏棋文化的传承和保护. *西藏研究*(4), 157 - 160. doi:10.3969/j.issn.1000-0003.2017.04.021
- 马骁, 王轩, & 王晓龙. (2010). 一类非完备信息博弈的信息模型. *计算机研究与发展*, 47(12), 2100 - 2109. doi:CNKI:SUN:JFYZ.0.2010-12-015
- 梅险, 孙德强, 杜世锦, 王陈云, 曾凡夫, & 臧朝蕊. (2014). *非完全信息博弈游戏的实践研究*. Paper presented at the 2013 年中国信息通信研究新进展论文集.

- 聂慧, 彭娇, 金晶, & 李康顺. (2017). 基于多核系统的并行线性 RankSVM 算法. *计算机应用研究*, 34(1), 46 - 51. doi:10.3969/j.issn.1001-3695.2017.01.009
- 邱虹坤. (2013). *具有完备信息的非合作计算机博弈系统的研究与实现*. (硕士). Available from Cnki
- 邱虹坤, 王浩宇, & 王亚杰. (2022). Q 学习实现亚马逊棋评估函数自调参. *重庆理工大学学报: 自然科学*, 36(12), 136 - 141.
- 邱虹坤, 郑晓东, & 王亚杰. (2021). 基于数据库和经验分析的桥牌混合策略打牌模型. *重庆理工大学学报: 自然科学*, 35(12), 134 - 139+154. doi:10.3969/j.issn.1674-8425(z).2021.12.017
- 沈强望, 丁濛, 杜文涛, & 赵文龙. (2021). 对藏棋"久"的分阶段算法研究. *智能计算机与应用*, 11(2), 88 - 92.
- 苏攀, 王熙照, & 李艳. (2011). 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. *计算机研究与发展*, 48(5), 841 - 847. doi:CNKI:SUN:JFYZ.0.2011-05-019
- 唐杰, 许华虎, & 谈广云. (2020). 面向智能博弈游戏的卷积神经网络估值方法. *计算机应用与软件*, 37(7), 259 - 265. doi:10.3969/j.issn.1000-386x.2020.07.043
- 田大江, 吴志刚, 曾庆华, 叶宵宇, 任恩泽, 宋甫俊, & 郭运伟. (2024). 干扰生成对抗强化学习位置跟踪控制器. *小型微型计算机系统*(5), 1 - 11.

- 仝秉达, 段海滨, & 魏晨. (2024). 仿鹰鸽捕食逃逸行为的多无人机分组对抗博弈方法. *控制理论与应用*, 41(5), 855 - 865.
- 王骄, 孙英龙, 吕辉展, & 杨辉. (2012). 中国象棋中亚洲棋规循环问题的解决. *东北大学学报: 自然科学版*, 33(4), 476 - 481.
- 王骄, 王涛, 罗艳红, & 徐心和. (2005). 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. *东北大学学报: 自然科学版*, 26(10), 949-952. doi:10.3321/j.issn:1005-3026.2005.10.009
- 王骄, 王., 罗艳红, 徐心和. (2005). 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. *东北大学学报*(10), 949 - 952.
- 王京辉, & 乔卫民. (2005). 基于 PVM 的博弈树的网络并行搜索. *计算机工程*, 31(9), 29 - 31. doi:10.3969/j.issn.1000-3428.2005.09.011
- 王琳蒙, 王玉惠, 陈谋, & 刘昊天. (2022). 基于改进麻雀算法的非完备信息博弈策略研究. *吉林大学学报: 信息科学版*, 40(4), 589 - 599. doi:10.19292/j.cnki.jdxpx.2022.04.024
- 王璐瑶, & 吴蕾. (2025). 基于深度神经网络的桥牌叫牌策略研究. *应用科技*, 52(01), 198 - 204. Retrieved from <https://link.cnki.net/urlid/23.1191.U.20241018.1059.004>
- 王骥, & 孙建伶. (2008). 基于优化迭代的博弈树算法. *计算机应用与软件*, 25(2), 228 - 230. doi:10.3969/j.issn.1000-386X.2008.02.083
- 王世尧. (2023). “久棋”的计算机博弈算法研究与实现. (Master). 中央民族大学, 北京.

- 王书宇, 李龙澍, & 汪群山. (2008). 改进的协同进化遗传算法在机器博弈中的应用. *计算机技术与发展*, 18(12), 4 - 7.
doi:10.3969/j.issn.1673-629X.2008.12.002
- 王帅, & 雷跃明. (2017). 一种德州扑克的牌力评估方法. *计算机工程与科学*, 39(7), 1352 - 1358. doi:CNKI:SUN:JSJK.0.2017-07-023
- 王松. (2018). *久棋强化学习博弈研究及多媒体课件开发*. (Master). 中央民族大学, 北京.
- 王鑫. (2024). *斗地主博弈智能体的深度强化学习方法研究*. (硕士). Available from Cnki
- 王亚杰, 乔继林, 梁凯, & 谢延延. (2022). 结合先验知识与蒙特卡罗模拟的麻将博弈研究. *智能系统学报*, 17(01), 69-78.
- 王亚杰, 邱虹坤, 吴燕燕, 李飞, & 杨周凤. (2016). 计算机博弈的研究与发展. *智能系统学报*, 11(6), 788 - 798.
doi:10.11992/tis.201609006
- 魏钦刚, 王骄, 徐心和, & 南晓斐. (2007). 中国象棋计算机博弈开局库研究与设计. *智能系统学报*, 2(1), 85 - 89.
doi:10.3969/j.issn.1673-4785.2007.01.013
- 魏印福, & 李舟军. (2019). 动态规划求解中国象棋状态总数. *智能系统学报*, 14(01), 108 - 114.
- 吴立成, 吴启飞, 钟宏鸣, 王世尧, & 李霞丽. (2023). 基于卷积神经网络的"拱猪"博弈算法. *智能系统学报*, 18(4), 775 - 782.

- 徐方婧, 魏鲲鹏, 王以松, 彭啟文, & 于小民. (2020). 基于卷积神经网络的“斗地主”策略. *计算机与现代化*(11), 28 - 32.
- 徐心和, 邓志立, 王骄, 徐长明, 刘纪红, & 马宗民. (2008). 机器博弈研究面临的各种挑战. *智能系统学报*, 3(4), 288 - 293.
doi:10.3969/j.issn.1673-4785.2008.04.002
- 徐心和, & 王骄. (2006). 中国象棋计算机博弈关键技术分析. *小型微型计算机系统*(06), 961 - 969.
- 徐长明. (2010). 基于连珠模式的六子棋机器博弈关键技术研究. 东北大学,
- 徐长明, 马宗民, & 徐心和. (2009). 一种新的连珠棋局面表示法及其在六子棋中的应用. *东北大学学报: 自然科学版*, 30(4), 514 - 517.
- 徐志凡, 李媛, 王静文, 李卓轩, & 曹一丁. 一种 Q 学习制作海克斯棋开局库方法. *南通大学学报(自然科学版)*, 1 - 7.
- 严小卫, 莫. (1999). 智能五子棋的设计与实现. *广西师范大学学报(自然科学版)*(04), 11 - 14.
- 杨冰. (2015). 多 CPU 多核系统上的并行预处理算法. 中国工程物理研究院,
- 衣御寒, 王亚杰, 吴燕燕, 刘松, 张兴慧, & 蒋传禹. (2024). 结合 A2C 和手牌估值方法的麻将博弈研究. *重庆理工大学学报(自然科学)*, 38(05), 154 - 161.

- 张加佳. (2016). 非完备信息机器博弈中风险及对手模型的研究. 哈尔滨工业大学,
- 张利群. (2016). 实现苏拉卡尔塔棋网络博弈平台的吃子算法. 计算机工程与应用, 52(7), 62 - 66.
doi:10.3778/j.issn.1002-8331.1508-0122
- 张明亮, & 李凡长. (2009). 一种新的博弈树搜索方法. 山东大学学报(工学版), 39(6), 1 - 7.
- 张明亮, 吴俊, & 李凡长. (2012). 五子棋机器博弈系统评估函数的设计. 计算机应用, 32(07), 1969 - 1972+1990.
- 张芃芃, 孟坤, & 杨震栋. (2020). 基于强化学习的海克斯棋博弈算法研究与实现. 智能计算机与应用, 10(03), 142 - 145.
- 张千, 王庆玮, 张悦, 纪校锋, 张宇翔, 祝赫, & 赵昌志. (2019). 基于深度学习的文本特征提取研究综述. 计算机技术与发展, 29(12), 61 - 65. Retrieved from <https://link.cnki.net/urlid/61.1450.TP.20190924.1534.006>
- 张小川, 陈光年, 张世强, 孙可均, & 李祖枢. (2010). 六子棋博弈的评估函数. 重庆理工大学学报: 自然科学, 24(2), 64 - 68.
doi:CNKI:SUN:CGGL.0.2010-02-016
- 张小川, 刘溜, 陈龙, & 涂飞. (2021). 一种非遗藏族久棋项目计算机博弈智能体的评估方法. 重庆理工大学学报: 自然科学, 35(12), 119 - 126.

- 张小川, 严明珠, 涂飞, 陈俊宇, & 魏乐天. (2024). 一种大众麻将计算机博弈的快速出牌方法. *重庆理工大学学报(自然科学)*, 38(05), 102 - 107.
- 张晓倩, 黄鸿, & 程子豪. (2019). 基于蒙特卡洛树搜索的优化算法在幻影围棋中的应用. Paper presented at the 第31届中国控制与决策会议(CCDC2019), 中国, 南昌.
- 张志刚. (2015). 双明手桥牌打牌辅助分析程序研究. *现代计算机*(23), 47 - 50. doi:10.3969/j.issn.1007-1423.2015.34.013
- 赵海璐. (2022). 大众麻将计算机博弈智能搜索算法的应用研究. (硕士). Available from Cnki
- 郑巍, 汤佳豪, 熊小平, & 樊鑫. (2025). 基于非对称自博弈的非均势空战智能决策方法. *系统仿真学报*, 1 - 14. doi:10.16182/j.issn1004731x.joss.25-0621
- 中国人工智能学会机器博弈专业委员会. (2014a). 五子棋 (Connect5) 项目规则. Retrieved from <http://computergames.caa.cn/jsgz16.html>
- 中国人工智能学会机器博弈专业委员会. (2014b). 亚洲象棋规则. Retrieved from <http://computergames.caa.cn/jsgz11.html>
- 周超, 胡晓峰, 郑书奎, & 夏荣祥. (2017). 战略战役兵棋演习系统兵力聚合问题研究. *指挥与控制学报*, 3(1), 19 - 26. doi:10.3969/j.issn.2096-0204.2017.01.0019

- 周雷, 尹奇跃, & 黄凯奇. (2022). 人机对抗中的博弈学习方法. *计算机学报*, 45(09), 1859 - 1876. Retrieved from https://kns.cnki.net/kcms2/article/abstract?v=iSoVlldxB1M4Lx4XJ9Md98Xqsw0qhfszs8mKPJ5Uc73VRwt6NKifNgGtqfoLTqrEtVJxoWQY5_WFAEwAv695VXMP6nrO2NWmvU-e-tdlHb3dqFnc44LkV2Wzq-fNUhfwb4qk90U-eS2GHWJAYtY75njtctkud-bWIOg3HcHEYqX2C3DHR3bKcRLt7uSW5x7&uniplatform=NZKPT&language=CHS
- 周玮, 张贇, & 周静怡. (2006). 基于对弈局势的二次估值方法. *系统仿真学报*(09), 2665 - 2668.
- 朱丰, & 胡晓峰. (2016). 基于深度学习的战场态势评估综述与研究展望. *军事运筹与系统工程*, 30(3), 22 - 27. doi:CNKI:SUN:JSYC.0.2016-03-004
- 朱舟, & 闵华松. 利用统计数据并行蒙特卡洛树搜索算法的中国象棋博弈. *计算机工程与应用*, 1 - 10.