

基于计算机博弈的计算机问题求解实践能力 培养探索*

蔡彪, 吴静, 朱星

成都理工大学计算机与网络安全学院,
成都 610059

摘要 为了提高计算机类专业学生的“计算机问题求解能力”, 该论文提出了一种基于机器博弈的计算机类学生计算机问题求解能力培养模式, 通过把中国计算机博弈大赛的棋种作为课程的题目, 设计多维度的过程考核方式, 着力培养学生的自主学习能力。论文以爱恩斯坦棋为例介绍案例的知识体系及三种不通难度要求的决策算法设计。教学改革实践的效果说明了所提出模式的有效性(见本文第5节), 同时论文也具有较强可推广性和可借鉴性。

关键字 计算机博弈, 计算机问题求解, 实践能力, 爱恩斯坦棋

Exploration on Practical Problem-Solving Abilities in Computing Based on Computer Game Playing*

Biao CAI, Jing Wu, Xing Zhu

College of computer &
cybernetic ,Chengdu University of
Technology,Chengdu,610059

Abstract—To advance the computational problem-solving competency of undergraduate students majoring in computer science and related disciplines, this paper puts forward a novel pedagogical framework for cultivating such competency rooted in machine game competitions. By leveraging the event categories from the National Chinese Computer Game Tournament as project assignments in relevant courses, and implementing a multi-dimensional process-oriented evaluation mechanism, the framework is dedicated to fostering students' self-directed learning capabilities. Taking the Einstein Chess as a concrete illustrative case, this paper elaborates on the case-specific knowledge architecture, along with the design of decision-making algorithms corresponding to three distinct tiers of difficulty requirements. The effectiveness of the proposed pedagogical model has been empirically validated via the teaching reform practices documented in Section 5 of this manuscript. Furthermore, the proposed framework demonstrates prominent generalizability and high reference value for analogous educational practices.

Keywords—Computer Game Playing, Computer Problem Solving, Practical Ability, Einstein Chess

1 引言

2016年作为人工智能诞生60周年, 围棋程序AlphaGo战胜世界围棋冠军李世石, 人工智能再次掀起研究热潮。2017年国务院发布的《新一代人工智能发展规划》中指出:“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”[1, 2], 这使得计算机博弈竞赛特别具有时代意义[3-6]。由中国人工智

能学会和教育部计算机类专业教指委共同主办全国大学生计算机博弈大赛已组织了16届, 辐射全国100多所高校, 竞赛水平逐年提升。本文将介绍我校将计算机博弈与计算机问题求解能力融合的实践案例。

我校在计算机类专业在学生“计算机问题求解能力”的培养方面经过多轮尝试, 在人工智能已经成为当前潮流的前提下, 2019年开始将以人工智能算法为基础计算机博弈内容融入到日常教学活动中。依托基于计算机博弈的《计算机问题求解》课程设计这一门计算机类专业的必修实践课程, 来加强计算机类专业学生在大二分流前的实践动手能力和人工智能思维能力。该课程定位为: 在学习程序基础、面向对象程序设计、离散数学和数据结构等专业课程学习后的第一次综合实践课程, 是一次全面考察所学知识和应变能力素质的有机融合, 第一次要求学生自主设计思路来解决复杂计算机问题求解的综合能力, 在课堂教学知识的基础上提升思维高度, 满足课程在综合知识和综

***基金资助:** 本文得到四川省一流实践课程(No. YLKC1369)、成都理工大学教学质量重点项目(No. JG212021)、四川省高阶课程(No. GJKC2025109)、四川省高等教育人才培养质量和教学改革项目“面向地学应用的跨学科通用人工智能创新人才培养模式探索与实践”(JG2024-0624 2024—2026)、成都理工大学2024—2026年高等教育人才培养质量和教学改革项目“智能时代产教深度融合新工科课程群体系探究与实践”资助。

*通讯作者: 蔡彪 caibiao@cdut.edu.cn。

合能力培养方面高阶性要求；同时需要探索课程中还没有学习过的人工智能算法来提升自己的开发程序的对抗能力，特别是鼓励同学在选择算法时，选用当前主流的开源算法整合进自己项目实践，将反映前沿性和时代性内容纳入课程实际，尤其在课程教学过程和考核方式上，彻底摒弃过去实践课程那种以交“作品+报告”来结课的传统方式，采用“比赛+答辩+平时考核+论文撰写”的考核方式，在教学内容和教学模式上具有较强的创新型。通过这门课程学习，计算机类专业学生在基础程序设计能力和算法分析能力方面培养方面会得到极大的提升，同时课程还特别注重调动同学的自主学习能力，需要学生通过“跳一跳”的挑战勇气[12, 15]，只有通过力所能及的新知识进行有益探索和补充才能够实现课程具有较高的完成度，课程的挑战度非常强。

这门课程经过几年的发展，对培养我校计算机类专业学生在基于计算机博弈项目开发演化的软件开发能力、算法设计能力和工程实践能力培养中起到了重要的作用，该项工作对我校计算机类专业的专业建设和内涵式发展起到了非常重要的支撑作用。因此，我校的基于机器博弈的《计算机问题求解》课程改革探索研究模型很好应用价值和借鉴意义。

2 实践案例课程建设

2.1 计算机博弈开发与计算机问题求解能力培养的要求一致性

计算机类专业学生的计算机问题求解实践能力的要素，通过本次项目建设形成一套计算机问题求解能力的解决方案。计算机问题求解能力培养包括软件开发能力、算法设计能力、工程实施能力、人机交互能力等。而计算机博弈程序开发恰好与计算机专业人才所必须具备的计算机问题求解能力培养相一致[7, 8]。因此，本项目拟通过将计算机博弈项目引入到课程设计中，从而实现计算机专业学生的计算机问题求解能力培养。

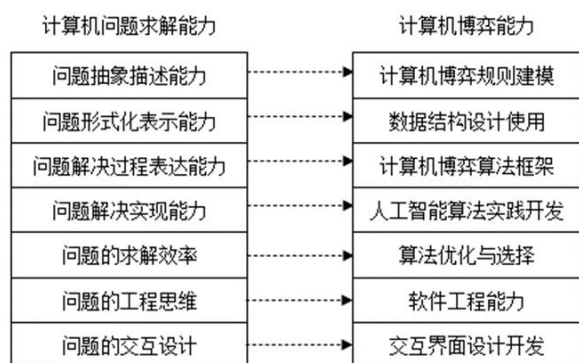


图1 计算机问题求解能力与计算机博弈能力对应关系

2.2 计算机博弈内容构建的高阶性

在进行课程内容调整前“计算机问题求解课程设计”实践内容设置存在问题主要体现在难度系数不易控制，内容比较传统，难以引起学生兴趣等问题，而计算机博弈大赛的项目设置种类繁多，不仅包括围棋、象棋、藏族久棋、五子棋、六子棋等，还包括一些西洋棋种，比如西洋跳棋、亚马逊棋等，这些棋种对多才多艺、求知欲很强的大学生来说，具有很大的吸引力。因此，根据博弈大赛项目构建《计算机问题求解课程设计》实践基础、提高、挑战多层次实验内容，如图2所示。

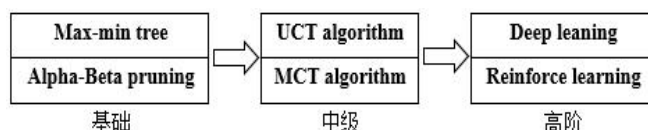


图2 不同难度算法选择

计算机博弈程序构造的内容难度包括了不同层次，基础算法如：极大-极小树、Alpha-Beta 剪枝算法；中级算法如：UCT 算法、蒙特卡洛模拟等；具有挑战度的高阶算法如：深度神经网络、强化学习等。学生可以根据实际情况，选择不同的实验内容。由于课程设置在大学第二学期，在该阶段任何一个算法都需要同学们去自主学习，通过学习获取成功来激励对学习信心和热爱。

2.3 课程内容和考核方式的创新性

传统的实践课程评估方式一般根据学生的完成情况、课堂表现、出勤率评估实验成绩。这种模式对综合性的实践课程并不科学和合理，因为实践课程还有很多过程性、课外性因素需要加以考虑[9-11]，因此本课程组在研究综合传统方式的考核机制的基础上设计了更加合理的评价机制，注重在学习过程中合理评价学生，培养一流人才，考核评价安排如图3所示。

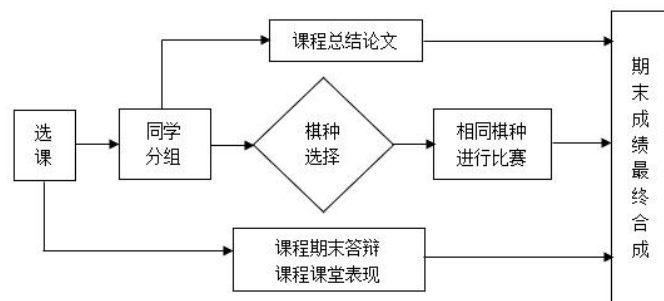


图3 课程整体成绩构成

为了使得过程评价和公平评价成为了更加重要的评价因素，课程组设置了激励型成绩评估机制除，根据学生选择的不同层次的实验内容设置激励系数，难度高则激励系数高，成绩系数就高；项目完成后，除了由教师验收每组同学开发的博弈程序的功能及性能外，同一项目不同程序之间通过博弈竞赛来确定程序

棋力成绩排名。这种新设计的激励型实验成绩评估机制能够极大地提高学生的实验热情, 锻炼其竞技能力也使得成绩评定更加合理, 并有利于选拔有兴趣和能力的同学进一步参加国赛。在整个项目式实践过程管理中, 对标工程认证中的使用现代工具、个人与团队、项目管理三个能力或素质培养要求。采用工程的方法和技术, 对项目过程进行管理。对学生小组的管理, 采用软件公司中流行的现代程序员组管理架构。学生自主结合, 成立一个项目小组, 成员不超过5人, 选出组长, 各成员之间进行明确分工。对项目开发过程的管理, 采用敏捷开发过程, 各个成员之间及时面对面沟通交流, 共同完成项目的系统分析、搜索算法和局面评估算法设计、落子策略等代码实现阶段。每个阶段都要按照规范编写文档, 教师对阶段性成果和文档进行评估。

2.4 新的课程内容对教师学生具有挑战性

棋类项目实验的博弈树的状态空间通常都很复杂, 因此算法的时间复杂度和空间复杂度都很高, 对计算机的硬件设备要求比一般实验的要求要高得多, 不仅对学生的专业能力提出了更高的要求, 对教师的技能也提出新的挑战。尤其是对于高级组算法的实验需要GPU服务器等设施 and 配套的实验环境才能顺利训练深度神经网络。在我们的课程的实践中, 对教师的能力也提出了新的要求。对实验室人员的要求不仅仅是以前的设备维护, 现在还包含了GPU环境的配置、各种Python版本控制和Pytorch框架的版本管理等内容; 对课程指导老师, 不仅仅只要求理论知识, 还需要完全参与学生的博弈程序开发中, 同时还要最追踪计算机博弈算法的前沿技术和框架。因此, 通过本次课程内容调整, 不仅仅是对学生学习能力、态度和热情的挑战, 同时也是对教师、对实验管理人员的个人能力的挑战。本课程在推动专业教师团队完成教学的同时, 科研水平也得到了提升。

为了更加清楚地介绍本实践课程内容设置在计算机问题求解能力培养中的知识概况, 本文以爱恩斯坦棋博弈程序开发为例进行介绍, 包括第三章和第四章两个部分: 第三章为博弈程序开发通用接口管理, 第四章则设计了对应高中低三种不同难度的博弈决策案例。

3 爱恩斯坦棋实现通用接口案例

3.1 棋盘状态管理接口

在开发博弈类人工智能系统时, AI 策略(如极大极小搜索、MCTS)需要频繁访问棋盘状态, 判断可以怎么走、执行一步落子、撤销一步、判断是否胜利等。所以要进行棋盘状态类管理, 负责记录。为了满足智能策略模块对棋盘状态的灵活调用与控制, 本实践系

统系统设计了Board 类承担了棋局初始化、状态更新、合法走法生成、胜负判断、悔棋回退等核心职责。该类面向底层棋盘数据结构与规则逻辑的统一封装, 是整个爱恩斯坦棋AI系统的状态控制中枢。其职责可分为以下六类:

(1) 棋局初始化与布局控制

本类支持棋盘初始化、开局布局设定及棋子状态重建等操作, 确保棋局在任意阶段均可复现与调试:

`initBoard()` 随机初始化棋盘, 包括红蓝双方棋子初始位置及其编号

`place_red_pos() / place_blue_pos()` 设置红蓝双方自定义顺序的初始棋子位置

`red_best_place() / blue_best_place()` 设置红蓝双方推荐的最优布局(基于启发式顺序)

`red_legal_place_pos / blue_legal_place_pos` 存储红蓝方初始合法放置点集合(二维坐标)

`update_pieces()` 扫描棋盘重构棋子列表, 用于悔棋后同步状态

(2) 合法走法生成与移动编码

系统采用四位整数编码(如 2031)来表示一次完整的移动(从(2,0)移至(3,1)), 并建立了合法移动规则集合供系统判断与搜索使用:

`get_available_pieces()` 根据当前骰子值, 返回当前玩家可移动的棋子编号列表

`get_available_moves()` 获取当前骰子和轮次下的全部合法移动(编码)

`red_legal_moves / blue_legal_moves` 红 / 蓝 双方全局允许的合法移动编码列表

`location_to_move(location)` 将棋子移动从坐标格式转换为编码(如 (2,0,3,1) → 2031)

`move_to_location(move)` 将移动编码还原为起始与目标位置坐标元组

(3) 棋子操作与回合推进

提供棋子的执行操作接口、落子逻辑处理以及当前行棋方控制, 用于模拟完整的落子步骤及状态推进:

`do_move(move)` 执行一次移动, 修改棋盘状态并交换轮次

`moveChess(from_pos, to_pos)` UI前端接口, 支持通过坐标执行移动

`setChess(position, value)` 在指定位置放置指定编号的棋子, 支持正负代表红蓝

`swapturn()` 在每步落子后自动切换当前执子方
`setDice(diceNum) / getDice()` 设置或获取当前轮到的骰子点数

`setturn(color) / getturn() / getturnStr()` 设置或查询当前轮到的玩家颜色与描述字符串

(4) 悔棋机制与状态回退

为支持AI搜索回溯与用户交互悔棋，系统实现了基于状态快照的悔棋功能：

`backupBoard()` 在每次落子前，对棋盘进行深拷贝并入栈备份

`undo()` 弹出最近一步棋局快照，回滚棋盘、轮次与点数

`boardBackup` 棋盘历史状态栈

`movements / points` 保存每步的移动编码与骰子点数序列

(5) 棋局终局判断与胜负判断

系统通过检查目标点是否被占据或棋子全灭来判断胜负归属，支持策略结束条件判定：

`checkWin()` 检查当前是否已分出胜负，返回 RED / BLUE / None

`checkBoardSet()` 检查开局阶段是否已经完成全部合法放置（共12枚）

`getChessSign(x, y)` 返回某坐标上棋子归属（红 / 蓝 / 空）

`getChess(x, y)` 获取指定位置的原始棋子编号（带符号）

(6) 状态表示与神经网络支持

为支持强化学习与策略网络的训练，系统提供了基于 `numpy` 的张量状态编码接口，用于模型输入：

`get_current_state()` 返回当前棋盘的 $4 \times 5 \times 5$ 状态张量，包括棋局、上一步、可动棋子和当前轮次

`save_initial_pos() / initial_pos_info()` 存储与获取初始布局的标准棋谱编码

`get_movements_info() / last_move_info()` 返回当前局面或最近一步的完整棋谱记录（标准格式）

3.2 静态价值评估函数设计

在智能博弈策略系统中，静态评估函数是非终局状态下进行局势判断的核心工具。其作用是接收当前棋盘状态与玩家信息，并输出一个连续型实数值，用于衡量当前玩家在该局面下的优势程度。为此，本文实现了基于规则启发的静态评估模块 `FenxiPinggu`，该模块融合了棋子的活跃度、位置价值、推进能力与局部威胁等多个维度，为搜索算法（如极大极小搜索与 MCTS）提供非终局节点的局势评分支持。该模块设计为纯粹函数式实现，完全独立于搜索策略流程。只需传入棋盘状态与当前玩家，即可返回评分值。它既可作为启发式引导器，又可作为模拟终止节点的近似评价器，具备良好的灵活性与可替换性。

(1) 评估函数的评分公式

评估函数的本质是一个状态到实数的映射：

$$Score = \alpha_1 \times Exp_{my} + \alpha_2 \times Exp_{oppo} + \alpha_3 \times Threat_{my} + \alpha_4 \times R \quad (1)$$

其中： Exp_{my} 表示当前玩家所有可动棋子所处格子的静态位置价值乘以其活跃度（可移动性）后的总和，代表进攻潜力； Exp_{oppo} 表示敌方所有可动棋子在当前位置的静态分值乘以活跃度的总和，作为负值体现敌方威胁； $Threat_{my}$ 表示我方棋子在当前位置上面面临敌方吃子的局部威胁值，数值越高说明越危险； R 是我方棋子接近胜利终点的潜力评分； $\alpha_1 \sim \alpha_4$ 是不同特征项的权重，视评分模式灵活调整。例如，在完整模式下，权重配置如下：

$$Score = 5 \times Exp_{my} + 5 \times Exp_{oppo} + 1 \times Threat_{my} + 2 \times R \quad (2)$$

而在快速模式（简化版本）中，可使用：

$$Score = 10 \times Exp_{my} + 5 \times Exp_{oppo} + 1 \times Threa \quad (3)$$

这种线性组合的评分结构使得策略在评估过程中可快速近似判断“当前形势是否利我”，从而合理排序节点或终止模拟。

(2) 评估函数核心代码实现

该模块的主要入口函数如下：

```
class FenxiPinggu:
    def getValue(self, board: BoardInterface, player: str,
                which=1) -> float:
        self.board = board
        self.player = player
        self.R = self.compute_advancement()
        self.Exp_my = self.compute_my_exp()
        self.Exp_oppo = self.compute_oppo_exp()
        self.Threat_my = self.compute_threat()
        if which == 0:
            return 10*self.Exp_my+5*self.
            Exp_oppo+1*self.Threat_my
        else:
            return 5*self.Exp_my+5*self.Exp_
            oppo+1*self.Threat_my+2*self.R
```

其中`compute_my_exp()`遍历我方棋子，计算可动性与位置得分；`compute_oppo_exp()`同理计算敌方威胁值；`compute_threat()`检查我方每枚棋子是否被对方威胁；`compute_advancement()`基于几何距离评估棋子推进终点的能力。这四个方法共同完成了棋局特征提取与量化，进而返回一个数值化评价。

(3) 位置价值矩阵设计

在爱因斯坦棋中，胜利条件为抵达对角终点。棋子越靠近该角，战略价值越高。为此系统预设了两个 5×5 的位置得分矩阵。蓝方矩阵如下结构所示：

```
self.boardvblue = [[16, 10, 5, 2.5, 1],[10, 8, 4, 2, 1],[5, 4, 4, 2, 1],[2.5, 2, 2, 2, 1],[1, 1, 1, 1, 0]]
```

红方矩阵则为其中心对称版本。棋盘状态传入时根据当前玩家颜色自动选择矩阵，从而保障得分合理性。例如红方越靠近 (4,4)，分数越高；蓝方越靠近 (0,0)，分数越高。

4 爱因斯坦棋决策函数初、中高级算法

案例

4.1 初级极大极小搜索算法

极大极小搜索算法 (Minimax) 作为决策逻辑基本思想是“最大化己方得分、最小化对手得分”，在有限深度内对博弈状态进行模拟并评分。以下从五个阶段详细解释算法的运行逻辑：

(1) 状态递归：从当前局面出发，模拟可能局势

策略首先读取当前棋盘状态，并模拟所有合法走法。对于每一个走法，系统会：在虚拟棋盘上执行该走法（不影响真实棋盘）；进入下一轮搜索（递归调用），假设敌方随后也会做出合理响应；这一过程持续展开，直到到达设定的搜索深度 d 或游戏自然结束为止。例如，如果当前设定搜索深度为 3，策略会构建一棵最多包含 3 层的博弈树 (Game Tree)，每个节点代表一个棋盘状态。

(2) 玩家交替：每层视为当前玩家或对手决策

在极大极小算法中，层数的奇偶性决定了谁在思考：偶数层（第0、2层）为当前玩家自己，称为“极大方”，目标是选择使得评估分值大的走法；奇数层（第1、3层）为对手，称为“极小方”，系统假设对手会尽力让当前玩家得分最小。因此，策略在搜索过程中始终模拟一种“轮流下棋”的过程，不断切换极大极小角色。

(3) 叶节点评估：调用静态函数对最终局面评分

当搜索递归进入以下任一情况时，将停止扩展，并使用静态评估函数 `FenxiPinggu.get_value()` 对该状态进行打分：已达到最大深度限制 d ；当前局面已分出胜负（调用 `checkWin()`）；此时，系统会计算该局面当前玩家的得分。

(4) 得分回传：极大选择最大，极小选择最小

完成对每个叶子局面的评分后，系统将结果“回传”至上一层。若当前为“极大方”（即当前玩家），选择其子节点中的最大值；若为“极小方”（即假想对手），选择其子节点中的最小值；如此一层层向上回溯，直到回到搜索树的根节点，最终选择得分最优的那条路径。

(5) 走法选取：选择评分最高的首步动作

在搜索树构造完毕后，根节点的每个分支代表当前回合的一个候选走法，系统会选择其中静态评分值最高的那一个作为实际落子：即：当前玩家在“假设敌我都完美应对”的前提下，做出的最有利抉择。这一步称为策略的“决策执行”，可通过 `board.do_move(best_move)` 完成。

(6) 示例图

假设当前为我方（极大方 Max）思考轮次，骰子为 3，生成了 3 个可能走法：M1, M2, M3。每个走法执行后，对手（极小方 Min）将有 3 个对应响应：m1, m2, m3（即假设对手也思考走法）。

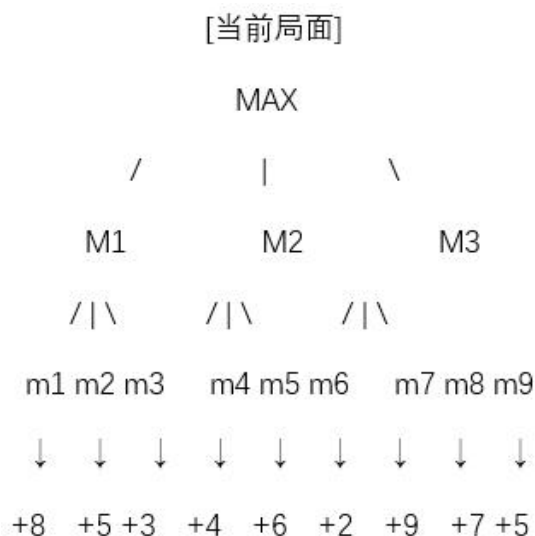


图4 极大极小搜索过程示意图

最底层为叶子节点，静态评估函数打出评分分值；极小方选择每组子节点的最小值：

$$M1 \rightarrow \min(+8, +5, +3) = +3$$

$$M2 \rightarrow \min(+4, +6, +2) = +2$$

$$M3 \rightarrow \min(+9, +7, +5) = +5$$

极大方选择最大值：

$$\max(+3, +2, +5) = +5$$

所以最终决策为选择 M3 这一步。

4.2 中级蒙特卡洛搜索算法

依赖于人工设定的搜索深度，难以处理动态性强、博弈过程长的局面。因此，中级案例引入一种更具“模拟博弈”能力的策略——基于蒙特卡洛对局模拟的走法选择方法。该方法不建立搜索树，也不进行深度剪枝，而是通过在每个走法下完整模拟多次对局，统计其胜率，从而选择最优动作。策略本质是“通过实战表现来评估当前每个走法的优劣”，在不增加搜索结构复杂度的前提下，增强了决策的现实性和鲁棒性。该方法逻辑上可以划分为以下五步：

(1) 获取当前合法走法

系统首先调用 `board.get_available_moves()` 获取当前骰子与轮次下所有可选走法，每一个走法将作为一个评估候选。

(2) 模拟完整对局

对于每个走法，系统将使用其在虚拟棋盘上执行该步后形成的新局面为起点，进行 N 次完整对局模拟。模拟过程中，双方玩家均使用 `StaticEvaluationMove` 策略（即静态评估函数 `FenxiPinggu` 驱动）来进行自动博弈，直至游戏结束。

(3) 胜率统计

每次模拟结束后，若最终获胜方为当前玩家，则该走法的胜场数 `winCount` 加一。最终，该走法的估计胜率为：

$$\text{Winrate}(\text{move}) = \frac{\text{WinCount}}{N} \quad (4)$$

决策执行阶段将根据蒙特卡洛搜索将所有走法模拟完毕后，系统将选取胜率最高的那个作为最终落子策略。

(4) 多线程加速

考虑到每个走法下都要模拟 N 次完整对局，若走法数量多，模拟量大，为提高效率，可对所有走法并行处理以提升效率。

4.3 高级深度学习算法

随着人工智能博弈研究的不断深入，单纯依赖搜索与静态评估函数的策略逐渐暴露出局限性，难以精准应对博弈中高维复杂状态。爱恩斯坦棋中受骰子影响存在明显的非确定性，使得传统极大极小搜索在高层搜索空间中容易遭遇维度灾难。因此，本案例在借鉴 AlphaGo Zero 框架的基础上，设计并实现了结合深度神经网络算法进一步提升对抗性与策略鲁棒性。

(1) 神经网络策略结构设计

本案例中的神经网络分为两部分：策略网络（Policy Head）与价值网络（Value Head），其结构沿用 AlphaGo Zero 中“卷积层 + 残差块 + 策略头/价值头”的设计理念，并结合爱恩斯坦棋 5×5 棋盘与棋子数字特征，构造适应性输入格式。数据结构如下：

输入张量维度： $N \times 4 \times 5 \times 5$ ，其中 N 为 batch size，4 表示通道（棋局、上一步、可动棋子、轮次）， 5×5 为棋盘大小；

卷积块结构：采用 3×3 卷积核，ReLU 激活，padding=1；

残差块结构：两个卷积层后残差连接；

策略头输出：56 个可能落子位置的 Softmax 概率分布，公式如下：

$$P = \text{Softmax}(f(x; \theta)) \quad (5)$$

价值头输出：预测当前局面对当前玩家的胜率，映射范围为 $[-1, 1]$ ；

$$V = \tanh(f(x; \theta)) \quad (6)$$

(2) 神经网络在搜索算法中的作用

为了克服爱恩斯坦棋中由于骰子随机性导致的策

略不稳定问题，本案例将构建好的神经网络输出结果深度融合至搜索策略中，形成了“深度学习 + 蒙特卡洛树搜索（MCTS）”的高级策略框架。该融合主要体现在策略网络（Policy Head）和价值网络（Value Head）分别指导搜索树扩展方向与模拟评估回传值，提升搜索质量与效率。

策略网络的作用为引导搜索分支。策略网络的输出为当前棋盘状态下通过深度学习算法计算出各可能移动的概率分布记作 P ，该概率向量会用于蒙特卡洛树搜索（MCTS）扩展阶段，决定新节点的优先探索顺序。具体地，在 MCTS 每次选择下一节点时，使用如下公式：

$$a^* = \arg \max (Q(s, a) + c_{puct} \cdot P \cdot \frac{\sqrt{\Sigma_b N(s, b)}}{1 + N(s, a)}) \quad (7)$$

其中： $Q(s, a)$ ：a当前的平均获胜率； c_{puct} ：控制探索-利用平衡的超参数； $N(s, b)$ ：该动作被访问的次数。较于传统 MCTS 中“平均探索”所有分支的做法，引入策略先验使得树搜索更加聚焦于高潜力动作，大幅节省模拟资源并提高策略执行质量。

价值网络的作用是替代模拟回传值。在传统 MCTS 中，遇到叶子节点后通常采用随机自博弈模拟到游戏结束，然后以胜负值（+1/-1）作为回传分值。这种做法计算代价高且误差大。为提升效率，本案例引入基于深度学习的价值网络预测当前状态下的胜率，公式为：

$$V = \tanh(f(x; \theta)) \quad (8)$$

其中输出 V 为 $[-1, 1]$ ，表示当前玩家从状态 s 开始获胜的概率估计。当搜索扩展到一个新状态时，立即调用价值网络返回该状态的胜率估计，作为当前路径的回传评估值，无需再进行 rollout。其中的深度学习网络可采用不同的基于卷积神经网络或者基于图神经网络的特征提取器。

5 课程取得成果

经过几年的探索和发展，我校计算机类专业学生的实践能力和人工智能思维和意思均得到了极大的提升，在人才培养、教学研究、专业建设和课程建设方面取得了很多有价值的成果，具体体现在以下方面。

5.1 课程的达成度情况

本文统计了课程最近两年的达成度情况表如表1。从表可以看出，本课程均完成了达成度的要求，并且达成度情况一直较高并有继续提高达成度值的趋势。

5.2 学生培养成果

本课程组经过这一轮课程改革以来，我校计算机类学生在大二结束暑假参加全国大学生计算机博弈大赛和计算机博弈锦标赛，获得了全国大赛一等23项，

表1: 课程近两年的达成度情况

评价目标	年份	评价成绩	达成度值	目标达成	权重	达成情况
目标1:比赛成绩	2023年	79.24	0.65	0.96	40%	达成
	2024年	83.17	0.65	0.83		达成
目标2:内容答辩	2023年	89.5	0.65	0.97	30%	达成
	2024年	85.95	0.65	0.86		达成
目标3:报告编写	2023年	75.95	0.65	0.98	20%	达成
	2024年	85.63	0.65	0.86		达成
目标4:课堂表现。	2023年	93.28	0.65	0.90	10%	达成
	2024年	97.82	0.65	0.98		达成

二等奖40余项，三等奖20多项的优异比赛成绩；同时同学们在经过大二的博弈训练后，在程序设计能力和算法学习能力方面也得到了很好提升，为大三参加计算机设计大赛的人工智能和大数据组比赛、蓝桥杯算法比赛、计算机系统能力大赛积累了丰富的比赛经验，在最近两年的大三学生参加比赛中获得蓝桥杯国赛一等奖10项、二等奖30余项，计算机设计大赛、计算机系统能力大赛、ICPC比赛、学生考研、保研以及就业等方面等都取得了很好的成绩，本科生还基于计算机博弈内容发表中文核心期刊论文2篇[13,14]。对于对于不太擅长学科竞赛的学生而言，通过在题目中分配基础开发类任务来提高其编程能力，为后续课程提高专业实践能力也起到很好的促进作用。

5.3 教学内容更新

课程组编写了《基于机器博弈的计算机问题求解课程设计》题目手册，该手册经过3次迭代升级，目前形成完整的基于计算机博弈为内容的题目手册，为“计算机问题求解课程设计”的实施提供了比较完整的解决方案。

5.4 教学研究成果

基于计算机博弈的依托学科竞赛的学生能力培养举措，近两年取得了众多的教学成果，2021年获得成都理工大学大学教学成果特等奖一项、二等奖2项，同时“基于计算机博弈的计算机问题求解能力培养模式探索”获批成都理工大学大学教学质量工程重点项目，本课程组教师近三年同时还获批教育部协同育人项目4项。

5.5 专业课程建设

计算机科学与技术专业在2021年获批四川省一流专业，《数字媒体技术专业实训》课程获批2021年四川省省级一流社会实践课程，这些教学研究成果和教学研究表明本次教改项目研究的结果得到比较广泛的认可，为下一步继续打磨该课程和改革模式具有重要

支撑作用。

6 结束语

针对传统计算机类专业学生的计算机问题求解实践能力培养方法存在的过程考核不足的缺点，本文介绍了我们课程组在该问题上的有益探索，设计了基于计算机博弈项目的计算机问题求解能力培养新模式。该模式不仅在内容上具有高阶性，在模式上也具有提出了具有创新型的考核方法，促使学生和教师必须不断学习新知识，才能满足不断出现的新挑战。课程组的下一步工作是继续完善细化该模式在教学内容、资源整合和教学成果方面打磨，力争打造具有较强示范性的计算机问题求解能力培养通用模式。

参考文献

[1] 王亚杰, 王晓岩, 邱虹坤等. 建设棋牌谱标准, 构建计算机博弈竞赛持续发展新生态[j], 实验技术与管理. 2020, 37 (02), 19-23.

[2] 李霞丽, 吴立成, 李永集. 基于棋型的藏族“久”棋计算机博弈研究[J], 智能系统学报. 2018, 13 (04), 577-583.

[3] 罗俊仁, 张万鹏, 苏炯铭等, 计算机博弈中序贯不完美信息博弈求解研究进展[J], 控制与决策, 网络首发, DOI:10.13195/j. kzyjc. 2022. 0698.

[4] 梅险, 侯利元, 谢文等. 计算机博弈创新教育的研究与实践[J], 黑龙江教育(理论与实践). 2017(12), 45-46.

[5] 李媛, 王静文, 王博. 以计算机博弈课程为基础的专创融合教学改革与实践[J], 高教学刊. 2022, 8(14), 130-132+137.

[6] 李淑琴, 丁濛, 孟坤. 计算机博弈教学中的计算思维能力培养[J], 计算机教育. 2017(05), 22-24.

[7] 王亚杰, 邱虹坤, 尹航等. 计算机博弈竞赛与创新人才培养模式改革[J], 实验技术与管理. 2016, 33(10), 10-14.

[8] 王亚杰, 邱虹坤, 吴燕燕等, 计算机博弈的研究与发展[J], 智能系统学报. 2016, 11 (06), 788-798.

[9] 柳丽召, 蔡彪, 吴媛媛等, 以成果为导向的课外实践教学体系构建与实践[J], 计算机教育. 2023(06), 136-140.

[10] 吴静, 蔡彪, 胡艳梅等, 智慧教学环境下的立体课堂互动模型构建与实践探索[J], 计算机教育. 2022(08), 108-112+117.

[11] 蒋宗礼. 计算思维之我见[J], 中国大学教学. 2013(09), 5-10.

[12] 王志强, 刘芳芳, 基于计算思维的计算机基础课程改革研究[J], 中国大学教学. 2013(06), 59-60+36.

[13] 王志明, 胡洋成, 蔡彪等. 爱恩斯坦棋博弈的图神经网络算法研究[J/OL]. 重庆理工大学学报(自然科学), 1-10[2025-06-10]. <http://kns.cnki.net/kcms/detail/50.1205.T.20250416.1539.005.html>.

[14] 蔡彪, 徐昕怡, 谢婷等, 改进深度神经网络在爱恩斯坦棋中的应用研究[J], 重庆理工大学学报(自然科学), 2024年第38卷第5期, 108-114.

[15] 张龙, 夏冰, 邢颖等, 价值引领网下网络空间安全专业课程思政的实践路径研究[J], 《计算机技术与教育学报》, 2025年07月第13卷第2期, P24-28.