

北太天元

北太天元论文集

北太天元科学计算开发者大赛论文集

目录

赛题 8: 基于北太天元的数学建模案例开发.....	3
1. 赛题描述.....	3
1.1 赛题背景与目标.....	3
1.2 题目设置.....	3
1.3 技术要求.....	3
1.4 提交材料.....	4
2. 赛题解析.....	4
2023 年全国大学生数学建模竞赛 C 题《蔬菜类商品的自动定价与补货决策》赛题解析.....	4
2024 年全国大学生数学建模竞赛 C 题《农作物的种植策略》赛题解析	33
2020 年全国大学生数学建模竞赛 C 题《中小微企业的信贷决策》赛题解析	19
2024 年全国大学生数学建模竞赛 A 题《“板凳龙”闹元宵》赛题解析	4
2023 年全国大学生数学建模竞赛 B 题《多波束测线问题》赛题解析	15
2025 年亚太地区大学生数学建模竞赛中文赛题 A 题《农业灌溉系统优化》赛题解析.....	10
2022 年认证杯国际数学建模竞赛 C 题《人类活动分类》赛题解析	24
3 优秀论文.....	38
202503080316 应用北太天元的生鲜商超补货策略优化模型	38
202503080008 基于北太天元软件的农作物种植问题的求解	389
202503080096 基于 QMC-DEGA 与 CVaR 风险控制的农作物种植策略优化	426
基于交易票据的中小微企业信贷风险量化与策略优化研究	201
202503080091 多波束测深测线优化研究.....	177
202503080123 基于几何模型的《板凳龙》运动状态与路线研究	38
202503080104 基于多元模型与智能算法的灌溉系统协同优化及应急决策研究	73
202503080045 针对相似人类活动识别的 XR-KS 多层分类器模型	234

赛题 8：基于北太天元的数学建模案例开发

1. 赛题描述

1.1 赛题背景与目标

数学建模是运用数学方法解决实际问题的的重要手段，而数学软件工具的使用是数学建模的关键环节。北太天元作为我国自主研发的科学计算软件，在数学建模、科学计算等领域具有重要应用价值。本题目旨在推广北太天元在数学建模中的应用，检验其在实际问题求解中的能力和表现，同时为北太天元的功能完善提供用户反馈。

1.2 题目设置

从历年全国大学生数学建模竞赛（2020-2024 年）的 A、B、C 题中自主选择一道题目，使用北太天元作为主要工具完成该题目的全部问题求解，并撰写完整的数学建模论文。

1.3 技术要求

北太天元使用要求

- (1) **使用比例要求：**北太天元的使用比例不低于 80%
- (2) **功能覆盖：**尽可能使用北太天元的多种功能模块
 - a. 数值计算和矩阵运算
 - b. 数据可视化功能
 - c. 优化工具箱
 - d. 统计与机器学习工具
 - e. 其他专业工具箱
- (3) **替代方案：**对于北太天元暂不支持的功能，可使用其他工具替代，但需：
 - a. 明确说明使用的替代工具及原因
 - b. 详细记录北太天元的功能缺失情况
 - c. 提供改进建议

代码规范要求

代码结构清晰，功能模块化

关键代码段提供详细注释

变量命名规范，增强可读性

提供完整的运行示例和测试数据

1.4 提交材料

(1) 数学建模论文（30 页以内），建议按如下进行组织：

- a. **摘要**：问题概述、建模方法、主要结果
- b. **问题分析**：问题背景、关键因素识别、假设条件
- c. **模型建立**：数学模型构建、变量定义、参数说明
- d. **模型求解**：算法选择、求解过程、北太天元使用详情
- e. **结果分析**：计算结果、敏感性分析、实际意义
- f. **模型评价**：优缺点分析、改进方向、适用范围

(2) 完整代码包（zip 格式），建议组织结构：

- a. **data 文件夹**：存放所有数据相关文件；
- b. **src 文件夹**：存放所有源代码文件；
- c. **results 文件夹**：存放所有输出结果；
- d. **main.m 文件**：主程序入口，整合所有模块，提供一键运行的完整解决方案。

(3) 北太天元使用体验报告（5-10 页），建议包含如下内容：

- a. **使用总结**：北太天元功能使用情况统计
- b. **优势分析**：在解决特定问题中的突出表现
- c. **问题反馈**：遇到的 bug、功能缺失、性能问题
- d. **对比分析**：与 MATLAB、Python 等工具的对比（如有使用）
- e. **改进建议**：功能增强、界面优化、文档完善等建议
- f. **应用前景**：在数学建模和科研中的应用潜力

(4) PPT 与讲解视频

2. 赛题解析

2024 年全国大学生数学建模竞赛 A 题《“板凳龙”闹元宵》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面传统文化活动中一类典型且极具物理美学的运动规划问题——长序列刚性连接体在平面约束路径下的运动学建模与优化。板凳龙作为浙

闽地区传统民俗活动，其盘入、调头、盘出的全过程涉及大量几何约束、碰撞规避、路径优化与速度控制问题，具有鲜明的应用数学与计算力学交叉特征。

命题组期望通过本题目，促使学生在真实物理与几何约束下，完成从“给定路径下的运动学正解”到“碰撞临界状态判定”，再到“路径几何参数优化”以及“速度传递与极限能力分析”的完整科学探究链条。其深层立意有三：其一，强调“几何直观与解析表达”的结合，要求学生将板凳龙抽象为多刚体链式系统，利用平面曲线参数方程精确刻画每一时刻每一节点的空间位形；其二，突出“约束驱动求解”的工程思维，即碰撞避免是硬约束，所有决策都必须在保证安全的前提下追求效率或可行性；其三，呼应国产数值计算工具在复杂运动规划问题中的适用性验证，鼓励学生借助北太天元实现大规模时序数据生成、非线性方程组求解、单目标连续优化与迭代搜索算法，从而在虚拟环境中推演真实舞龙运动的极限性能。

整体而言，本题以传统文化为载体，以运动学为骨架，以优化为灵魂，检验学生对连续系统离散化、几何约束建模、临界状态搜索和速度传递机制的深刻理解与数值实现能力。

二、考察学生的知识点与能力

本部分按问题一至问题五分开表述，全题可视为基于阿基米德螺线的多刚体链运动学与碰撞约束优化问题，在不同问次中依次递进地考察正运动学计算、临界状态检测、路径参数优化、调头曲线设计与速度极限求解。

问题一 等距螺线盘入运动的全链位形与速度时程生成

总体考察方向

本问旨在考察学生建立板凳龙在给定阿基米德螺线路径上的运动学正解模型的能力。核心任务是将连续的运动过程离散化为每秒的位形快照，精确计算龙头、每一节龙身以及龙尾各关键把手中心的平面位置与瞬时速度矢量。这要求学生对螺线几何有精确的参数化描述，理解链条各节之间的刚性连接约束（相邻把手间距固定等于板长扣除孔位偏移），并建立龙头速度恒定条件下的全链速度传递关系。

问题 1 与对应模型及北太天元可执行内容

阿基米德螺线路径的精确参数化生成：需根据给定螺距（55 厘米）和初始圈数位置（第 16 圈 A 点），建立整条盘入螺线的平面坐标与极角之间的精确映射。北太天元可高效生成从初始位置沿螺线向内盘入的密集点列，支持任意弧长参数下的坐标插值查询，为后续各把手定位提供基础路径数据库。

链条等距约束下的节点定位递推算法：已知龙头前把手在螺线上的位置（由行进距离确

定），其余每一节龙身前把手以及龙尾后把手均需满足与前一节点的距离恒等于该节板凳的有效连接长度（即板长减去孔到板端距离的两倍，因相邻两孔间距为两节板各自孔距之和）。该定位问题本质上是给定起始点和目标曲线，沿曲线搜索满足固定弦长的下一节点。北太天元可基于路径弧长参数，采用逐步搜索或弦截迭代方法，逐节定位所有把手在螺线上的坐标，并确保相邻节点沿螺线的前后顺序不颠倒。

速度矢量的传递计算：龙头前把手速度大小恒定、方向沿螺线切线方向；后续各节点的速度由其位置随时间的变化率决定，可通过相邻时刻的位移差分或基于链式约束的解析传递获得。北太天元可利用差分法批量计算每一秒各节点的速度矢量分量，并输出标量速率，同时支持结果的结构化存储，直接生成符合模板格式的 Excel 文件。

关键时点数据抽取与表格生成：需从全时程 300 秒结果中抽取 0、60、120、180、240、300 秒六个时刻的特定节点位形与速度，按论文格式排版。北太天元支持灵活的索引切片和数据透视功能，可快速提取指定时刻、指定节点的数据子集，并自动格式化输出。

小结：本问重点考察平面曲线参数化建模、约束链的逐节点几何定位、时域差分速度计算以及大规模时序数据的批处理与输出能力。北太天元在路径插值、迭代搜索、差分计算和结果结构化存储方面提供完整的数值环境，确保 300 秒、每秒 223 节板凳的位形数据可精确、高效生成。

问题二 碰撞约束下的盘入终止时刻判定

总体考察方向

本问将问题一的纯运动学计算升级为带碰撞安全约束的临界状态搜索问题。要求确定板凳龙沿螺线继续盘入至哪一时刻，链条内部将首次发生相邻或非相邻板凳之间的几何干涉（即碰撞），该时刻即为盘入终止时刻。核心难点在于碰撞判据的量化——不仅要考虑相邻板凳通过铰接自然不会碰撞，更要关注非相邻板凳在螺线紧密盘绕时因曲率增大和间距缩小而发生的空间侵占。

问题 2 与对应模型及北太天元可执行内容

碰撞检测模型的构建：需对任意两节非相邻板凳，根据其四个把手（或板体矩形区域）的瞬时位置，判断是否存在区域重叠或间距小于安全裕度。由于板凳具有有限宽度（30 厘米）和板长，需将每一节板凳抽象为有向线段或矩形刚体，并定义两刚体之间的最小距离函数。北太天元可编程实现批量化的两两碰撞检测函数，利用其矩阵运算能力，对每一时刻的 223 节板凳进行高效成对检测，避免四重循环导致的计算爆炸。

基于步长迭代法的终止时刻搜索：由于碰撞发生是一个从无到有的连续过程，可直接沿问题一生成的时间序列逐步检测，首次出现碰撞的时刻即为终止时刻；亦可采用更精细的步长迭代法，在临界附近加密时间步长以逼近真实碰撞瞬间。北太天元支持灵活的循环控制与条件中断逻辑，可设定检测步长自适应缩小，在保证精度的同时控制计算量，最终输出终止时刻及该时刻的全链位形与速度数据。

临界时刻的数据抽取与输出：需将终止时刻下指定的关键节点位置和速度按模板格式存入 result2.xlsx。北太天元支持动态时间索引提取，自动完成数据组装与文件写入。

小结：本问重点考察刚体碰撞检测的几何建模、大规模成对检测的计算效率优化，以及临界时刻的精确搜索策略（步长迭代法）。北太天元凭借其矩阵化并行计算能力和灵活的迭代控制，使学生在合理时间内完成数千时间步、数万对检测的计算任务，准确锁定盘入极限。

问题三 调头空间约束下的最小螺距优化

总体考察方向

本问跳出固定参数的路径跟踪，转向路径本身的几何设计优化。要求在调头空间（以螺线中心为圆心、直径 9 米的圆形禁入区）约束下，确定阿基米德螺线的最小螺距，使得龙头前把手恰好能沿螺线盘入至该圆形区域边界。该问题是典型的单目标连续优化问题，优化变量为螺距，目标函数为最小化螺距，约束为龙头轨迹与圆形区域边界相切或至少不侵入内部。

问题 3 与对应模型及北太天元可执行内容

路径-障碍边界接触条件的数学表达：需建立螺线最内圈（即龙头所能到达的最深位置）与圆形调头区域边界之间的几何关系，确定螺距变化如何影响龙头可达到的最小极径。北太天元可构造该接触条件的数值计算函数，输入螺距值，输出龙头沿螺线行进至极限位置时的极径与圆心距离的偏差量，作为约束函数。

单目标连续优化模型的求解：以最小螺距为优化目标，以龙头轨迹不进入圆形禁入区为约束，采用一维搜索或连续优化算法（如黄金分割法、二次插值法或更通用的序列二次规划）进行求解。北太天元内置优化工具箱可支持约束非线性优化问题的求解，亦支持用户自行编写步长搜索或区间收缩算法，快速收敛至满足边界相切条件的最小螺距。

结果验证与敏感度分析：求得最优螺距后，需回代验证龙头在最内圈处与圆形边界恰好相切，并可视化轨迹与调头空间的位置关系。北太天元可绘制螺线路径与圆形区域的叠加图形，直观确认约束满足情况，同时可进行螺距微扰下的偏差分析，评估解的稳定性。

小结：本问重点考察单目标连续优化建模、非线性约束处理以及一维/多维搜索算法的应

用能力。北太天元作为优化计算平台，提供从目标与约束函数定义、优化求解器调用到结果可视化验证的完整 workflow，培养学生面对几何设计问题时的参数优化思维。

问题四 S 形调头曲线的几何设计与全链运动学时程生成

总体考察方向

本问将问题从单一的盘入螺线扩展为“盘入螺线—调头 S 形曲线—盘出螺线”三段拼接路径，其中调头曲线由两段相切圆弧组成，前段半径为后段的两倍，且整体关于螺线中心呈中心对称的盘出路径要求调头曲线两端分别与盘入和盘出螺线相切。核心任务分为两层：其一，在给定盘入螺距（1.7 米）和圆形调头空间约束下，确定该特定构型 S 形曲线是否可调（即改变两段圆弧的半径比例或位置），在保持相切约束的前提下使调头曲线更短；其二，完成全路径拼接后，生成从 -100 秒到 100 秒（以调头开始为零时刻）的全链位形与速度时程。

问题 4 与对应模型及北太天元可执行内容

S 形调头曲线的几何构型分析与优化：需解析两段圆弧相切且分别与入、出螺线相切的全部几何约束方程，确定曲线族的自由度数。在给定约束下，探究是否能在保持相切的前提下调整两段圆弧半径之比（原为 2 倍关系）或圆心位置以缩短总弧长。北太天元可建立该几何系统的数值求解框架，利用非线性方程组求解器求解不同半径比下的可行构型，并计算对应的总曲线长度，从而通过参数扫描或连续优化找到最短调头路径对应的几何参数。

三段路径的拼接与弧长参数化统一：需将盘入螺线、调头 S 曲线、盘出螺线按顺序拼接为一条连续且一阶连续（切线连续）的完整路径，并建立全局弧长参数与各段局部参数的映射。北太天元支持分段函数的统一编码和弧长累积计算，可生成任意弧长对应的平面坐标与切线方向，为后续运动学求解提供统一路径查询接口。

全链运动学时程生成（-100 秒至 100 秒）：以调头开始时刻为零时刻，龙头前把手以 1 米/秒匀速沿拼接路径行进，采用与问题一相同的等距约束递推定位方法，逐秒计算全部把手的位置与速度，涵盖盘入后半段、完整调头段和盘出前半段。北太天元可复用问题一的定位迭代算法，仅需将路径替换为拼接路径，即可批量生成 201 秒内每秒的完整位形数据，并按模板抽取 -100、-50、0、50、100 秒五个时刻指定节点的数据。

小结：本问重点考察复杂平面曲线的几何构型求解、非线性方程组数值求解、分段路径拼接与弧长参数化、以及扩展路径下的全链运动学计算。北太天元在非线方程组求解、参数扫描优化和分段路径统一查询方面的能力，使学生能够完成调头曲线的几何优化与全时程运动仿真的双重任务。

问题五 速度传递约束下的龙头最大行进速度确定

总体考察方向：本问关注速度传递放大效应这一关键运动学现象。在链条刚性连接且沿弯曲路径行进时，各节点的速度大小并非均匀一致——曲率变化和链节间的相对转角会导致某些节点的瞬时速率显著高于龙头速度。本问要求在龙头行进速度可变的前提下，在问题四给定的完整路径上，确定龙头速度的上限，使得全部 223 节板凳上所有把手的速率均不超过 2 米/秒的安全阈值。

问题 5 与对应模型及北太天元可执行内容

速度传递比的计算与路径敏感性分析：需沿整条拼接路径逐点计算龙头单位速度下各节点的瞬时速率，得到每个节点的速度放大系数，该系数是路径曲率和链条姿态的函数。北太天元可基于问题四生成的位形序列，通过差分法计算各节点在不同路径位置的速度传递比，并统计整条路径上所有节点中的最大放大系数。

最大安全速度的标定：龙头最大行进速度等于速度上限（2 米/秒）除以整条路径上出现的最大的速度传递比。该计算简洁直接，但要求速度传递比的计算精度足够高，且需覆盖所有节点和所有时间步。北太天元可批量完成全路径、全节点速度传递比的扫描，自动定位危险节点（即速率最大的节点）及其出现的路径位置和时间，为最终结论提供明确依据。

结果验证与安全裕度建议：在求得最大速度后，需回代验证在该速度下确实没有任何节点超过 2 米/秒，并可视乎情况给出留有一定裕度的建议速度。北太天元支持参数扫描和批量验证计算，可快速评估不同龙头速度下的最大节点速率曲线，供决策参考。

小结：本问重点考察运动链速度传递的数值计算、全路径极值搜索以及安全阈值的逆向标定能力。北太天元在差分计算、极值扫描和参数验证方面的高效性，使学生能够准确锁定速度瓶颈环节，并给出既安全又尽量满足观赏性需求的行进速度上限。

综合评述：全题从固定路径的正运动学计算，到碰撞临界搜索，到路径几何参数优化，到 S 形拼接路径设计，再到速度极限标定，形成了一条完整的“建模—计算—优化—验证”科研链条。学生在解决过程中，须系统掌握平面曲线参数化、多刚体链约束定位、碰撞检测算法、单目标连续优化、非线性方程组求解、分段路径拼接、速度传递分析等多元方法，并具备将物理直觉转化为可计算模型的抽象能力。北太天元在这一体系中，不仅作为数值计算引擎，更作为一个集路径生成、约束求解、优化搜索、时程仿真和可视化验证于一体的综合性科研平台，为学生在有限竞赛时间内完成高精度、大规模、多阶段的复杂运动规划任务提供坚实的技术保障。

2025 年亚太地区大学生数学建模竞赛中文赛题 A 题《农业灌溉系统优化》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面智慧农业中一项具有高度现实紧迫性的资源优化问题——在多种土壤类型、动态气象条件、有限水源和差异化作物需水规律共同作用下，如何科学设计灌溉基础设施（引水管网与储水罐）并制定动态灌溉调度策略，以实现水资源高效利用与作物安全保障的双重目标。农业灌溉用水占全球淡水取用量的近七成，而气候变化加剧了降水时空分布的不确定性，使得“平时用河水、旱时用储水”的传统经验亟需量化、模型化的决策支持。

命题组期望通过本题目，促使学生在真实农场空间约束下，完成从“气象-土壤湿度关系建模”到“静态灌溉基础设施优化”，再到“应急水源配置与风险分析”，最终到“动态月际灌溉方案规划与系统适应性校核”的完整智慧灌溉决策链条。其深层立意有三：其一，强调“数据驱动的环境感知”能力，要求学生从历史气象与土壤湿度时序数据中挖掘可预测的响应关系，为灌溉决策提供前瞻性依据；其二，突出“基础设施投资与运营调度”的两阶段优化思维——先确定管道与储罐的最优空间布局（一次性投入），再在给定系统下制定逐日/逐月的用水调度方案（运营决策），且两者相互制约；其三，呼应“极端气候韧性农业”的国家需求，引导学生将应急储备水源比例与旱灾概率进行量化关联，使优化方案不仅经济最优，更具备应对不利情景的鲁棒性。

整体而言，本题以智能灌溉为载体，以预测-规划-调度为主线，以优化为方法论核心，检验学生融合时间序列分析、空间几何布局、非线性成本建模、随机风险量化与动态调度规划的综合研究能力。

二、考察学生的知识点与能力

本部分按问题一至问题四分述，每一问题内部采用“总一分一总”结构，先点明该问整体考察方向，再按子任务拆解对应的模型类别与北太天元可支撑的计算环节，最后小结该问的综合能力目标。全题内核可视为数据驱动预测与基础设施-调度联合优化的复杂系统决策问题，在不同问次中依次递进地考察预测建模、静态布局优化、风险量化与动态调度规划。

问题一 气象要素与土壤湿度的回归预测建模

总体考察方向

本问旨在考察学生基于历史气象观测数据，建立土壤湿度（5 厘米深度）与多种气象因子（如温度、气压、风向风速、降水量等）之间的统计响应模型。核心任务是从附件提供的

气象时序数据中识别关键影响因子，构建具有良好预测精度的回归模型，并利用该模型对表 2 中给定的小时气象数据进行土壤湿度预测。该问强调的是数据预处理、特征筛选、模型选择与验证的完整数据科学流程。

问题 1 对应模型及北太天元可执行内容

数据清洗与特征工程：需对附件中的气象时序数据进行缺失值处理、异常值剔除和单位统一；需从原始字段中构造可能的派生特征，如累积降水量、风速的平稳性指标、温湿交互项等。北太天元支持表格数据的读取、清洗和变换操作，可快速完成数据预处理，并生成扩展特征矩阵供后续建模使用。

相关性分析与关键因子筛选：需计算各气象特征与土壤湿度之间的相关系数，结合领域知识初步筛选对土壤湿度影响显著的预测变量，避免多重共线性干扰。北太天元可批量计算相关矩阵并可视化热力图，辅助判断变量间的关联强度和方向，为回归模型的特征选择提供统计依据。

多元线性回归基准模型的建立：以筛选后的气象因子为自变量、土壤湿度为因变量，构建多元线性回归模型，估计各因子的边际效应。北太天元可调用内置回归函数完成参数估计、显著性检验、残差诊断和拟合优度评估，快速建立可解释的基准预测模型。

非线性回归模型的拓展与比较：当线性模型拟合不足时，需引入非线性项（如二次项、交互项、对数变换等）或采用非线性回归框架，以捕捉气象因子与土壤湿度之间的复杂响应关系（如降水对湿度的饱和效应、温度与湿度的交互作用）。北太天元支持用户自定义非线性模型结构，并可利用其优化工具箱进行参数估计，同时可与线性模型进行交叉验证比较，选择预测精度最优的模型。

逐时预测与结果输出：将训练好的最优模型应用于表 2 中给定的小时气象数据，生成当天各时刻的土壤湿度预测值，并按规定格式填入表格。北太天元支持批量预测计算和结果结构化输出，确保预测结果精确可复现。

小结：本问重点考察时序数据清洗与特征工程、相关分析、多元线性回归与非线性回归建模、模型比较与验证、以及批量预测应用的全流程数据科学能力。北太天元提供从数据读入、变换、建模到预测输出的完整工具链，使学生在有限时间内完成从原始数据到可用预测模型的高效迭代。

问题二 固定时段下的灌溉基础设施优化布局（管道与储水罐）

总体考察方向

本问将焦点从预测转向决策，要求在不考虑作物不同生长时期差异化需水量的前提下，仅以土壤湿度不低于存活阈值（0.22）为约束，利用 2021 年 7 月整月数据，合理规划引水管道布线（含河流取水点至各灌溉区域）与储水罐（容积与位置），在满足作物存活的前提下使总建设花费最小。该问本质上是带连续空间决策和离散设施选址的混合优化问题，成本函数具有非线性特性（管道成本与长度和流量的非线性关系，储罐成本与容积线性相关）。

问题 2 对应模型及北太天元可执行内容

空间离散化与候选设施点生成：需将 1 公顷正方形农场按喷头覆盖半径（15 米）和相邻喷头最小间距约束（15 米）进行网格化离散，确定潜在的喷头安装位置与储水罐候选位置。北太天元可生成规则网格点阵，并根据喷头覆盖半径筛选有效覆盖全农场的最小喷头集合，同时标记河流边界上的可取水点位置。

逐时土壤湿度动态模拟与灌溉需求计算：利用附件中 2021 年 7 月的逐时气象数据和问题一建立的土壤湿度预测模型，模拟在不灌溉条件下土壤湿度的自然演变过程；当湿度低于阈值时，触发灌溉需求，计算每次所需灌溉水量以恢复至合理水平。北太天元可基于时间循环执行逐时湿度更新与缺水量累计统计，生成全月各喷头覆盖子区域的总灌溉需求量和峰值日流量，为管道通流量设计提供输入。

非线性成本约束下的管道网络布局优化：需在河流取水点、储水罐位置和喷头之间设计管道连接关系，使总管道成本最小化。管道成本为长度和流量的非线性增函数，需在满足各喷头流量需求和储罐补给流量的前提下，选择最优的管道拓扑和管径（反映在流量上限中）。北太天元可利用其优化工具箱，结合非线性规划方法（如序列二次规划法）求解该带非线性成本函数的连续-离散混合优化问题，或采用贪心算法进行启发式近似求解，在可接受时间内获得高质量工程方案。

储水罐容积与位置的联合优化：储水罐可覆盖半径 15 米的圆形区域，成本按容积线性增加。需在农场内选择最优的储水罐位置（一个或多个），并确定其容积，以平衡管道建设费用与储罐建设费用。北太天元可对候选位置进行枚举或利用连续优化方法，将储罐容积与位置作为决策变量纳入总体成本函数，联合求解最优配置。

建设成本汇总与布线规划图生成：优化完成后需输出最小总建设花费，并绘制包含取水点、管道路径、储水罐位置、喷头分布和覆盖范围的规划图。北太天元支持二维图形绘制和

标注功能，可输出清晰的基础设施布局可视化结果。

小结:本问重点考察空间离散化、逐时土壤湿度动态模拟、非线性规划（含序列二次规划法）、贪心启发式算法以及多设施联合优化的综合能力。北太天元在非线性优化求解、时序循环模拟和空间可视化方面的强大支持，使学生能够将复杂的工程布局问题转化为可计算的优化模型并获得可落地方案。

问题三 应急储备水源配置与旱灾风险分析

总体考察方向

本问在问题二优化得到的灌溉系统基础上，引入河流水源不稳定的现实风险，要求完成两项任务：其一，评估当河流供水能力下降至原设计流量的 80%时，现有系统能保证多少作物存活和正常生长；其二，分析储水罐应急储备水源比例与旱灾发生概率之间的关系，给出不同旱灾概率下建议的应急储备比例，并判断该比例能否保证作物全部存活。该问考察的是在给定系统下进行情景模拟与风险量化，以及在风险-成本之间进行权衡决策的能力。

问题 3 对应模型及北太天元可执行内容

供水衰减情景下的作物存活与生长评估：将河流日供水量上限设为问题二设计流量的 80%，储水罐中预留应急水源（不可用于日常灌溉）后，重新模拟 7 月全月的灌溉过程，统计在限制供水条件下，各作物的实际灌溉量是否满足存活需水量和正常生长需水量，计算存活面积与正常生长面积。北太天元可复用问题二的逐时模拟框架，仅调整供水约束和储罐可用容量，快速生成衰减情景下的作物状态统计表。

应急储备比例与旱灾概率的关系建模：需建立旱灾概率（以某种方式定义或给定）与应预留应急水源占总储水罐容积比例之间的函数关系。这一关系可基于最坏情况分析或概率分位数方法构建——旱灾概率越高，越需预留更大比例的应急水源以应对长时间低流量，但过高的预留比例会挤压日常灌溉可用水量，可能影响正常年份的作物生长。北太天元可对不同旱灾概率情景（如 5%、10%、20%、30%等）逐一进行供水模拟，搜索在保证作物全部存活的前提下所需的最小应急储备比例，从而生成旱灾概率-储备比例的对对应表，并标注各比例下是否能保证全部存活。

储备比例的敏感性分析：为验证建议储备比例的稳健性，需分析其随气象条件（如 7 月降水多寡）、储水罐总容积和管道流量的变化趋势。北太天元可进行多情景参数扫描，输出储备比例对关键参数的敏感度曲线，为农场主提供更具可操作性的风险防范建议。

小结:本问重点考察情景模拟、风险量化、概率-策略对应关系构建以及敏感性分析能力。

北太天元在参数化模拟循环、条件判断和结果汇总方面的灵活性，使学生能够系统评估不同风险偏好下的最优储备决策，实现经济性与安全性的量化权衡。

问题四 全生育期动态灌溉方案规划与系统适应性校核

总体考察方向

本问将时间维度从单一月份扩展至整个生长季（2021 年 5 月 1 日至 7 月 31 日），并引入作物不同生长时期（播种期、开花期、成熟期）的差异化需水量，以及各作物的具体生育期时长（如成熟期均为 20 天）。要求基于全时段气象数据，规划每月每种作物的灌溉方案（含总灌溉量、河水与储水罐的水源比例），并判断前序问题中设计的灌溉系统是否能满足全生育期的灌溉需求；若不能，则需修改系统布线并重新求解。该问考察的是全周期动态调度与基础设施再优化的闭环能力。

问题 4 对应模型及北太天元可执行内容

全生育期逐日需水量计算与灌溉时机规划：根据每种作物的播种日期（5 月 1 日统一播种）、各生育期的起止时间和单位面积需水量，结合逐日气象数据驱动的土壤湿度预测，制定逐日灌溉策略——在满足最低土壤湿度（0.22）的前提下，根据作物当前所处的生育阶段，补充该阶段对应的需水量缺口。北太天元可构建全生育期逐日模拟循环，每日更新土壤湿度、作物生育阶段和累计灌溉量，自动生成各月各作物的理论灌溉需求时间序列。

现有系统能力评估：将问题二中优化得到的管道通量上限和储水罐可用容量，与全生育期内每日最大灌溉需求进行逐日比对，判断是否存在灌溉需求超过系统供水能力的日期；若存在，则记录超出量和发生时段。北太天元可对全模拟期的每日供需进行匹配度检查，自动输出能力不足的时间段和缺水量，为系统修改提供定量依据。

系统布线的再优化与调整：若现有系统无法满足全生育期需求，则需放宽问题二的约束条件（如增加储罐容积、增大管道设计流量或增设新的储罐位置），重新进行基础设施优化，但需注意新增投资应尽可能小。北太天元可复用问题二的优化框架，仅扩展约束边界并延长模拟周期，重新求解满足全生育期需求的修正方案，并输出调整后的系统布局 and 新增花费。

月灌溉方案表生成与水源比例统计：按月份（5 月、6 月、7 月）汇总每种作物的总灌溉量，并区分来自河水直供和储水罐供给的水量，计算各自比例，填写灌溉安排表。北太天元可自动分类汇总并生成符合模板格式的表格。

月灌溉情况可视化：需以图形方式展示每月每种作物的灌溉总量、水源构成（河水/储水罐）以及成熟期结束的作物标记。北太天元支持柱状堆叠图、多系列折线图或桑基图等多种

可视化形式，直观呈现灌溉策略的时间分布与水源结构。

小结:本问重点考察全生育期动态灌溉调度规划、多作物差异化需水整合、系统承载能力的时序校核、以及基于能力评估的系统再优化与可视化呈现。北太天元凭借其强大的时序模拟、条件逻辑、优化重算和图形绘制能力，使学生能够完成从长期调度策略制定到基础设施动态调整的完整闭环，体现智慧灌溉系统“规划-运营-反馈-修正”的工程实践逻辑。

综合评述:全题从气象-土壤湿度预测建模入手，到静态灌溉基础设施的优化布局，再到应急风险情景下的储备策略分析，最终到全生育期动态调度与系统适应性校核，形成了一条完整的“感知-规划-储备-调度”智慧灌溉决策链路。学生在解决过程中，须系统掌握时序数据回归建模（多元与非线性的回归模型）、非线性规划（含序列二次规划法）、启发式贪心算法、情景模拟与风险量化、动态调度规划等多元方法，并具备将农业水文知识与数学优化工具深度融合的交叉学科素养。北太天元作为贯穿全程的计算平台，从数据驱动预测、空间布局优化、风险模拟评估到动态调度可视化为学生提供了一体化的数值计算与图形呈现环境，有力支撑复杂农业系统工程问题的高效求解与方案落地。

2023 年全国大学生数学建模竞赛 B 题《多波束测线问题》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面海洋测绘领域中一项兼具理论深度与工程应用价值的核心问题——在海底地形起伏变化条件下，如何科学设计多波束测深系统的测量航线，以实现测量效率与数据质量之间的最优平衡。多波束测深技术是现代海洋测绘、海底资源勘探和航道保障的关键手段，其覆盖宽度受水深、波束开角及地形坡度等多因素耦合影响，而测线间距的设定则直接决定了相邻条带的重叠率，进而影响数据冗余度与漏测风险。

命题组期望通过本题目，促使学生在真实地理空间约束下，完成从“局部几何关系建模”到“空间变参数下的覆盖计算”，再到“大面积海域的测线网络优化”以及“基于历史单波束数据的智能布线”这一完整技术链条。其深层立意有三：其一，强调“几何直观与空间解析”的工程数学能力，要求学生将三维海底地形与波束传播路径抽象为可计算的平面与斜面几何关系；其二，突出“效率-质量”多目标权衡的系统工程思维，即在保证全覆盖的前提下，既要控制重叠率不过高以节省测量成本，又要防止漏测以确保数据完整性；其三，呼应“历史数据再利用”与“智能化测量规划”的行业趋势，鼓励学生利用已有单波束稀疏数据，为高效的多波束测量提供布线下预判，体现数据驱动决策的现代测绘理念。

整体而言，本题以海洋测绘为背景，以几何建模为主线，以优化为核心，检验学生对空

间几何关系解析、参数化建模、大规模路径优化和基于离散数据的区域覆盖等综合能力。

二、考察学生的知识点与能力

本部分按问题一至问题四分开描述，全题可视为基于地形坡度与波束几何的覆盖宽度解析模型及其在大区域测线设计中的应用，在不同问次中依次递进地考察局部几何建模、空间方位扩展、测线网络优化以及数据驱动布线的智能化决策。

问题一 海底坡度条件下的覆盖宽度与重叠率建模

总体考察方向

本问旨在考察学生建立多波束测深系统在倾斜海底条件下的局部几何关系模型的能力。核心在于：当测线方向垂直于坡度走向时，波束发射扇面在坡面上的截取形态发生非对称变化，覆盖宽度不再简单地等于水深除以开角正切的两倍，而是受到上下坡方向不同入射角的影响。要求建立覆盖宽度与水深、开角、坡度之间的解析关系，并据此计算沿垂直于中心测线方向不同偏移位置处的海水深度、覆盖宽度以及与相邻测线的重叠率。

问题 1 与对应模型及北太天元可执行内容

坡度影响下的覆盖宽度几何建模：需根据波束开角、局部水深和坡度角，分别计算沿上坡方向和下坡方向的有效照射距离，两者之和即为该测线位置的实际覆盖宽度。该几何关系可通过对波束边缘射线与坡面交点的求解获得。北太天元可构建该几何关系的数值函数，输入水深、坡度和开角，输出覆盖宽度，并支持批量计算多个位置点。

海水深度沿测线方向的变化推算：已知海域中心点水深和坡度，可推算沿垂直于测线方向（即测线距中心点距离变化方向）各位置的实际水深。北太天元可快速生成距离序列对应的水深序列，为覆盖宽度计算提供输入。

重叠率的逐段计算：相邻测线的重叠率取决于两条测线各自覆盖宽度之和与测线间距的关系。在坡度存在时，需注意上下坡方向的非对称性，重叠区域的计算需精确取相邻条带覆盖区间的交集。北太天元可批量成对计算沿多条平行测线的覆盖区间端点，自动导出各条测线相对于前一条测线的重叠率，并按要求生成表格数据。

计算结果的结构化输出：需将表 1 所列位置的海水深度、覆盖宽度和重叠率按格式填入正文并保存为 Excel 文件。北太天元支持数据框形式的灵活组装和直接写入 xlsx 文件，确保结果呈现规范、精确。

小结：本问重点考察空间几何解析能力、参数化函数的数值实现以及批量计算与结果输出。北太天元作为函数定义与批量调用的计算平台，使学生能够快速完成从几何推演到数值

求解再到结果输出的全流程，奠定后续复杂问题的基础模型。

问题二 任意测线方向下的覆盖宽度空间分布模型

总体考察方向

本问将问题一中的“测线垂直于坡度走向”这一特殊条件推广至“测线方向与坡度法向在水平面投影呈任意夹角”的一般情形。此时，有效坡度不再是实际海底坡度，而是沿测线方向的“视坡度”——即测线方向与最大坡度方向之间夹角的余弦与该最大坡度的乘积。要求建立覆盖宽度随测线方向夹角变化的数学模型，并计算矩形海域内不同测线方向角和不同距中心点距离处的覆盖宽度矩阵。

问题 2 与对应模型及北太天元可执行内容

视坡度的几何投影计算：需将三维海底坡度的法向投影至水平面，然后计算测线方向与该投影方向之间的夹角，进而得到沿测线方向的等效坡度。北太天元可支持向量夹角计算和投影运算，批量处理多个测线方向角与位置点的组合。

覆盖宽度的方向依赖模型构建：将问题一中建立的覆盖宽度函数中的坡度替换为视坡度，同时需考虑测线方向变化导致覆盖宽度对水深变化的敏感度调整。北太天元可封装该方向依赖模型为二元函数（输入测线方向角与位置偏移量，输出覆盖宽度），并支持在表 2 所给出的 8 个方向角和 9 个距离档位上进行全组合批量计算。

计算结果的矩阵化组装与输出：需生成覆盖宽度随测线方向角和距离变化的二维表格。北太天元可利用其矩阵化数据结构和循环控制，快速填充表格并导出，支持后续的可视化和分析。

小结：本问重点考察三维空间几何向水平面投影的解析能力、方向依赖函数的构建以及多参数组合的批量化计算能力。北太天元在向量运算和矩阵化数据处理方面的优势，使学生能够高效完成方向-距离二维全覆盖计算，为问题三的大区域测线设计提供核心计算模块。

问题三 矩形海域全覆盖测线网络的长度优化设计

总体考察方向

本问将问题一和问题二的局部覆盖模型扩展至一个具体的矩形海域（南北长 2 海里、东西宽 4 海里），要求设计一组测线，使得在满足相邻条带重叠率 10%—20%的硬性约束下，测线总长度最短。这是一类典型的带约束的空间布线优化问题，需要在地形坡度已知（西深东浅，坡度 1.5° ）的条件下，确定测线的走向、条数、间距及每条测线的起止位置，从而在保证全覆盖和合理重叠率的前提下最小化总测量航程。

问题 3 对应模型及北太天元可执行内容

测线方向的选择与优化：测线方向直接影响沿测线的视坡度和覆盖宽度分布，进而影响所需测线条数和总长度。通常需比较沿东西方向（平行于等深线）、南北方向（垂直于等深线）或其他角度下的总长度差异。北太天元可对多个候选方向进行扫描计算，评估每种方向下的最优测线间距和总长度，自动筛选出最优方向。

变间距测线的布局优化：由于海域内水深线性变化（西深东浅），覆盖宽度也随之变化，为满足重叠率约束，测线间距不应恒定，而应随水深变化动态调整——浅水区覆盖宽度小，测线应更密集；深水区覆盖宽度大，测线可更稀疏。北太天元可构建以测线位置为决策变量的优化模型，通过迭代搜索或最小二乘拟合，确定使总长度最小且所有相邻测线重叠率均落在 10%—20% 区间内的测线布局方案。

边界覆盖完整性的校核：首尾测线必须确保海域边界被完全覆盖，即边界处条带边缘不向内缩进导致漏测。北太天元可自动检查边界覆盖条件，并据此调整首末测线位置或增加额外测线。

总长度的精确计算：每条测线的长度需根据其在海域内的实际穿行距离确定（可能因测线倾斜而跨越不同的纬度和经度跨度）。北太天元可批量计算各测线长度并累加，输出最终优化结果。

小结：本问重点考察带约束的空间布线优化、变参数下的测线间距自适应设计、候选方向扫描评估以及边界完整性核验能力。北太天元在方向扫描、迭代优化和约束检查方面的灵活性，使学生能够设计出满足工程要求的最短测线网络，体现优化方法在测绘规划中的核心价值。

问题四 基于历史单波束数据的多波束测线智能设计

总体考察方向

本问是整道赛题的综合应用与升华，其核心挑战在于：仅有若干年前单波束测量的稀疏离散水深数据（南北长 5 海里、东西宽 4 海里），需利用这些不完整的历史信息，为新的多波束测量设计测线方案，要求同时满足三个相互制约的目标——条带尽可能全覆盖、重叠率尽量控制在 20% 以下、测线总长度尽可能短。这是一个典型的多目标优化问题，且输入为离散网格数据，需先进行海底地形的空间插值重建，再基于重建地形进行测线布局优化。

问题 4 与对应模型及北太天元可执行内容

离散水深数据的空间插值建模：利用附件提供的离散单波束测深点，构建整个待测海域

的连续水深数字高程模型，插值方法可选择基于最小二乘原理的趋势面拟合、克里金插值或反距离加权插值等。北太天元支持多种插值算法的自定义实现或调用内置插值函数，可生成规则网格上的水深估计值，为后续覆盖宽度计算提供地形输入。

多目标优化模型的构建与求解：需将三个目标（覆盖完整性、重叠率上限控制、总长度最小）纳入统一优化框架。可将其中的漏测面积占比和测线总长度作为主要优化目标，将重叠率超过 20% 的部分长度作为约束或次要惩罚项，通过加权求和或分层优化策略进行求解。北太天元可构建多目标优化迭代框架，支持对候选测线方案进行三维指标评估，并基于进化算法或多起点搜索在方案空间中寻找帕累托前沿。

基于插值地形的逐点覆盖宽度计算与重叠率统计：对每一组候选测线布局，需沿每条测线逐点根据该点水深和视坡度计算覆盖宽度，然后对相邻测线逐段比较覆盖区间，统计漏测区域面积和超重叠率部分长度。北太天元可利用其矩阵运算能力批量完成覆盖区间的成对比较和面积/长度累加，显著加速优化过程中的适应度评估。

结果指标的计算与输出：需输出三个核心指标——测线总长度、漏测面积占比、重叠率超标部分总长度。北太天元可自动汇总这些指标，并支持将优化后测线绘制在地形图上，直观展示条带覆盖效果。

小结：本问重点考察离散数据的空间插值重建、多目标优化建模与求解、基于数字高程模型的覆盖模拟与指标统计等综合能力。北太天元在此提供了从数据处理、地形重建、优化搜索到结果评估与可视化的完整技术栈，使学生能够在历史稀疏数据基础上，设计出兼具科学性与工程实用性的多波束测量布线方案，体现现代测绘中“数据驱动规划”的先进理念。

综合评述：全题从局部几何解析到方向扩展，再到区域测线优化，最终到基于历史数据的智能布线，形成了一条完整的海洋测绘测线设计技术链路。学生在解决过程中，须系统掌握空间几何解析、参数化函数建模、带约束优化方法（含最小二乘拟合和迭代搜索）、空间插值理论以及多目标优化策略等多元方法，并具备将复杂工程问题拆解为可计算子问题的抽象能力。北太天元在这一体系中，不仅作为数值计算引擎，更作为一个集函数定义、批量计算、空间插值、优化迭代与可视化展示于一体的综合性科研平台，为学生高效完成多阶段、多约束、多目标的复杂工程优化任务提供坚实的技术保障。

2020 年全国大学生数学建模竞赛 C 题《中小微企业的信贷决策》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面商业银行在服务实体经济过程中面临的一项

核心风控与经营决策难题——如何在缺乏传统抵押资产、信息不对称程度较高的中小微企业群体中，基于异构、稀疏、多源的交易票据数据，科学评估信贷风险并制定差异化的信贷策略。中小微企业在我国经济体系中占据重要地位，但其“融资难、融资贵”问题长期存在，其根源恰在于银行难以低成本、高效率地辨识企业真实经营状况与还款能力。

命题组期望通过本题目，促使学生在真实金融业务约束下，完成从“多源异构数据清洗与特征工程”到“企业信用风险量化建模”，再到“总额约束下的信贷资源最优配置”以及“突发冲击下的动态策略调整”这一完整决策链条。其深层立意有三：其一，强调“数据驱动信用评价”的量化思维，要求学生从进销项发票的时序、结构、有效性与异常性中提取企业经营健康状况的信号，而非仅依赖人工评级；其二，突出“风险—收益—关系”三元平衡的信贷决策逻辑，即银行不仅考虑违约风险，还需权衡利率定价与客户关系维护（流失率）之间的博弈；其三，呼应普惠金融与宏观冲击应对的现实需求，考察学生在突发公共事件（如疫情）背景下，如何快速调整信贷组合以控制行业集中度和系统性风险暴露，体现金融稳企纾困的政策导向。

整体而言本题需要融合信用评分、回归分析、组合优化、情景模拟等多种方法，检验学生面对真实金融数据时的问题抽象能力、特征构建能力以及多目标决策能力。

二、考察学生的知识点与能力

本部分按问题一至问题三分开表述，全题可视为信用风险评估与资源分配优化的组合问题，在后续问次中依次引入无标注样本的迁移推断和突发情景的扰动建模，逐步逼近真实信贷管理全流程。

问题一 有信贷记录企业的信用风险量化与总额约束下的信贷策略

总体考察方向

本问旨在考察学生针对有标签样本（123 家已获贷企业，含信誉评级与历史经营数据）进行信用风险量化建模的能力，并在年度信贷总额固定的硬约束下，给出每家企业是否放贷、贷款额度与利率水平的组合策略。重点在于从大量进销项发票中提炼出能真实反映企业偿债能力、经营稳定性和上下游议价能力的风险特征，并建立风险与违约概率之间的映射关系，同时将总额分配问题转化为带风险预算的优化决策。

问题一对应模型及北太天元可执行内容

发票数据的清洗与风险特征工程：需对进项与销项发票的数量、金额、时序规律、作废

率、负数发票率等进行系统梳理，构造反映企业经营活跃度、上下游集中度、交易稳定性、回款周期特征以及发票异常比例的指标。北太天元可高效完成大规模发票明细的分组聚合、时间序列重采样、比值计算与异常值识别，尤其支持对作废发票与负数发票占比、上下游供应商/客户数量的统计汇总，为后续建模提供结构化特征矩阵。

信用风险的量化建模——多元非线性回归框架的引入：以企业是否违约或信誉等级作为风险度量目标，利用多元非线性回归模型，捕捉各风险特征对违约概率的非线性边际影响（如交易金额过小或过大均可能预示风险升高，呈现 U 形关系）。北太天元可自定义非线性回归模型结构（如包含幂次项、交互项、分段线性或对数变换），并基于 123 家样本进行参数估计与显著性检验，同时可通过交叉验证评估模型泛化能力，最终输出每家企业量化的违约概率或风险评分。

信贷策略的优化分配：在年度信贷总额固定的前提下，以降低整体违约损失或最大化期望收益为目标，结合风险评分与银行利率区间（4%—15%），同时考虑最低放贷门槛（10 万元）和上限（100 万元），对每家备选企业决策是否放贷及具体额度与利率。北太天元可构建带整数决策变量（是否放贷）和连续变量（额度、利率）的混合整数规划，并调用内置优化求解器实现全局近似最优分配；亦可引入风险预算约束（如整体加权风险不超阈值），使策略兼具风险可控性。

小结：本问重点考察数据清洗与特征工程、多元非线性回归建模、混合整数规划优化三方面能力。北太天元承担从原始发票明细到结构化特征、从回归参数估计到优化策略求解的全流程数值计算，为学生提供统一的建模环境，确保信用评价与资源分配的逻辑一致性。

问题二 无信贷记录企业的风险迁移推断与一亿元总额下的信贷配置

总体考察方向

本问将问题一的有标签建模成果扩展至无信贷记录企业群体（302 家），核心难点在于：这部分企业缺乏历史违约结果和人工信誉评级，需从发票数据中自行推断其风险水平，即完成“领域适应”或“评分迁移”。同时，给定年度信贷总额为 1 亿元这一具体数值，要求输出可执行的信贷策略，相较问题一更具实际业务意义。此问考察学生如何利用问题一建立的回归模型结构，在无真实标签情况下进行稳健的风险排序与额度分配。

问题 2 与对应模型及北太天元可执行内容

特征对齐与标准化处理：需确保 302 家企业所构造的风险特征与问题一中 123 家企业的

特征定义、量纲、计算方法完全一致，避免因特征不一致导致评分偏差。北太天元可复用问题一的特征提取脚本，批量处理 302 家企业的原始发票数据，并自动输出对齐后的特征矩阵，同时支持对缺失或极端值进行统一插补或截尾处理。

无标签企业的风险预测与校准：将问题一训练得到的多元非线性回归模型直接应用于 302 家企业，获得初始违约概率预测值。但由于缺乏真实标签，需进行分布校准——例如比对两组企业在进销项总额、发票活跃度等可观测分布上的差异，对预测得分进行纠偏或调整。北太天元可实现模型预测的批量计算，并支持对预测结果的分布统计、分位数划分以及基于业务规则的分数校准（如根据行业平均交易水平进行聚类内调整），使风险排序更具可靠性。

总额 1 亿元下的多企业信贷组合优化：将 302 家企业连同问题一中未获贷或额度未用满的企业共同纳入备选池，在总额 1 亿元、单户额度 10—100 万元、利率 4%—15% 的约束下，以最大化整体收益或最小化整体风险为目标进行组合选择。北太天元可扩展问题一的优化模型，增大决策变量规模至数百家，并利用其高效的大规模线性/混合整数规划求解能力，快速输出每家企业的放贷额度与利率建议，同时可生成风险-收益前沿图，展示不同风险偏好下的策略变化。

小结：本问重点考察模型迁移应用能力、无标签预测的校准策略以及大规模信贷组合优化的求解能力。北太天元在特征批量复用、预测分发、校准统计和组合优化扩容方面提供连贯的技术支撑，使学生能够系统比较有标签与无标签企业之间的风险分布差异，并据此做出合理的信贷资源配置。

问题三 突发因素影响下的行业冲击建模与信贷调整策略

总体考察方向

本问引入现实中最具挑战性的维度——突发公共事件（以新冠疫情为代表）对不同行业、不同类别企业的非对称冲击。要求不再仅依赖历史发票数据所反映的静态经营状况，而是主动判断哪些行业受冲击更为严重（如餐饮、零售），哪些可能反而受益（如医药、在线服务），并据此对问题二给出的初始信贷策略进行系统性调整。该问重点考察学生对宏观冲击传导机制的理解，以及在部分信息可估但无法精确预知的情景下，如何通过情景模拟和规则调整制定稳健的信贷应对方案。

问题 3 与对应模型及北太天元可执行内容

（一种思路）行业脆弱性/韧性指数的构建：需根据附件中各企业的行业标签，结合疫情对不同行业的业务中断程度、供应链冲击、需求萎缩等定性知识，量化设定各行业的冲击系

数（如营业收入预计下降比例、应收账款回收延迟天数等）。北太天元可辅助建立行业维度的参数向量，并通过交互界面快速调整不同行业系数，支持多情景定义。

企业层面抗冲击能力的二次评估：在行业冲击系数的基础上，叠加企业自身的财务韧性指标（如现金储备可由进销项差额间接推断、上下游集中度影响供应链断裂风险），形成每个企业的综合抗风险评分，从而修正问题二中的违约概率预测。北太天元可实现原有违约概率与行业冲击系数、企业韧性因子的非线性组合计算（例如多元非线性回归的扩展形式，加入行业交互项），生成调整后的风险评分。

信贷调整策略的制定：在总额 1 亿元不变的前提下，根据调整后的风险评分重新进行信贷组合优化，但需额外引入行业集中度约束（如受冲击严重行业的总贷款占比不得超过某一上限）和紧急流动性支持政策（如允许对部分受困但长期经营良好的企业提高额度或降低利率）。北太天元可扩展优化模型，加入行业上限约束、政策定向约束，并支持对利率进行差异化调整（如将受冲击严重行业的利率下限调低）。进一步，可生成多个冲击强度情景（乐观、刚好达标、悲观），分别求解放贷策略，供银行决策参考。

客户流失率与利率的联动再平衡：由于附件 3 提供了年利率与客户流失率的关系，在突发冲击下，银行若大幅提高风险溢价可能引发优质客户流失，加剧损失。北太天元可将流失率曲线嵌入目标函数，使优化在权衡“违约损失”与“客户流失损失”之间进行动态平衡，从而得到更符合银行长期利益的利率设定策略。

小结：本问重点考察情景建模、行业冲击系数设定、抗风险能力二次修正、带行业约束的组合再优化以及客户流失—利率联动平衡的综合决策能力。北太天元在此提供多情景循环仿真、约束动态添加、目标函数调整和结果对比分析的全套工具支持，使学生能够在不确定性环境下制定兼具风控底线和纾困导向的差异化信贷调整策略，体现金融决策中的社会责任与经营稳健性的统一。

综合评述：全题从有标签企业的风险量化起步，到无标签企业的迁移推断与大规模组合配置，再到突发冲击下的动态策略调整，形成了一条完整的商业银行中小微企业信贷决策 workflow。学生在解决过程中，须系统掌握特征工程与数据清洗、多元非线性回归建模、混合整数规划优化、模型迁移与校准、情景分析与约束调整等多元方法，并具备将宏观经济认知转化为模型参数的抽象能力。北太天元作为一体化数值计算平台，不仅支撑从数据预处理、回归估计、优化求解到情景仿真的所有计算环节，更因其灵活的矩阵运算和自定义函数能力，使学生在模型迭代和策略对比中获得高效的技术响应，充分契合竞赛对复杂真实问题求解的

创新性、系统性与落地性并重的要求。

2022 年认证杯国际数学建模竞赛 C 题《人类活动分类》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面可穿戴计算与行为识别领域中一项极具挑战性的核心问题——基于多源惯性传感器（加速度计、陀螺仪、磁力计）的多模态时序数据，对人体日常活动进行高精度分类。该问题处于传感器技术、信号处理、模式识别与机器学习的交叉前沿，在老年监护、运动分析、康复医疗、虚拟现实和人机交互等领域具有广泛的现实应用价值。数据本身具有高维度、多通道、时序依赖、个体风格差异显著等特点，且每一类活动的样本量相对有限（每个活动仅 480 个样本片段），这对模型的泛化能力和抗过拟合能力提出了严峻考验。

命题组期望通过本题目，促使学生在真实传感器数据约束下，完成从“多通道时序信号的特征工程”到“多类分类器的设计与训练”，再到“模型泛化能力的系统评估”以及“过拟合问题的诊断与克服”这一完整机器学习应用流程。其深层立意有三：其一，强调“特征驱动分类”的经典模式识别思维，要求学生从原始 45 维时序信号中手工或自动提取具有类间区分度和类内稳定性的判别性特征，而非盲目依赖端到端黑箱模型；其二，突出“有限数据下的泛化性保障”这一机器学习核心矛盾，引导学生在数据量有限、个体间差异大的情况下，通过交叉验证、正则化、特征选择等手段构建真正可部署于新用户的稳健模型；其三，呼应“可解释性与计算效率”的工程需求，鼓励学生从多种分类器（包括集成学习、核方法等）中筛选出在准确率、训练速度和内存占用上综合最优的方案，体现学术研究与实际系统部署之间的平衡意识。

整体而言，本题以人体活动识别为载体，以特征工程和分类器设计为主线，以泛化能力评估和过拟合防控为深度探究方向，检验学生对完整监督学习任务链的系统掌握和灵活运用能力。

二、考察学生的知识点与能力

本部分按问题一至问题三分述，每一问题内部采用“总一分一总”结构，先点明该问整体考察方向，再按子任务拆解对应的模型类别与北太天元可支撑的计算环节，最后小结该问的综合能力目标。全题内核可视为标准的多类模式识别流程，在不同问次中依次递进地考察

分类器构建、泛化性评估和过拟合防控这三个机器学习中最核心的实践环节。

问题一 多类人类活动的特征设计与分类算法构建

总体考察方向

本问旨在考察学生面对高维多模态时序传感器数据时，设计有效特征表示并选择合适的分类算法实现 19 类活动高精度识别的能力。核心难点包括：原始数据为每个样本片段（5 秒）包含 45 个通道、每个通道 125 个时间点，直接使用原始时序作为特征将导致维度灾难且难以捕捉活动本质的运动模式；不同活动（如坐、站、躺、走、跑、跳跃、骑车等）在运动频率、幅度、对称性、姿态变化等方面的差异需要针对性的特征设计；8 位受试者各自的活动风格差异使得分类器必须对个体间变化具有一定鲁棒性。

问题 1 对应模型及北太天元可执行内容

- **数据加载、解析与结构化整理：**需遍历 19 个活动文件夹 $\times$ 8 个受试者 $\times$ 60 个片段文件，将每个片段的 45 通道 $\times$ 125 行数据正确读入并标记其活动类别标签。北太天元支持层级化文件夹遍历、文本数据批量读取和大型多维数组的构建，可将全部 9120 个样本（ $19 \times 8 \times 60$ ）组织为统一的数据张量，为后续特征提取提供高效的索引和切片操作基础。
- **时域与频域特征工程：**需为每个传感器通道的 125 个时间点序列计算一系列统计量和变换域特征，如均值、方差、峰值、过零率、均方根、四分位距等时域特征，以及通过频谱分析提取主频分量、频谱质心、频带能量比等频域特征。此外，还需对不同身体部位（躯干、左右臂、左右腿）的传感器信号进行交叉组合，提取如两两轴间的相关系数、身体部位间的相对运动一致性等空间关联特征。北太天元内置了丰富的统计函数、快速傅里叶变换接口和矩阵运算能力，可在一次遍历中为每个样本批量生成数百维特征向量，构建完整的特征矩阵。
- **特征选择与降维：**为避免特征冗余引发的维度诅咒和过拟合，需采用过滤式或包裹式特征选择方法，筛选出对 19 类分类最具判别力的特征子集；亦可采用主成分分析或线性判别分析等降维技术，将高维特征投影至低维判别空间，在压缩特征维度的同时保持或提升类间可分性。北太天元支持协方差矩阵计算、特征值分解以及多种特征重要性排序指标的快速实现，助力学生进行特征筛选的定量化决策。

- **多类分类器的训练与比较：**需在训练集上训练多种代表性分类器，包括但不限于支持向量机、随机森林、核费舍尔判别分析以及基于梯度提升树的极限梯度提升（XGBoost）等，并通过验证集比较各分类器在准确率、精确率、召回率和 F1 分数上的综合表现。北太天元虽以数值计算见长，但其开放的架构支持学生自主实现上述分类器的核心算法（如 SVM 的序列最小优化训练、决策树的递归分裂、集成学习的 Boosting 轮次迭代），尤其适合竞赛场景下对算法内部机制的深入理解和调试，而非简单调用黑箱库函数。

- **模型参数调优与交叉验证：**需对各分类器的关键超参数（如 SVM 的核参数与惩罚系数、随机森林的树数量和分裂准则、XGBoost 的学习率与最大深度等）进行系统搜索，结合交叉验证选择最优参数组合。北太天元可支撑多轮参数扫描和性能记录，通过网格搜索或随机搜索策略自动寻找最优配置。

小结：本问重点考察多通道时序数据读取与预处理、时域频域组合特征工程、特征选择与降维、多类分类器设计与比较、超参数调优等完整分类建模能力。北太天元作为底层数值计算平台，为在无预置机器学习工具包的情况下自主实现核心算法、灵活控制计算细节和进行大规模特征批处理提供了坚实的编程基础，强化了学生对算法原理的理解而非单纯依赖高级封装。

问题二 有限数据条件下的模型泛化能力评估

总体考察方向

本问从“构建分类器”上升到“审视分类器”的元层面，要求学生在明确意识到数据采集成本高昂、样本量有限的前提下，设计可行的方法来系统评估模型的泛化能力——即模型在未见过的受试者数据上是否仍能保持与训练集上相近的分类性能。泛化能力的评估不能简单依赖随机划分训练/测试集，而必须模拟真实部署场景：新用户的活动数据不会出现在训练集中，且个体间的运动风格差异可能使模型失效。该问考察学生对机器学习评价范式的深刻理解，尤其是对“用户无关”评估协议的设计能力。

问题 2 对应模型及北太天元可执行内容

- **留一受试者交叉验证方案的构建：**由于 8 位受试者彼此独立且活动风格各异，最贴近实际部署场景的评估方法是每次将一位受试者的所有数据作为测试集，其余 7

位受试者的数据作为训练集，轮转 8 次后取平均性能。该方案直接测量模型面对全新用户时的表现。北太天元可灵活实现按受试者 ID 分组的索引划分，自动化执行 8 折交叉验证循环，记录每一折的各类别准确率和整体性能，并计算均值与标准差，为泛化能力提供无偏估计。

- **训练集规模缩减实验与学习曲线绘制：**为研究“数据量有限”的具体影响程度，可设计不同训练子集规模（如每位受试者取 10、20、40、60 个片段）下的泛化性能变化趋势，绘制学习曲线，观察模型性能是否随数据增加而持续提升、是否进入平台期。北太天元可自动化完成多组规模实验，并绘制训练集准确率和验证集准确率随样本量变化的曲线图，定量展现数据稀缺性对泛化能力的制约。

- **跨活动类别的性能细粒度分析：**泛化能力的整体指标可能掩盖某些特定活动（如跑步与快走、阶梯上行与下行、立式与卧式骑车）之间的混淆程度。需构建每个活动的单独精确率和召回率，以及两两活动之间的混淆矩阵，识别哪些活动对泛化最具挑战性。北太天元可批量生成混淆矩阵并可视化热力图，辅助判别哪些活动由于运动模式相似而导致分类器在新用户上易出错。

- **泛化能力的统计显著性检验：**为证明不同分类器之间泛化能力的差异并非随机波动，可在 8 折交叉验证结果上执行配对统计检验。北太天元支持基础的统计推断计算，使学生能够量化评估不同方法泛化性能差异的置信程度。

小结：本问重点考察机器学习实验设计的严谨性、用户无关的交叉验证协议、学习曲线实验、混淆矩阵细粒度分析以及统计显著性检验等泛化评估方法论。北太天元在多层循环控制、批量实验自动化、结果汇总与图形化分析方面提供灵活支撑，使学生能够对模型在新用户上的真实表现给出可靠、可解释的量化评价。

问题三 过拟合问题的诊断、克服与通用性保障

总体考察方向

本问是机器学习实践中最具现实价值的一环——识别并解决过拟合，确保分类算法不仅能在训练集和验证集上表现优异，更能广泛适用于未知用户的动作分类任务。过拟合在本问题中可能源于多种因素：特征维度过高而样本量不足、分类器模型容量过大（如深度决策树、高次核函数）、训练数据中混入了个体特定的噪声模式等。要求学生在问题二建立的泛化评

估框架基础上，系统性地采取多种策略来抑制过拟合，并验证这些策略的实际效果。该问考察的是模型正则化、复杂度控制和鲁棒性增强的综合能力。

问题 3 对应模型及北太天元可执行内容

- **过拟合的诊断与量化：**需通过比较训练集准确率与验证集准确率之间的显著差距来诊断过拟合的存在与程度。若训练集接近完美而验证集性能明显落后，则判定发生过拟合。北太天元可同时计算两组准确率并输出差距值，作为过拟合风险指标，并支持在不同训练轮次或不同模型容量设置下跟踪该差距的变化轨迹。
- **正则化技术与模型复杂度控制：**需采用多种手段控制模型复杂度。对于基于树的方法，可限制树的深度、最小分裂样本数和叶节点最小样本数；对于支持向量机，可调整软间隔惩罚系数和核带宽；对于集成方法，可控制基学习器的数量和学习率。此外，特征选择本身也是一种有效正则化——通过减少无关或噪声特征降低模型过拟合倾向。北太天元允许学生在自主实现的分类器训练循环中灵活加入正则化项、早停机制和剪枝逻辑，并在不同正则化强度下比较训练集与验证集性能的收敛情况。
- **数据增强与噪声注入：**为增加训练样本的有效多样性，可对原始传感器信号进行轻微变换，如时间轴扰动、幅值缩放、添加高斯噪声或随机通道丢弃，生成更具泛化能力的增广样本。北太天元支持向量化的信号变换操作，可在内存中高效生成大规模增广数据集，无需重复读取磁盘文件。
- **集成学习的抗过拟合优势利用：**随机森林和梯度提升树等集成方法天然具有降低方差的特性，可重点比较集成方法与单一强分类器在过拟合程度上的差异，展示集成策略对泛化能力的提升效果。北太天元可支撑学生对不同集成规模（树的数量）下的过拟合趋势进行分析，验证“更多树并不一定导致更多过拟合”的集成学习理论。
- **最优模型的最终泛化能力验证：**在综合应用上述多种抗过拟合策略后，需在留一受试者框架下重新评估最终模型的泛化性能，并与问题一中未加正则化的基准模型进行对比，量化改进幅度。北太天元可完成基准模型与正则化模型在多折交叉验证上的成对比较，并输出改进百分比和统计显著性结论。

小结：本问重点考察过拟合的诊断方法、模型复杂度管理、正则化策略（包括结构限制、早停、剪枝、特征选择）、数据增强技术以及集成学习的方差抑制效应。北太天元凭借其开

放的算法实现环境和灵活的实验控制能力，使学生能够深入探索各种抗过拟合手段的内在机理，并通过定量对比验证每项措施的实际贡献，最终构建出兼具高精度和强泛化能力的实用动作分类系统。

综合评述：全题从特征工程与分类器构建起步，到严谨的泛化能力评估协议设计，再到系统性的过拟合防控与通用性保障，完整覆盖了一个机器学习项目从“可用”到“可靠”再到“可推广”的三个关键阶段。学生在解决过程中，须系统掌握时序信号特征提取（时域、频域、时频域）、多类分类器原理（支持向量机、核判别分析、随机森林、极限梯度提升）、特征选择与降维、留一交叉验证、学习曲线实验、模型正则化与早停、数据增强、混淆矩阵分析等全套技能。北太天元作为贯穿全程的数值计算基础平台，虽然不直接提供黑箱机器学习库，但正因其开放性和灵活性，使学生能够亲手实现核心算法、精细控制训练流程、自由设计实验方案，从而真正理解每个环节的数学本质和工程考量，这对培养具备深厚建模功底和算法实现能力的复合型人才具有不可替代的价值。

2023 年全国大学生数学建模竞赛 C 题《蔬菜类商品的自动定价与补货决策》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面生鲜零售业中一类典型、高频且极具现实挑战的运营决策问题——短生命周期生鲜品的联合补货与定价决策。蔬菜类商品具有保鲜期极短、品质时变、需求波动大、供给季节性强、损耗不可逆等多重特性，使得其补货与定价无法简单沿用传统工业品的库存理论，而必须融合需求感知、成本传导、空间约束与风险对冲等多元决策要素。

命题组期望通过本题目，促使学生在真实业务约束下，完成从“数据驱动下的规律挖掘”到“多目标、多周期、多品类联合决策”，再到“精细化管理与信息增补建议”的完整决策链条推演。其深层立意有三：其一，强调“数据—规律—决策”的闭环思维，要求学生不满足于统计描述，而是走向可执行的商业策略；其二，突出“不确定性环境下的最优化”理念，即商超在未知当日具体进货价格与单品可供货量的前提下，需依赖历史信息和需求分布做出稳健决策；其三，呼应新商科背景下数学建模与计算工具的融合，鼓励学生借助高效数值计算环境（如北太天元）实现大规模数据的清洗、建模与求解。

整体而言，本题并非单纯考查某一经典统计模型的套用，而是检验学生对现实复杂系统进行抽象、简化和量化，并在多约束条件下寻求可行最优解的综合建模素养。

二、考察学生的知识点与能力

本部分按问题一至问题四分开表述，值得一提的是本题目在本质上可视为报童模型在生鲜多品场景下的扩展应用——核心在于权衡过量补货带来的损耗风险与补货不足造成的销售损失，并在此基础上叠加定价、品类关联、陈列空间等现实约束，各问均在不同层面回归或延展这一思想内核。

问题一 蔬菜各品类及单品销售量的分布规律与相互关系分析

总体考察方向：本问旨在考察学生针对大规模零售流水数据的探索性分析能力，重点在于识别不同品类及单品在时间维度上的销售波动形态、季节性特征、生命周期规律，以及品类间、单品间的替代或互补关系。该问不涉及后续决策优化的问题，而是为后续补货与定价模型提供统计学依据，故要求分析结果兼具统计稳健性与业务可解释性。

第一问对应模型及北太天元可执行内容：销售量分布规律的刻画：需采用描述性统计与分布拟合方法，对各品类日销售量的集中趋势、离散程度、偏态及尾部特征进行量化；对单品则需进一步考察其销售频次与零销天数。北太天元可高效完成大规模数据的分组汇总、分位数计算、直方图绘制及常见概率分布（如泊松、负二项）的参数拟合与拟合优度检验，以确定各销售序列的最优分布形态。

时间关联规律的挖掘：需从周内效应、节假日效应及长期趋势维度分析销售量与时间的耦合关系，并识别品类自身的季节性供应波动。北太天元可借助时间序列分解工具（如移动平均、STL 分解）提取趋势项、周期项与残余项，并计算自相关函数与偏自相关函数，以量化销售滞后影响，为后续需求预测提供时滞特征。

品类间与单品间关联关系的建模：需构建成对关联度量指标，识别品类间是呈现需求替代（如叶菜与茄果此消彼长）还是需求互补（如配菜与主菜同步增长），同理对单品亦需进行关联性筛查。北太天元可支持计算皮尔逊相关系数、斯皮尔曼秩相关系数及更稳健的互信息系数，并可通过热力图或网络图可视化关联强度，辅助筛选强关联对；对于高维单品数据，还可引入聚类分析（如 K-means 或层次聚类）将相似销售模式的单品归组，从而简化关联结构。

小结：本问重点考察数据预处理、分布拟合、时间序列分解、相关性度量与聚类分析等多元统计与数据挖掘技能的协同运用，要求学生具备从原始数据中提炼可量化业务规律的能力，北太天元在此承担全流程数值计算与可视化输出，为后续决策模型提供输入特征。

问题二 以品类为单位的补货总量与定价联合决策

总体考察方向：本问将分析提升至决策层面，核心在于刻画各品类销售总量与其成本加成定价之间的响应关系（即需求价格弹性），并在此基础上，联合优化未来一周逐日的补货总量与销售定价，以实现商超期望收益最大化。该问的本质是多品类、多周期的报童扩展问题，需同时处理需求不确定性、价格对需求的牵引作用、补货提前期固定且信息不完整等现实因素。

第二问对应模型及北太天元可执行内容：需求与价格关系的建模：需基于历史日销售总量与当日实际定价（可由售价与批发价推算成本加成比例），拟合各品类的需求价格响应函数。鉴于线性关系仅适用于有限区间，可考虑幂函数、对数或指数形式的非线性需求曲线。北太天元可调用非线性回归或广义线性模型框架，完成参数估计、残差诊断与模型选择，并输出各品类的价格弹性系数，作为后续优化的关键输入。

未来一周逐日补货总量与定价的联合优化：需构建以商超日收益（销售收入扣除进货成本与未售出损耗成本）最大化为目标，以各品类日补货量、定价上限下限及历史损耗率为约束的随机规划模型。由于当日进货价格在决策时尚不确定，需将批发价格视为随机变量并基于附件 3 建立其分布估计，从而形成两阶段随机决策结构。北太天元可高效实现随机样本生成（如蒙特卡洛模拟）、目标函数的数值期望计算，并可调用内置的线性/非线性规划求解器，针对该优化问题进行迭代求解；对多品类耦合约束，还可引入拉格朗日松弛或罚函数法进行近似处理。

成本加成定价的校准：需确保优化出的定价策略不低于进货成本与必要加成率，同时兼顾打折清损的折价策略。北太天元可支持分段定价规则的编码，并在优化循环中嵌入动态折扣逻辑，实现“定价—补货”联合迭代。

小结：本问核心考察需求价格弹性建模、随机规划与数值优化求解能力，要求学生在不完全信息下构建可操作的周度补货与定价日历。北太天元在随机模拟、约束优化求解及敏感度分析中发挥核心计算支撑作用，使学生能够量化不同风险偏好下的最优策略差异。

问题三 有限陈列空间约束下的单品级补货与定价优化

总体考察方向：本问进一步贴近业务现实，引入最核心的空间约束——可售单品总数须控制在 27 至 33 个之间，且每个单品订购量不低于最小陈列量 2.5 千克，同时要求在满足各品类整体需求覆盖的前提下，制定 7 月 1 日当日的单品补货量与定价策略。该问将决策粒度从品类级下放至单品级，并从多周期简化为单周期决策，但其组合爆炸特性（品种选择与数

量分配）使得求解难度显著上升，属于带资源分配的组合优化与报童问题的高度融合。

第三问对应模型及北太天元可执行内容

品种选择与数量分配的两阶段决策：需先确定哪些单品入选当日可售清单（满足 27 至 33 个总数约束），再为入选单品分配补货量及定价，同时确保各品类总体需求被覆盖（不可过度集中于某几个品类）。北太天元可构建离散连续混合决策变量，将品种选择表示为 0-1 变量，补货量与价格表示为连续变量，形成混合整数非线性规划，并利用内置的混合整数优化工具或启发式算法（如遗传算法、模拟退火）进行求解，尤其适合处理中等规模单品数据。

最小陈列量约束与损耗风险权衡：每个入选单品的补货量不得低于 2.5 千克，这意味着无法用极小量试销，必须承担基础库存风险。北太天元可将该约束直接写入优化模型的下界向量，并在目标函数中结合附件 4 提供的单品级损耗率，精确计算每增加一个单品所带来的边际期望损耗成本，从而自动筛选出单位空间收益潜力更高的单品组合。

需求满足度的度量与约束：需量化“尽量满足市场对各品类蔬菜商品需求”这一柔性目标，可采用各品类实际补货总量与其历史需求期望值的比值作为覆盖度，并设定各品类最小覆盖率要求。北太天元可灵活定义多目标加权函数或罚函数，将覆盖率约束转化为目标项或约束条件，实现品种丰富度与收益目标的帕累托权衡。

基于 6 月 24 至 30 日可售品种的迁移建模：需明确 7 月 1 日可补货品种只能从该周可售品种集合中选取，不允许引入未见新品。北太天元可完成品种集的逻辑筛选与索引映射，确保优化空间合规，并支持快速重跑不同候选品种子集下的最优解对比，辅助判断最终方案稳定性。

小结：本问重点考察混合整数规划建模、组合约束处理、多目标权衡分析及启发式搜索算法应用，要求学生具备在强离散约束下寻找经济可行解的能力，北太天元作为数值计算平台，能够有效支撑非凸、离散、非线性混合问题的求解迭代与结果后验。

问题四 数据采集建议及其对决策的改进价值

总体考察方向

这个是一个开放性问题，反向考察学生的数据感知能力与决策链路闭环意识，即站在商超实际运营角度，识别当前数据体系中缺失但对补货与定价至关重要的额外信息，并阐明其如何提升模型效果或简化决策难度。该问不要求建模求解，但非常考验学生对前序建模过程中假设条件和不确定因素来源的敏感度。

第四问对应分析内容与北太天元可辅助的验证方向

建议采集的数据类型，至少包括但不限于：

每日各品类及单品的实际到货量与供应商交货准时率（用于修正批发价格随机分布的先验）；门店实时库存盘点数据（含当日开市前实际库存、营业中补货记录、闭市剩余量），以精确计算真实损耗而非依赖统一损耗率；商品品质分级或品相评分，以便区分正常销售与折价清货的不同价格策略；天气、节假日、周边竞争活动等外部协变量，用于提升需求预测精度；顾客购买篮数据（组合购买记录），直接推断品类/单品间的真实替代与互补强度，而不再仅依赖历史销售的共现性。

上述数据对问题解决的帮助：可显著降低需求分布估计的偏差，使定价与补货决策更加贴近实际进销存动态；可改进随机优化中的先验设定，减少因信息不对称导致的保守或激进策略；可将损耗率从固定参数升级为依赖陈列时长与品相退化的动态函数，从而在定价中实现更精准的清损时机决策；外部变量的加入可提升周度预测的可靠度，减少紧急调拨或清仓压力。

北太天元的验证能力：学生可论证利用北太天元对新采集数据进行增量建模，例如通过协变量回归检验天气因素对叶菜需求的显著性，或通过仿真对比引入动态损耗率前后最优收益的变化幅度，从而量化新数据的边际价值，增强建议的说服力。

小结：本问重点考察数据完备性评估、不确定性来源识别以及数据价值论证能力，要求学生能够从前序模型的敏感性和脆弱性出发，反向推导出关键缺失信息，并合理解释其对决策改善的机理，体现从“用数据”到“要数据”的主动决策思维。

综合评述：全题由浅入深，从描述性分析到多约束联合决策，再到数据反向设计，完整覆盖了生鲜零售决策的数智化转型路径。学生在解决过程中，既要掌握分布拟合、时间序列、相关性分析、随机规划、混合整数优化、蒙特卡洛模拟等多元方法，更要具备模型假设与现实对照的批判意识。北太天元在这一体系中不仅承担传统的数值计算任务，更因其灵活的矩阵运算、优化工具箱和仿真能力，成为支撑学生快速迭代建模假设、检验模型稳健性以及可视化决策边界的理想国产计算平台，其开放性亦便于学生按需自定义算法，契合竞赛对创新性与可操作性的双重追求。

2024 年全国大学生数学建模竞赛 C 题《农作物的种植策略》赛题解析

一、命题立意

本赛题的命题初衷，在于引导参赛学生直面中国北方山区有限耕地资源约束下，农业种植决策中的多周期、多品类、多地块、多不确定性因素的复杂优化问题。乡村耕地类型多样

（平旱地、梯田、山坡地、水浇地、普通大棚、智慧大棚），每种类型适宜种植的作物类别各异，且受茬茬约束、豆类轮作要求、种植分散度管理、产销平衡限制等现实规则制约，使得种植策略的制定远超简单的单品种面积分配，而成为一个兼具空间配置、时间排程与风险管理的系统性决策命题。

命题组期望通过本题目，促使学生在真实农业经营场景中，完成从“静态产销平衡规划”到“动态不确定性下的稳健优化”，再到“关联效应与仿真比较”的完整决策进阶。其深层立意有三：其一，强调“耕地资源精细画像”与“作物生态适应性”的匹配思维，要求学生尊重土地属性差异，实现因地制宜；其二，突出“产量—价格—成本—需求”四重不确定性下的风险管理理念，引入条件风险价值等工具进行损失控制，而非仅追求期望收益最大化；其三，呼应智慧农业与国产计算工具融合趋势，鼓励学生借助北太天元的高效数值计算与优化求解能力，在大规模时空决策空间中寻找可执行、可解释的种植方案，并具备对不确定因素进行模拟仿真与敏感性分析的技术素养。

整体而言，本题希望检验学生在农业系统建模中对约束精细刻画、对不确定性量化对冲、对多目标权衡决策的综合研究能力，同时考察其能否在模型复杂度与求解可行性之间取得合理平衡。

二、考察学生的知识点与能力

本部分按问题一至问题三分条陈述，全题内核可视为多阶段、多地块、多作物的随机规划与组合优化问题，在不同问次中分别引入确定性约束、不确定性刻画和关联效应建模，逐步逼近真实农业决策的复杂全貌。

问题一 稳定参数下的多周期种植面积分配与产销平衡优化

总体考察方向：本问旨在考察学生在未来七年（2024—2030）种植规划中，面对稳定的产量、价格、成本与预期销售量假设时，如何构建一个满足多重空间约束、轮作约束和田间管理约束的确定性优化模型。重点在于将繁杂的现实规则转化为数学结构，并针对“超额产出滞销浪费”与“超额产出折价出售”两种情境，分别给出差异化最优方案。该问是后续不确定性分析的基础基准，要求模型具备清晰的可解释性和可扩展性。

本小问对应模型及北太天元可执行内容

- **地块-作物-季节的三维匹配建模：**需建立各年度、各地块、各季节（一季或两季）的作物种植面积决策变量，并严格限制每块耕地的面积总和不超过其可用面积，且每种作物的种植需与其地块类型适配（如水稻仅限水浇地、食用菌仅限普通大棚

等）。北太天元可借助其矩阵化数据结构和稀疏变量定义能力，高效构造超大规模 0-1 或整数决策数组，并通过逻辑索引快速实现地块类型与作物的合法性筛选，显著降低建模复杂度。

- **重茬约束与豆类轮作约束的时序建模：**需确保同一地块上同种作物在相邻年度或相邻季节不连续种植，同时要求 2024 至 2030 年间每个地块至少有一次豆类作物种植记录。北太天元可通过构建年份-地块-作物的状态转移矩阵，利用条件约束生成函数，自动生成跨期互斥逻辑和最低覆盖要求的约束表达式，并调用内置混合整数线性规划求解器进行可行性判定。

- **种植分散度与最小种植面积的软约束处理：**需量化“每种作物每季种植地不能太分散”以及“每种作物在单个地块种植面积不宜太小”这两项管理诉求，可采用设定每季每种作物的最大地块使用数量和最小单块种植面积下限的方式。北太天元支持用户自定义非线性或线性化罚函数，将这些管理偏好转化为目标函数中的惩罚项或硬性约束边界，并通过参数扫描对比不同分散度容忍度下的收益变化。

- **两种超额产出处理方案的最优化求解：**方案（1）要求超产部分完全损失，即各作物总产量不得超过预期销售量，形成刚性上限约束；方案（2）允许超产部分以半价出售，相当于在目标收益函数中引入分段线性收入项。北太天元可分别构建线性与分段线性目标函数，调用混合整数线性规划求解器进行全局最优求解，并自动输出每个地块每年的详细种植清单，直接填入指定的模板文件。

小结：本问重点考察确定性混合整数线性规划的建模与求解能力，要求学生熟练掌握空间、时序、管理规则三类约束的数学转化，以及分段线性目标函数的处理技巧。北太天元在大规模整数规划建模、约束批量生成与求解结果结构化输出方面发挥核心支撑作用，确保七年周期、多地块、多季度的复杂方案可在可接受时间内求解并落地。

问题二 多重不确定性下的稳健种植策略与风险管控

总体考察方向

本问将问题一的静态框架全面推向动态不确定环境，要求同时处理预期销售量趋势增长（小麦、玉米年增 5%—10%，其他作物±5%波动）、亩产量年际±10%波动、种植成本年均增长约 5%、粮食价格稳定、蔬菜价格年均增长约 5%、食用菌价格年降 1%—5%（羊肚菌降 5%）等多源不确定性因素。核心考察目标已从“寻找最优解”升级为“在多种风险源叠加下，寻

找兼顾期望收益与极端损失控制的稳健种植策略”，这要求引入风险管理工具（如条件风险价值）和高效随机仿真优化方法。

问题 2 对应模型及北太天元可执行内容

- **不确定性因素的分布建模与场景生成：**需对每一年的每一种不确定性参数（需求增长率、产量波动系数、成本增长率、价格变化率）设定合理的概率分布形态，并考虑其年度间的自相关性。北太天元可支持用户自定义概率分布（如三角分布、均匀分布、正态分布截断处理），并调用其随机数生成函数进行大规模蒙特卡洛场景抽样；为进一步提升计算效率，可采用拟蒙特卡洛方法生成低差异序列，使场景覆盖更均匀，加快目标函数期望的收敛速度。

- **基于条件风险价值的多目标种植决策模型：**在追求期望收益最大化的同时，需控制尾部损失风险，即对产量过剩或价格下行导致的收益大幅波动进行约束。典型做法是以期望收益为主目标，以条件风险价值为风险度量指标，将其纳入目标函数或约束条件，形成多目标优化框架。北太天元可支持用户自定义条件风险价值的数值近似计算，并在迭代优化过程中动态评估每套种植方案在不同随机场景下的收益分布，从而引导搜索算法向低风险前沿靠近。

- **启发式/进化求解策略的实施：**由于本问引入随机场景后决策空间急剧膨胀，混合整数线性规划难以直接求解，需借助启发式算法进行近似最优搜索。北太天元具备灵活的用户自定义迭代框架，可嵌入差分进化算法、遗传算法（含 NSGA-II 等多目标进化算法）等元启发式方法，利用其种群并行搜索能力，在大规模组合空间中逼近帕累托前沿。北太天元的矩阵化计算特性使得种群中每个个体的适应度评估可向量化执行，大幅加速多场景下的目标值批量计算。

- **多种参数情景下的对比分析与稳健性检验：**需对小麦玉米增长率取边界值（5% 与 10%）或其他作物波动幅度的极值情景进行额外求解，验证推荐方案在不同市场预期下的表现稳定性。北太天元可便捷实现参数循环扫描与结果自动汇总，并支持绘制风险-收益散点图或箱线图，辅助判断最优种植策略在风险维度上的可靠程度。

小结：本问重点考察随机规划建模、蒙特卡洛/拟蒙特卡洛场景生成、条件风险价值风险度量、启发式优化（尤其是差分进化与多目标进化算法）的综合运用能力。北太天元凭借其数值计算、随机模拟、自定义迭代优化框架三位一体的环境，支撑学生完成从场景生成到风

险优化再到结果鲁棒性检验的完整链路，体现了国产计算平台在不确定决策问题中的强大适配性。

问题三 作物关联效应引入下的综合优化与仿真比较

总体考察方向

本问在问题二的基础上，进一步纳入经济学意义上的作物间可替代性和互补性，以及预期销售量与销售价格、种植成本之间的内生相关性（例如某种作物价格上涨可能伴随需求下降，或某种作物种植成本增加可能引导农户转向替代作物）。考察内容发生改变，要求通过模拟数据生成与对比分析，系统比较考虑关联与否的方案差异，从而验证关联因素对最优种植结构的实质性影响。

问题三与对应模型及北太天元可执行内容

- **替代性与互补性关系的量化建模：**需定义作物对之间的交叉影响系数——若两种作物互为替代（如不同叶菜），则一种种植面积增加会拉低另一种的预期价格或需求；若互为补充（如主粮与配菜），则一种的产出增长可能带动另一种的需求上升。北太天元可支持构建关联弹性矩阵，将作物间的交叉弹性引入需求函数或价格函数中，使目标函数的计算不再基于独立参数，而是根据决策变量组合动态调整。
- **需求-价格-成本内生联动模型的构建：**需打破问题二中价格和需求量彼此独立的假设，建立如价格随销售量增加而下降（需求曲线）或种植成本随行业种植面积扩大而上升（供给拥挤效应）等反馈机制。北太天元可自定义非线性目标函数和约束表达式，并通过嵌入迭代更新机制，在进化搜索的每一代中重新计算当前方案对应的内生价格和需求量，确保目标评估与决策变量耦合一致。
- **基于仿真数据的多情景求解与对比框架：**由于真实作物关联参数难以从附件直接获取，需自行设定合理的模拟参数区间，生成多组关联强度不同的仿真数据集，并在每组数据下分别运行问题二和问题三的模型。北太天元可实现完整的仿真循环——外层循环控制关联参数组合，内层执行优化算法，并自动记录每个情景下的最优目标值、种植结构特征和风险指标，最终输出多维度对比表格和可视化图表。
- **与问题二结果的结构化比较分析：**需从收益水平、风险暴露、种植结构多样性、轮作执行率等角度，系统说明引入关联效应后种植策略发生偏移的方向与幅度，并解释其经济直觉。北太天元可辅助计算上述比较指标，并绘制平行坐标图或雷达图，

直观呈现两种策略在多维指标上的差异，为结论提供定量支撑。

小结：本问重点考察内生关联系统的建模能力、非线性目标与约束的处理技巧、仿真实验设计与对比分析方法，要求学生能够跳出固定参数的思维定式，在系统互馈视角下重新审视种植策略的稳定性与适应性。北太天元凭借其对自定义复杂函数、大规模循环仿真和进化算法的良好支持，为学生提供了灵活的实验平台，使其能够在多种合理假设下检验策略的敏感性和改进空间，体现科学研究中模型演进与对比验证的核心范式。

综合评述：全题从确定性多周期整数规划出发，逐步递增不确定性维度、风险管控维度和系统关联维度，形成一条完整的农业决策建模进阶路径。学生在解决过程中，须依次掌握混合整数线性规划、场景生成与蒙特卡洛模拟、条件风险价值建模、差分进化与多目标遗传算法、非线性系统仿真等多元方法，并具备将农业管理知识转化为数学约束的抽象能力。北太天元在这一体系中，不仅作为优化求解器和随机模拟引擎，更作为一个开放的计算实验环境，支撑学生从模型构建、算法设计、仿真推演到结果对比的全流程研究工作，其灵活性、可扩展性与高效率恰与竞赛所倡导的创新性与实践性要求高度契合。

3. 优秀论文

202503080123 基于几何模型的《板凳龙》运动状态与路线研究

作者：王心茹 王 森 申嘉旭

摘要：本文针对由 223 节板凳组成的龙队在螺旋路径上的盘入、调头和盘出三个阶段，建立了运动轨迹模型和单目标连续优化模型。

针对问题一，建立板凳龙沿固定螺距运动的轨迹模型。以阿基米德螺线为基础构建极坐标并结合螺线长度积分与路程的关系求解龙头前把手的坐标，再利用位置迭代公式计算剩余板凳前把手的位置。再基于同板凳前后把手沿板凳方向速度相同的性质，建立相邻两把手速度迭代公式，结合龙头前把手的恒定速度，最终可得到各时刻把手的速度。结果显示每秒龙身的速度随节数增加稍有下降。

针对问题二，建立板凳龙碰撞检测模型。基于问题一所得到的各时刻板凳龙的位置和速度结果确定易碰撞角点位置并结合几何方法判定是否发生碰撞，从而确定板凳龙不能继续盘入的位置，进而得出首次碰撞时刻为 412.758221s 并取求得终止时刻各把手的位置和速度。

针对问题三，建立使龙头达到调头空间的最小螺距单目标连续优化模型。以调头空间边

界为约束条件，结合上述问题的模型得到螺距与碰撞终止时刻龙头前把手极径的反比关系，采用步长迭代法调整螺距。最终求得最小螺距为 0.450338m，确保调头空间利用最优化。

针对问题四，建立调头路径模型。首先证明调整圆弧不可以使调头路径更短，随即将调头路线分为四个部分，通过几何关系求解各部分的曲线方程，将龙头的运动路程与四段弧线方程结合求解龙头前把手中心坐标。依据判断函数确定后把手所处于弧线段的位置并利用几何关系构建七类弧段位置、速度迭代公式，从而得出调头前后 100 秒内的位置和速度。

针对问题五，在调头路径模型的基础上，构建龙头最大速度的优化模型。提取速度矩阵中各把手最大速度，通过速度缩放系数推导龙头最大行进速度，最终得出龙头最大盘绕速度为 1.406276m/s。

本文充分贴合板凳龙运动特性，可推广至舞龙舞狮、多机器人编队等类似链式结构的轨迹规划与避障问题，为集体表演活动安全与效果优化提供科学依据。

关键词：板凳龙；阿基米德螺线；碰撞检测模型；单目标连续优化；步长迭代法

点 评：本文选题富有文化特色，将数学建模应用于传统民俗活动《板凳龙》的运动分析中，兼具学术趣味性和工程参考价值。建模方法以几何分析为主线，阿基米德螺线的极坐标建模、位置迭代公式的递推求解、碰撞检测的几何判定，均体现了清晰的物理直觉和扎实的数学功底。五个问题层层递进，从轨迹建模到碰撞检测到优化设计，覆盖了运动分析的完整链条。问题三的螺距优化和问题五的最大速度求解，将模型从描述推向优化，提升了研究的深度。在数值计算上，给出了精确到微秒级的碰撞时刻（412.758s）和毫米级的优化结果（0.450338m），体现了精细的计算态度。建议可进一步考虑龙身柔性对运动的影响，并增加可视化展示以增强结果直观性。

基于几何模型的“板凳龙”运动状态与路线研究

摘 要

本文针对由 223 节板凳组成的龙队在螺旋路径上的盘入、调头和盘出三个阶段，建立了运动轨迹模型和单目标连续优化模型。

针对问题一，建立板凳龙沿固定螺距运动的轨迹模型。以阿基米德螺线为基础构建极坐标并结合螺线长度积分与路程的关系求解龙头前把手的坐标，再利用位置迭代公式计算剩余板凳前把手的位置。再基于同板凳前后把手沿板凳方向速度相同的性质，建立相邻两把手速度迭代公式，结合龙头前把手的恒定速度，最终可得到各时刻把手的速度。结果显示每秒龙身的速度随节数增加稍有下降。

针对问题二，建立板凳龙碰撞检测模型。基于问题一所得到的各时刻板凳龙的位置和速度结果确定易碰撞角点位置并结合几何方法判定是否发生碰撞，从而确定板凳龙不能继续盘入的位置，进而得出首次碰撞时刻为 412.758221s 并取求得终止时刻各把手的位置和速度。

针对问题三，建立使龙头达到调头空间的最小螺距单目标连续优化模型。以调头空间边界为约束条件，结合上述问题的模型得到螺距与碰撞终止时刻龙头前把手极径的反比关系，采用步长迭代法调整螺距。最终求得最小螺距为 0.450338m，确保调头空间利用最优化。

针对问题四，建立调头路径模型。首先证明调整圆弧不可以使调头路径更短，随后将调头路线分为四个部分，通过几何关系求解各部分的曲线方程，将龙头的运动路程与四段弧线方程结合求解龙头前把手中心坐标。依据判断函数确定后把手所处于弧线段的位置并利用几何关系构建七类弧段位置、速度迭代公式，从而得出调头前后 100 秒内的位置和速度。

针对问题五，在调头路径模型的基础上，构建龙头最大速度的优化模型。提取速度矩阵中各把手最大速度，通过速度缩放系数推导龙头最大行进速度，最终得出龙头最大盘绕速度为 1.406276m/s。

本文充分贴合板凳龙运动特性，可推广至舞龙舞狮、多机器人编队等类似链式结构的轨迹规划与避障问题，为集体表演活动安全与效果优化提供科学依据。

关键词： 板凳龙；阿基米德螺线；碰撞检测模型；单目标连续优化模型；步长迭代法

一、问题重述

1.1 问题背景

“板凳龙”是来源于浙江和福建一带地区的传统民俗活动，蕴含着深厚的文化底蕴和悠久的历史积淀。在表演过程时，舞龙队伍将多个板凳首尾相接，沿螺旋轨迹盘绕行进，形成独具特色的队形结构。这种独特的编排不仅增强了表演的视觉观赏性，也有效缩小了所需场地面积，增强表演的效果。板凳龙的表演形式具有鲜明的力学和几何特性。队员们需要在行进中始终保持队形整齐，并精确控制每节板凳的运动轨迹与行进速度，因此科学的路线规划与灵活的速度调节，更是保障表演安全有序、兼具视觉冲击力与艺术表现力的核心环节。

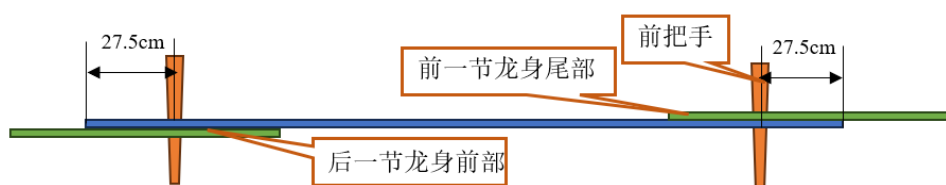


图 1_1 板凳龙连接的示意图

1.2 问题提出

本文以一条由 223 节板凳组成的板凳龙为研究对象，每节板凳宽度均为 30cm，重点探讨其在螺线路径上的运动学行为。板凳龙队伍由一个龙头、221 节龙身和一个龙尾组成，其中龙头板长为 341cm，龙身及龙尾板长均为 220cm，各节板凳均通过特定位置的孔径进行连接，形成整体盘龙结构。下图展示了龙头和龙身的示意图：

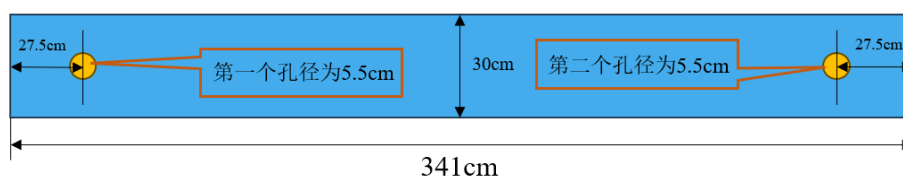


图 1_2 龙头各参数的示意图

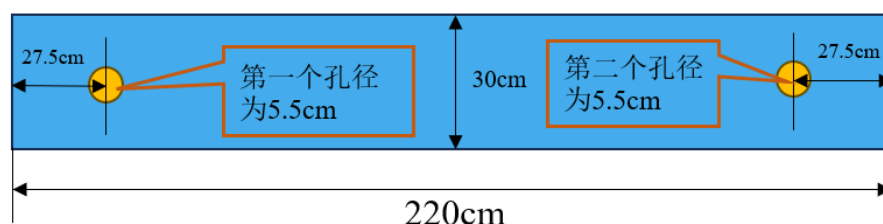


图 1_3 龙身各参数的示意图

基于上述内容，要求建立数学模型解决以下五个问题：

问题一：运动轨迹建模

板凳龙以第 16 圈 A 点出发，沿着固定螺距为 55cm 的等距螺线顺时针盘入，各个连接把手均位于该螺线上，龙头前把手初始速度为 1m/s，需构建合适的运动轨迹模型，求解 0 至 300 秒内，每秒各个板凳的所有把手的位置和速度。

问题二：碰撞终止条件

继续按照问题一的行进螺线，求解板凳之间即将发生碰撞的时刻，也就是确定舞龙队盘绕进入的结束时刻，并给出该时刻舞龙队中板凳的各个把手中心的位置和速度。

问题三：调头空间优化

当舞龙队到达盘入结束时刻后，需要调头切换为逆时针的盘出运动状态以免碰撞。如果调头空间是以螺线中心为圆心、直径为 9m 的圆形区域，建立最小螺距模型，使得龙头前把手能够沿着相应的螺线盘入到调头空间的边界。

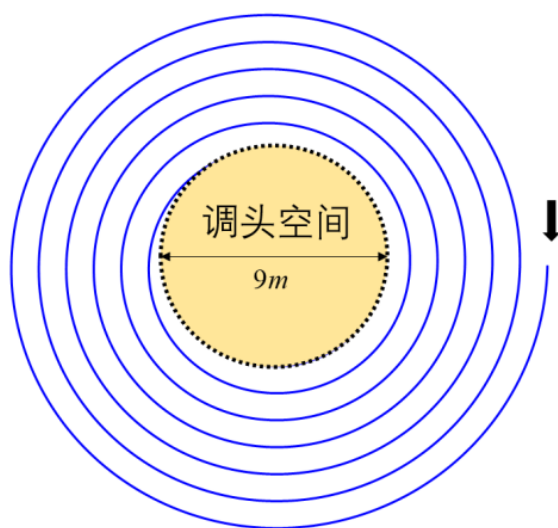


图 1_3 调头空间的示意图

问题四：调头路径规划

保持龙头前把手速度不变始终为 1m/s，“板凳龙”沿螺距为 1.7m 的螺线盘入和盘出，调头空间为直径 9m 的圆形区域，调头路径为由两段圆弧相切连接而成的 S 形曲线。确定是否可以调整圆弧，仍保持各部分相切，使得调头曲线变短。接着，以调头开始时间为 0 时刻，给出从 -100s 开始到 100s 为止，每秒各个板凳的所有把手的位置和速度。

问题五：速度上限分析

沿用问题四的行进路径设定，将各个把手的速度限制在 2m/s 之内，反推龙头最大行进速度。

二、问题分析

2.1 问题一的分析

对于问题一核心是构建运动轨迹模型，以确定板凳龙沿螺旋路径运动时各个把手在各时刻的位置和速度[4]。首先，我们已知盘旋进入螺线的螺距和起点，基于阿基米德螺线，先确定坐标系与关键参数进行分析，利用路程与螺线长度关系求极角以确定龙头位置，再依据龙头位置，结合极径递推关系与余弦定理，通过迭代确定所有龙身把手位置，最后利用同一板凳前后把手沿板凳方向速度相同的特性，计算龙身首尾端斜率，建立速度分量关系，推导角度并代入，通过迭代公式算出各段速度，为板凳龙 300 秒内运动轨迹提供求解基础。

2.2 问题二的分析

问题二分析以确定板凳龙即将碰撞的盘绕结束时刻为核心目标，整体围绕着问题一已建立的螺线模型与把手位置计算基础展开。首先，从龙身盘旋的实际运动特性出发，明确最易率先发生碰撞的关键部位，再基于问题一得出的把手中心点位置，结合板凳结构参数定义相关参考线，通过几何变换与方程联立的思路求出易碰撞部位的坐标；随后，采用点到直线距离的判定逻辑，设定碰撞阈值，通过计算易碰撞部位到对应参考线的距离与阈值的关系，判断是否发生碰撞，首次满足碰撞条件的时刻即为盘绕结束时刻，最后基于该时刻可进一步推导所有板凳把手的位置与速度。

2.3 问题三的分析

问题三是以问题一、二确立的螺距与盘入终止时刻、把手位置关联为基础，核心是求龙头前把手达调头空间边界的最小螺距。只需要通过调整螺距，使得在该螺距对应的终止时刻时，龙头前把手中心的位置恰好位于调头空间的边界上。求解时先设螺距初始值与步长，利用问题二模型判断前把手极径刚达 4.5m 时是否碰撞，无碰撞则按步长减小区间，有碰撞则以上次无碰撞螺距为新初始值，循环至区间达标后缩小步长迭代，最终精准得最小螺距。

2.4 问题四的分析

问题四是在问题一模型基础上的扩展。按照题目提供的调头路径条件，通过结合几何关系与运动特性判断是否可以调整圆弧来缩短路径。随后由于路径较为复杂，将其视作由四段曲线构成，再以龙头前把手恒定速度为基础，结合调头路径四段弧线方程与路程积分，延续

问题一的建模方式，先求出龙头前把手的每秒位置变化函数，再通过递推关系求解其余把手的位置函数。最后参考问题一速度求解逻辑，结合弧线段斜率与螺线速度迭代公式，得到各把手各时刻速度，以此完成 -100s 至 100s 每秒板凳的运动轨迹情况。

2.5 问题五的分析

问题五的目的是建立一个优化模型，求解龙头的最大盘绕速度。首先基于问题四已求得的 $[-100, 100]$ 秒内各把手中间点速度构建速度矩阵，计算全过程中各把手的最大速度。其次需要对龙头设定速度缩放系数，最后依据速度的约束条件即可求出龙头的最大盘旋速度。

三、模型假设

1. 假设板凳之间通过把手连接是理想的，在运动过程中不会因弯曲而产生额外的阻力或能量损耗。
2. 将每一节板凳视为刚体，其长度在运动过程中保持恒定，不会发生形变。
3. 忽略板凳的厚度，只关注板凳的长度和宽度对模型的影响，降低了模型的几何复杂度，同时保留了影响路径规划的核心数据特征。
4. 假设在整个运动过程中不受风力、地面摩擦或其他外力的影响，其运动状态仅由龙头的牵引作用决定。

四、符号说明

符号	说明
x_0	龙头在 t 时刻所处的 x 轴坐标
y_0	龙头在 t 时刻所处的 y 轴坐标
ρ_0	龙头在 t 时刻的极径（点到极点的距离）
θ_0	龙头在 t 时刻的极角（从极轴按逆时针旋转所得）
x_i	龙身、龙尾在第 i 节 t 时刻所处的 x 轴坐标（ $i = 1, 2, 3, \dots$ ）
y_i	龙身、龙尾在第 i 节 t 时刻所处的 y 轴坐标（ $i = 1, 2, 3, \dots$ ）
ρ_i	龙身、龙尾在第 i 节 t 时刻的极径（点到极点的距离）（ $i = 1, 2, 3, \dots$ ）
θ_i	龙身、龙尾在第 i 节 t 时刻的极角（从极轴按逆时针旋转所得）（ $i = 1, 2, 3, \dots$ ）
l	龙身、龙尾的板长
d	等距螺线的螺距
φ_i	第 i 节龙身与 x 轴的夹角（ $i = 1, 2, 3, \dots$ ）

符号	说明
$\varphi_{i1}, \varphi_{i2}$	第 i 节龙身首端、尾端与 x 轴的夹角 ($i1$ 代表龙身首端, $i2$ 代表龙身尾端)
α	φ_{i1} 与 φ_i 的夹角之差 ($i = 1, 2, 3, \dots$)
β	φ_{i2} 与 φ_i 的夹角之差 ($i = 1, 2, 3, \dots$)
v	龙头的速度
v_i	龙身、龙尾在第 i 节 t 时刻的速度
d_{i1}	板凳前后把手到顶部的距离
d_{i2}	板凳前后把手到侧边的距离
l_0	每个板凳的中线

五、模型建立与求解

5.1 问题一模型建立与求解

5.1.1 板凳龙沿固定螺距运动的轨迹模型建立

以极点 O 为原点, 在极坐标平面内构建直角坐标系, 使 Ox 轴与极轴重合。接着, 给出了阿基米德螺线[1]的极坐标方程:

$$\rho = \zeta(\theta) = \zeta_0 + k\theta \left(\frac{-\zeta_0}{k} \leq \theta \leq \infty \right)$$

该方程[1]描述了极径 ρ 随极角 θ 变化的规律。

Step1: 龙头前把手中心位置的确定

首先, 考虑在 $t_0 \in [0, 300]$ 时刻龙头所在位置 (x_0, y_0) , 将 (x_0, y_0) 转换至极坐标系, 满足:

$$\begin{cases} x_0 = \rho(\theta_0) \cos \theta_0, \\ y_0 = \rho(\theta_0) \sin \theta_0. \end{cases} (1)$$

由阿基米德螺线的性质, 结合题意中螺距 $d = 0.55m$, 可得极径与极角的关系为:

$$\rho(\theta) = \frac{d\theta}{2\pi}.$$

为求解 θ_0 , 根据 t_0 时刻经过的路程 vt_0 与螺线长度积分公式相等的关系, 可得:

$$vt_0 = \int_{\theta_0}^{32\pi} \sqrt{\rho'(\theta)^2 + \rho(\theta)^2} d\theta. (2)$$

由题可知, 初始速度 $v = 1 m/s$, 则

$$t_0 = \int_{\theta_0}^{32\pi} \sqrt{\rho'(\theta)^2 + \rho(\theta)^2} d\theta, \rho(\theta) = \frac{d\theta}{2\pi}.$$

将 $\rho(\theta) = \frac{d\theta}{2\pi}$ 代入上式，由此式可以算出 t_0 时刻所对应的 θ_0 值，将 θ_0 值代入式（1）中，即可得到龙头在 t_0 时刻所处位置 (x_0, y_0) 。

Step2: 龙身各把手中心位置的确定

在 t_0 时刻，根据龙头位置可确定出第一个龙身的位置，设定为 (x_1, y_1) ，其在极坐标系中表示为 (ρ_1, θ_1) 。利用上述过程，即可求出第二个龙身的位置，再由第二个位置重复类似步骤，可依次确定后续龙身把手的位置，直至龙尾位置结束。由阿基米德螺线的方程推导，相邻龙身把手的极径满足：

$$\rho_2 = \rho_1 + \frac{d}{2\pi}(\theta_2 - \theta_1). (3)$$

通过该式可递推得到各龙身把手在极坐标系中的位置关系。如下图所示：

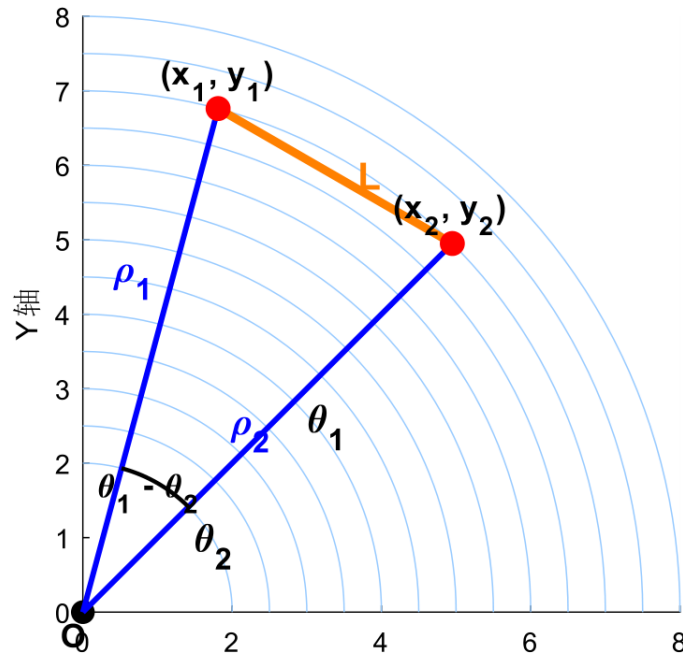


图 5_1 前后把手相对位置示意图

结合图 5_1 所示的前后把手相对位置示意图，并基于余弦定理可得：

$$\cos(\theta_2 - \theta_1) = \frac{\rho_1^2 + \rho_2^2 - l^2}{2\rho_1\rho_2}. (4)$$

其中木板长度 $l = 2.2m$ 。联立公式（3）与（4），可求解得到 (ρ_2, θ_2) ，再通过公式（1）

进一步可以确定 (x_2, y_2) 。

由于在龙身运动过程中木板长度始终保持不变，因此可通过迭代公式持续求解，直至龙尾位置。迭代公式如下：

$$\begin{cases} \rho_{i+1} = \rho_i + \frac{d}{2\pi}(\theta_{i+1} - \theta_i), \\ \rho_i^2 + \rho_{i+1}^2 - l^2 = 2\rho_i\rho_{i+1}\cos(\theta_{i+1} - \theta_i), \end{cases} \quad (1 \leq i \leq 222).$$

其中，螺距为 $d = 0.55m$ ，木板长度 $l = 2.2m$ 。

同时，将所得的 (ρ_i, θ_i) 代入极坐标与直角坐标的转换公式：

$$\begin{cases} x_i = \rho(\theta_i)\cos\theta_i, \\ y_i = \rho(\theta_i)\sin\theta_i, \end{cases} \quad \rho(\theta_i) = \frac{d}{2\pi}\theta_i (d = 0.55m),$$

即可计算得到各把手的直角坐标 (x_i, y_i) 。

Step3: 各位置速度迭代公式

由于同一个板凳有前后两个把手，故其前后把手沿板凳方向的速度是相同的。因此利用这一特性构建以下的速度迭代公式，建立过程如下：

假设第 1 节龙身首端的坐标为 (x_1, y_1) ，在极坐标系中为 (ρ_1, θ_1) ，现上述均为已知量，可由公式 (2) 得龙身尾端坐标为 (x_2, y_2) ，在极坐标系中为 (ρ_2, θ_2) ，如下图 5_2 所示展示出各龙身把手在极坐标系中的速度关系示意图：

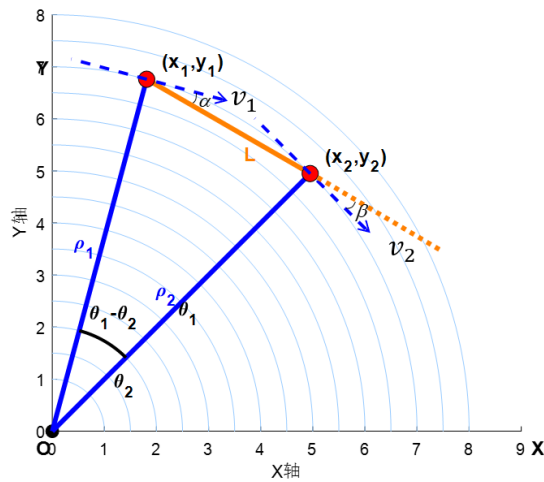


图 5_2 前后把手相对速度示意图

在龙身首端斜率为：

$$\tan\varphi_{i1} = \frac{dy}{dx} = \frac{\sin\theta_{i1} + \theta_{i1}\cos\theta_{i1}}{\cos\theta_{i1} - \theta_{i1}\sin\theta_{i1}}.$$

在龙身尾端斜率为：

$$\tan\varphi_{i2} = \frac{dy}{dx} = \frac{\sin\theta_{i2} + \theta_{i2}\cos\theta_{i2}}{\cos\theta_{i2} - \theta_{i2}\sin\theta_{i2}}.$$

由于前后把手都处于同一板凳上，故龙身首端在板凳方向的分向量与龙身尾端在板凳方向上的分向量相同，即满足

$$v_1\cos\alpha = v_2\cos\beta,$$

其中 α 和 β 均为未知量。

由板凳方向的斜率为：

$$\tan\varphi_i = \frac{y_2 - y_1}{x_2 - y_1},$$

可得：

$$\begin{aligned}\tan\alpha &= \tan(\varphi_i - \varphi_{i1}) = \frac{\tan\varphi_i - \tan\varphi_{i1}}{1 + \tan\varphi_i \tan\varphi_{i1}}, \\ \tan\beta &= \tan(\varphi_{i2} - \varphi_i) = \frac{\tan\varphi_{i2} - \tan\varphi_i}{1 + \tan\varphi_i \tan\varphi_{i2}}.\end{aligned}$$

即进一步有：

$$\begin{aligned}\alpha &= \arctan \frac{\tan\varphi_i - \tan\varphi_{i1}}{1 + \tan\varphi_i \tan\varphi_{i1}}, \\ \beta &= \frac{\tan\varphi_{i2} - \tan\varphi_i}{1 + \tan\varphi_i \tan\varphi_{i2}}.\end{aligned}$$

将 α 和 β 代入上式 $v_1\cos\alpha = v_2\cos\beta$ 中即可由 v_1 得 v_2 。又由于各板凳是完全相同的，故建立迭代公式：

$$\begin{cases} \alpha_i = \arctan \frac{\tan\varphi_i - \tan\varphi_{i1}}{1 + \tan\varphi_i \tan\varphi_{i1}}, \\ \beta_i = \frac{\tan\varphi_{i2} - \tan\varphi_i}{1 + \tan\varphi_i \tan\varphi_{i2}}, \end{cases}$$

$$v_i\cos\alpha = v_{i+1}\cos\beta.$$

通过该迭代公式，能够依次计算出龙身各段对应速度，为后续龙身运动状态等相关分析提供基础。

5.1.2 模型求解及结果分析

将数据代入 5.1.1 构建的板凳龙沿固定螺距运动的轨迹模型中，通过北太天元运行得出结果如下所示。

表 5_1 板凳龙特殊节点位置结果

	0s	60s	120s	180s	240s	300s
龙头 x(m)	8.800000	5.799209	-4.084887	-2.963609	2.594494	4.420274
龙头 y(m)	0.000000	-5.771092	-6.304479	6.094780	-5.356743	2.320429
第 1 节龙 身 x(m)	8.363824	7.456758	-1.445473	-5.237118	4.821221	2.459489
第 1 节龙 身 y(m)	2.826544	-3.440399	-7.405883	4.359627	-3.561949	4.402476
第 51 节 龙身 x(m)	-9.518732	-8.686317	-5.543150	2.890455	5.980011	-6.301346
第 51 节 龙身 y(m)	1.341137	2.540108	6.377946	7.249289	-3.827758	0.465829
第 101 节 龙身 x(m)	2.913983	5.687116	5.361939	1.898794	-4.917371	-6.237722
第 101 节 龙身 y(m)	-9.918311	-8.001384	-7.557638	-8.471614	-6.379874	3.936008
第 151 节 龙身 x(m)	10.861726	6.682311	2.388757	1.005154	2.965378	7.040740
第 151 节 龙身 y(m)	1.828753	8.134544	9.727411	9.424751	8.399721	4.393013
第 201 节 龙身 x(m)	4.555102	-6.619664	-10.627211	-9.287720	-7.457151	-7.458662
第 201 节 龙身 y(m)	10.725118	9.025570	1.359847	-4.246673	-6.180726	-5.263384
龙尾 (后) x(m)	-5.305444	7.364557	10.974348	7.383896	3.241051	1.785033
龙尾 (后)	-10.676584	-8.797992	0.843473	7.492370	9.469336	9.301164

	0s	60s	120s	180s	240s	300s
y (m)						

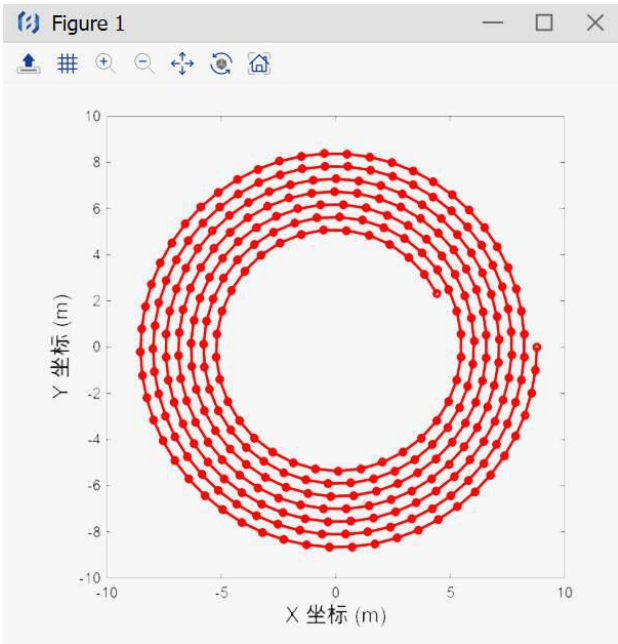


图 5_3 龙头 300 秒内位置轨迹图

根据所得到的板凳龙在 0 到 300 秒内的位置坐标，将 $t = 300$ 秒时板凳龙位置轨迹图，如下图 5_4 所示。

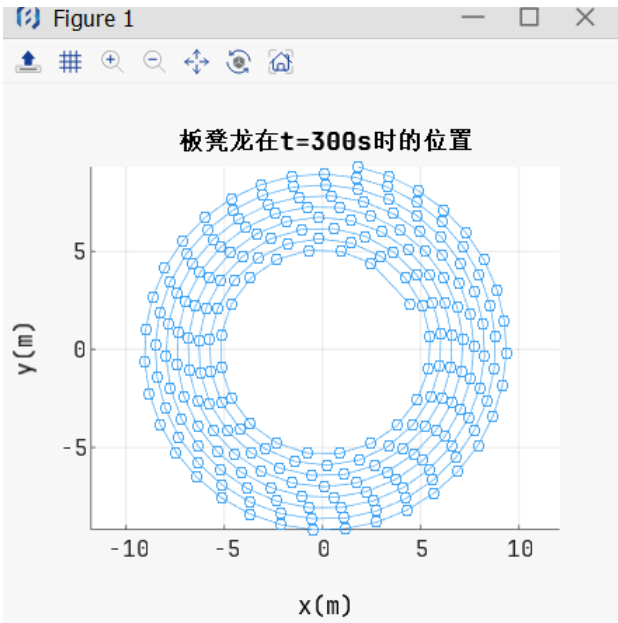


图 5_4 $t=300$ 秒时板凳龙位置轨迹图

利用速度迭代公式得出结果，如下表 5_2 所示：

表 5_2 板凳龙特殊节点速度结果

	0s	60s	120s	180s	240s	300s
龙头(m/s)	1	1	1	1	1	1
第 1 节龙身 (m/s)	0.999971	0.999961	0.999945	0.999917	0.99985	0.999710
第 51 节龙身 (m/s)	0.999742	0.999662	0.999538	0.999331	0.998941	0.998065
第 101 节龙 身(m/s)	0.999575	0.999453	0.999269	0.99897	0.998435	0.997302
第 151 节龙 身(m/s)	0.999448	0.999298	0.999078	0.998728	0.998114	0.996861
第 201 节龙 身(m/s)	0.999348	0.999180	0.998935	0.998551	0.997893	0.996574
龙尾（后） (m/s)	0.999311	0.999137	0.998883	0.998489	0.997816	0.996478

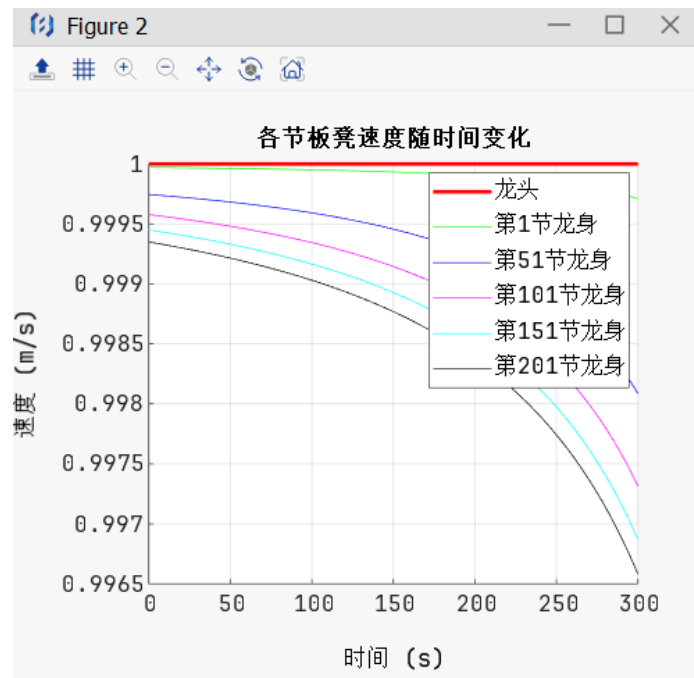


图 5_5 板凳龙特殊节点的速度变化图

5.2 问题二模型建立与求解

5.2.1 板凳龙碰撞检测模型建立

在龙身不断向螺线内部进行盘旋过程中，最先发送碰撞的一定是龙头最外侧的两个顶点或者是第一节龙身外侧的两个顶点，如下图 5_6 所示， A_1 和 A_2 是龙头的两个最易碰撞的点， A_3 和 A_4 是第一节龙身的两个顶点。

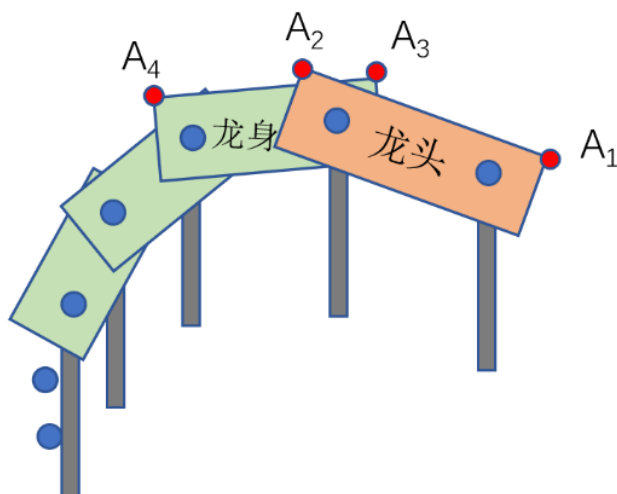


图 5_6 板凳龙最易碰撞的节点示意图

由问题一我们可以确定前把手中心点和后把手中心点位置，由问题一中所计算得到的点的位置，求出这四个顶点的坐标表达式，再由点到直线间的距离来判断是否发生碰撞。

Step1: 确定龙头与第一节龙身四个角点位置的坐标

对龙头来说，设 l_0 为龙头中线， d_{11} 为龙头前后把手到板凳顶部的距离， d_{12} 为龙头前后把手到板凳侧边的距离。 α_2 为前把手到前把手角点 A_1 的这条直线， α_3 为龙头中线 l_0 向左平移 d_{12} ， α_4 为后把手到后把手角点 A_2 的这条直线，以此类推对于任意一个龙身也是如此的操作，如下图 5_7 所示。

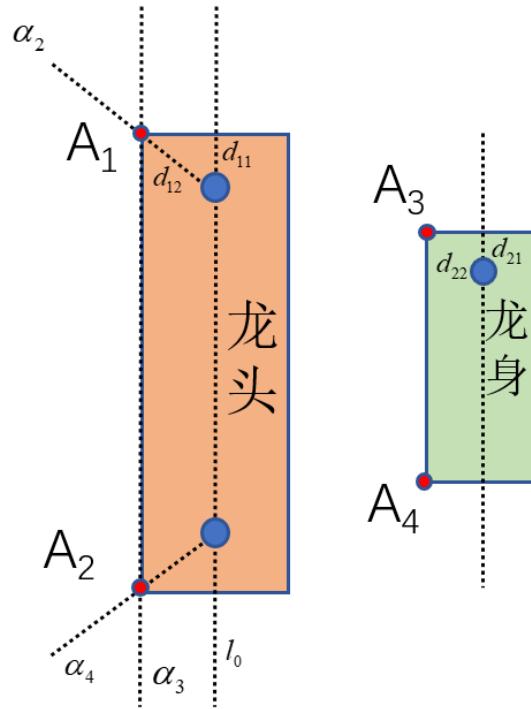


图 5_7 板凳参数示意图

设定 l_0 直线与 α_2 直线的夹角为 γ ，由对称性可知 l_0 直线与 α_3 直线的夹角也为 γ 。如果要求出四个角点的位置，可以求出直线 $\alpha_2, \alpha_3, \alpha_4$ 的方程。可以将直线 α_2 与直线 α_3 方程联立求解得出 A_1 点坐标，将直线 α_3 与直线 α_4 方程联立求解得出 A_2 点坐标。

已知 l_0 直线的斜率为：

$$k_{l_0} = \frac{y_1 - y_0}{x_1 - x_0}, y - y_0 = k_{l_0}(x - x_0).$$

直线 α_2 可由 l_0 直线绕着点 (x_0, y_0) 逆时针旋转 γ 角后可得，因此可由旋转角公式推得 α_2 直线的表达式为：

$$\begin{cases} \tan\gamma = \frac{d_2}{d_1}, \\ k_{\alpha_2} = \frac{\tan\gamma + k_{l_0}}{k_{l_0}\tan\gamma - 1}, \end{cases} \Rightarrow y - y_0 = k_{\alpha_2}(x - x_0).$$

直线 α_4 可由 l_0 直线绕着点 (x_1, y_1) 顺时针旋转 γ 角后可得，因此可由旋转角公式推得 α_4 直线的表达式为：

$$\begin{cases} \tan\gamma = \frac{d_2}{d_1}, \\ k_{\alpha_4} = \frac{k_{l_0} - \tan\gamma}{k_{l_0} \tan\gamma + 1}, \end{cases} \Rightarrow y - y_1 = k_{\alpha_4}(x - x_1).$$

直线 α_3 可由 l_0 直线向上平移 d_{12} 距离可得出，因此得到 α_3 直线的表达式为：

$$y = k_{l_0}x + b, \frac{b - y_1 + k_1x_1}{\sqrt{1 + k_1^2}} = d_2,$$

并且为了确保向上平移，要加上限制条件

$$b > y_1 - k_1x_2.$$

最后联立将直线 α_2 与直线 α_3 方程联立求解得出 A_1 点坐标：

$$(x_{A_1}, y_{A_1}) = \left(\frac{y_0 - k_{\alpha_2}x_0 - b}{k_{l_0} - k_{\alpha_2}}, \frac{k_{l_0}y_0 - k_{l_0}k_{\alpha_2}x_0 - k_{\alpha_2}b}{k_{l_0} - k_{\alpha_2}} \right).$$

将直线 α_3 与直线 α_4 方程联立求解得出 A_2 点坐标：

$$(x_{A_2}, y_{A_2}) = \left(\frac{y_1 - k_{\alpha_4}x_1 - b}{k_{l_0} - k_{\alpha_4}}, \frac{k_{l_0}y_1 - k_{l_0}k_{\alpha_4}x_1 - k_{\alpha_4}b}{k_{l_0} - k_{\alpha_4}} \right).$$

因此龙头 A_1 和 A_2 点坐标即可求出。第一节龙身由同样方法即可求出 A_3 和 A_4 坐标。

Step2: 判断板凳龙四个角点是否碰撞

用上述四点坐标利用点到直线的距离公式，得到：

$$d_{ij} = \frac{\left| \frac{y_i - y_{i+1}}{x_i - x_{i+1}}(x_{A_j} - x_i) - y_{A_j} + y_i \right|}{\sqrt{\left(\frac{y_i - y_{i+1}}{x_i - x_{i+1}} \right)^2 + 1}}, (i = 1, 2, \dots, 222; j = 1, 2, 3, 4).$$

- 如果 $d_{ij} > d_{22}$ ，则没有发生碰撞；
- 如果 $d_{ij} < d_{22}$ ，则发生了碰撞，由此可以得出板凳龙停止时刻。

5.2.2 模型求解及结果分析

将数据代入 5.2.1 模型中，求解得出板凳龙在 412.758221 秒，发送碰撞的点位置速度结果如下图 5_8 所示：

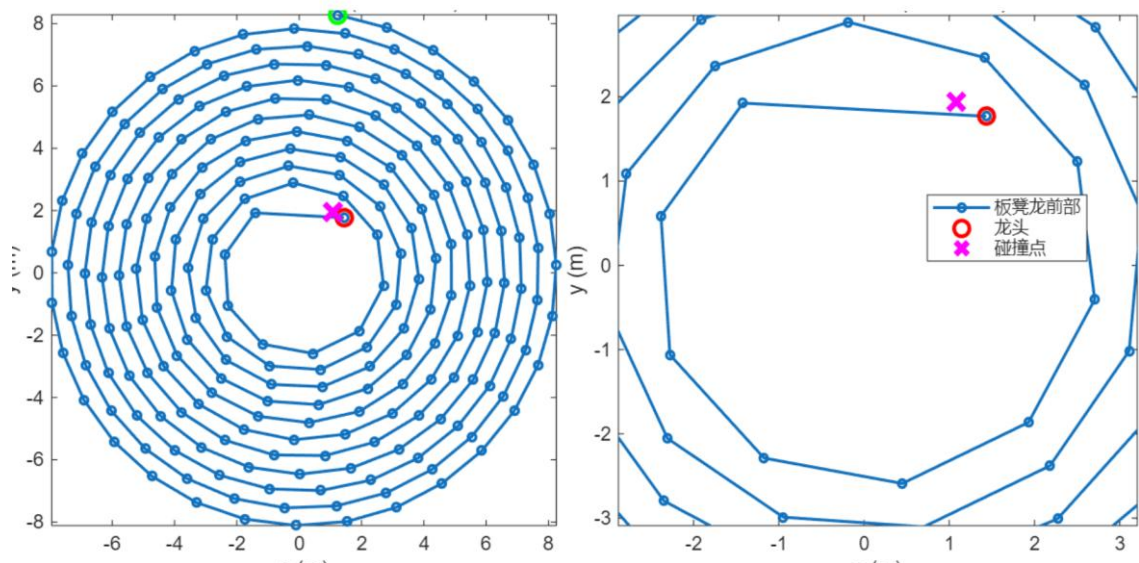


图 5_8 (a)碰撞时刻板凳龙位置($t=412.76s$)和(b)碰撞区域放大图

下面将板凳龙特殊节点碰撞时刻的位置及其速度记录在下表中：

表 5_3 板凳龙特殊节点碰撞位置结果

终止时刻为 412.758221s			
龙头 x(m)	1.434880782	龙身 y(m)	1.769098497
第 1 节龙身 x(m)	-1.420827736	第 1 节龙身 y(m)	1.925715403
第 51 节龙身 x(m)	1.543187421	第 51 节龙身 y(m)	4.23447727
第 101 节龙身 x(m)	-0.811003699	第 101 节龙身 y(m)	-5.844428963
第 151 节龙身 x(m)	0.693591928	第 151 节龙身 y(m)	-6.986818824
第 201 节龙身 x(m)	-7.928040024	第 201 节龙身 y(m)	-0.956305122
龙尾(后)x(m)	1.230080338	龙尾(后)y(m)	8.28376481

表 5_4 板凳龙特殊节点碰撞速度结果

终止时刻为 412.758221s	
龙头(m/s)	1.00000000 0
第 1 节龙身(m/s)	0.99135752 5
第 51 节龙身(m/s)	0.97649934

终止时刻为 412.758221s	
	4
第 101 节龙身 (m/s)	0.97418379 1
第 151 节龙身 (m/s)	0.97323987 3
第 201 节龙身 (m/s)	0.97272701 7
龙尾(后) (m/s)	0.97256875 6

5.3 问题三模型建立与求解

5.3.1 龙头达到调头空间的最小螺距单目标连续优化模型建立

结合问题一和问题二所构建的模型，能够确定：在其他条件不变的情况下，每个时刻板凳龙队各把手所处的位置以及板凳龙队盘入的终止时刻，均由螺距大小决定。实际上问题三是求解龙头达到调头空间的最小螺距优化问题[3]。

从问题一和问题二的结果可知，当螺距为 55cm 时，板凳龙停止时刻的前把手的极径为

$$\rho(\theta_0) = 2.288734m, \rho(\theta_0) < 4.5m,$$

这表明此螺距下可在该调头空间内完成调头。

进一步分析可得，螺距与板凳龙停止时刻前把手的极径呈反比关系，随着螺距的减少，板凳龙到达终止时间时板凳的前把手的极径在不断增大，那么由此可以把最小螺距的上限设为 55cm 并构建如下判定流程：

Step1: 设定螺距初始值为 $d = 0.55m$ ，每次步长减少量为 $\Delta d = 0.01m$ 。

Step2: 利用问题二中的模型，判断龙头前把手的极径刚到 4.5m 时是否发生碰撞。

Step3: 不断重复 Step2，如果没有发生碰撞，则按照设定步长减小区间；如果发生碰撞，则将上次没有发生碰撞的螺距作为初始值，由此循环可以得到一个长度为 0.01m 的区间，该区间满足前一螺距未发生碰撞，后一螺距发生碰撞的条件。随后，再将步长调整为原来的 $\frac{1}{10}$ ，继续迭代，直至步长循环至 10^{-6} 时结束。最终，能够得到一个较为精确的最小螺距值。

综合来看，在本题的求解过程中，先是给出螺距的上限，接着通过十分法持续调整螺距上限，从而确保在龙头于调头空间内恰好不发生碰撞时，得到最小螺距值。

5.3.2 模型结果及分析

根据上述的流程，得到了以不同的步长进行循环时所对应的最小螺距：

表 5_5 不同步长对应的最小螺距

步长 (m)	螺距 (m)
0.01	0.46
0.001	0.451
0.0001	0.4504
0.00001	0.45034
0.000001	0.450338

根据上面所展示的结果，我们根据题目中的要求，保留了六位小数来保证结果的精确性，得到了盘入调头空间时的最小螺距为 0.450338m。

5.4 问题四模型建立与求解

5.4.1 判断调整圆弧是否可以使调头路径更短

在问题四中，要先来证明是否可以通过调整圆弧来使调头路径更短[2]。其中圆弧的大小取决于该弧长的半径大小，故我们假设出大小圆弧的半径比为 k ，则我们可构建出弧长关于弧长半径的表达式，并由几何关系推导得到弧长半径的表达式，将这两个式子联立可求解出弧长不含弧长半径的表达式。最终，可以表明圆弧的总长度与半径是没有关系的，即调整圆弧无法使调头路径更短。

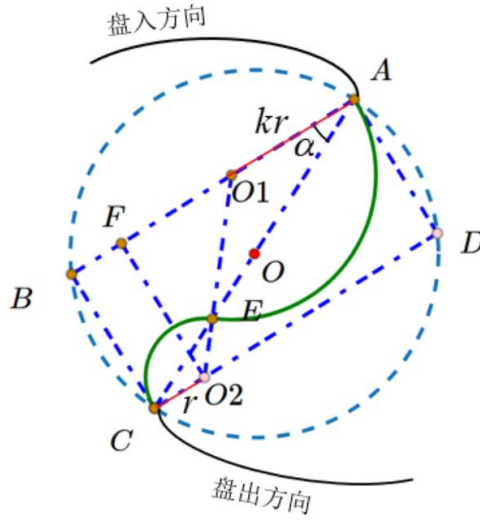


图 5_9 调头路径示意图

证明过程如下：

设角 α 为盘入螺线在切入点处的切线的垂线与调头空间的直径 AC 所组成的夹角，设该调头空间的直径 $AC = D$ ，则 $AB = D\cos\alpha$ ， $BC = D\sin\alpha$ ，可设大小圆弧的半径比为 k ，小圆弧半径为 r ，则大圆弧半径为 kr 。

则总弧长为：

$$s = (k + 1)r(\pi - 2\alpha).$$

再过点 O_2 作 AB 边的垂线，垂足为 F ，则在直角三角形 O_1FO_2 中，

$$O_1F = AB - AO_1 - BF = D\cos\alpha - (k + 1)r,$$

$$O_2F = BC = D\sin\alpha, O_1O_2 = (k + 1)r.$$

由勾股定理可得：

$$[D\cos\alpha - (k + 1)r]^2 + D^2\sin^2\alpha = [(k + 1)r]^2.$$

对上述等式展开并化简求解得：

$$r = \frac{D}{2(k + 1)\cos\alpha}.$$

将 r 带入弧长公式 $s = (k + 1)r(\pi - 2\alpha)$ 可得：

$$s = \frac{D(\pi - 2\alpha)}{2\cos\alpha}.$$

从最终的弧长表达式可以看出 s 与大小圆弧的半径比 k 无关，这就证明了调整圆弧无

法使调头路径更短。

5.4.2 求解板凳龙各时刻的位置与速度

在明确调整圆弧无法缩短调头路径后，我们需进一步求解调头过程中各个时刻整个舞龙队的位置和速度，在本文中具体按以下步骤进行：

Step1: 龙头前把手中心位置求解

已知龙头速度恒定为 1m/s，我们可将龙头的运动路程与调头路径的四段弧线方程相结合，再通过路程公式与积分联立的方式[6]，来求解调头过程中各个时刻龙头前把手中心的位置，关键在于将龙头的运动路程需与四段弧线方程结合。

Step2: 判断龙身其他把手位置并进行迭代求解

根据龙头前把手中心的位置点并结合把手间固定几何关系构建判断函数。通过对比前后把手中心的相对位置，确定后把手所在弧线段。再代入对应弧线段的位置迭代公式，即可算出任意把手在任意弧线上的位置坐标。

Step3: 板凳龙各把手的速度求解

根据 Step2 中所得到的位置，结合问题一中求速度的方式，建立起前后把手在盘入螺线和盘出螺线上的速度迭代公式，在弧线上的速度可结合前后把手中心所在直线的斜率建立速度迭代公式，从而可以得到各个把手处的速度。

(1) 模型构建前的准备工作

在给出上述步骤具体实施的流程前，要先进行一些必要的准备工作：

假设点 A 为龙头恰好不发生碰撞时进入调头空间的位置，该位置可以通过问题三中构建的模型来进行确定，设点 A 的直角坐标 (x_A, y_A) ，极坐标为 (ρ_A, θ_A) ，在点 A 处的切线斜率可表示为：

$$k_{A\text{切}} = \frac{dy}{dx} = \frac{\cos\theta - \theta\sin\theta}{\sin\theta + \theta\cos\theta}.$$

已知切入点 A 坐标和大小圆弧半径比为 2 以及调头空间直径为 9m，故我们可以算得小圆弧的半径和大圆弧的半径，这里将小圆弧的半径设为 r (r 已知)，那么大圆弧半径为 $2r$ 。因此大圆弧的圆心 O_1 坐标通过几何关系推导得出：

$$\begin{cases} x_{o_1} = x_A + 2r\cos\theta', \\ y_{o_1} = y_A + 2r\sin\theta', \\ \theta' = \arctan\left(\frac{-1}{k_{A\text{切}}}\right). \end{cases}$$

进而给出大圆弧的曲线方程为：

$$(x - x_{o_1})^2 + (y - y_{o_1})^2 = (2r)^2.$$

同理可给出小圆弧的圆心 o_2 的坐标：

$$\begin{cases} x_{o_2} = x_{o_1} + 3r\cos\alpha, \\ y_{o_2} = y_{o_1} + 3r\sin\alpha. \end{cases}$$

可给出小圆弧的曲线方程为：

$$(x - x_{o_2})^2 + (y - y_{o_2})^2 = r^2.$$

在上述的表达式中，仅有 α 为未知量，而当小圆弧与盘出螺线相切时，切点的极坐标需要同时满足盘出螺线方程和小圆弧的方程，并且盘出螺线方程与盘入螺线方程相同但方向相反，因此可将小圆弧方程与盘出螺线方程相联立求解，整理得：

$$\left[\frac{-1.7}{2\pi} \theta \cos\theta - (x_{o_1} + 3r\cos\alpha) \right]^2 + \left[\frac{-1.7}{2\pi} \theta \sin\theta - (y_{o_1} + 3r\sin\alpha) \right]^2 = r^2.$$

在该点时的 θ 值， r 值， x_{o_1}, y_{o_1} 值均为已知量，分析上式可知上式为关于 α 的隐式关系式，所以可求得 α 的值。经过上述分析，大小圆弧的圆心坐标和方程、 α 值、 r 值、点 C 和点 E 坐标均为已知量。

(2) 龙头前把手中心位置求解：分阶段求解（共四种运动阶段）

接下来，我们给出 Step1 具体实施的步骤。结合调头路径的四段弧线（原盘入螺线、大圆弧、小圆弧、盘出螺线），以龙头速度恒定为 1m/s 为核心条件，分阶段求解不同时刻龙头前把手中心的坐标，具体如下：

第一段：原盘入螺线段（进入调头空间前）

首先，根据龙头速度恒定特性，某时间段内走过的路程与时间呈线性关系。将所走过的路程和积分相联立即：

$$vt = \frac{1.7}{2\pi} \left[\frac{1}{2} \theta \sqrt{1 + \theta^2} + \frac{1}{2} \ln(\theta + \sqrt{1 + \theta^2}) \right]_{\theta(t)}^{32\pi}.$$

求解上述方程可得到对应时刻的极角，可解出每个时间段 t 所对应的 $\theta(t)$ ，再代入盘入螺线的极坐标方程组中：

$$\begin{cases} x(t) = \frac{1.7}{2\pi} \theta(t) \cos(\theta(t)), \\ y(t) = \frac{1.7}{2\pi} \theta(t) \sin(\theta(t)). \end{cases}$$

最后通过极坐标与直角坐标的转换，得到该阶段龙头前把手中心的直角坐标。

第二段：大圆弧上（沿顺时针方向从点 A 运动至点 E）

结合题目中给出的条件可知，在该题中将龙头到达切入点 A 时的时间设为 $t = 0s$ ，因此龙头在过 A 点后所走的弧长为 $s_1 = vt$ ，所对应的圆心角度数为 $\varphi_1 = \frac{vt}{2r}$ ，现已知点 A 坐标和圆心点 O_1 坐标，结合顺时针旋转特性，得出在大圆弧上的某一时刻 t 的龙头前把手中心坐标可以表示为：

$$\begin{cases} x(t) = x_{O_1} + 2r \cos \left[\arctan \left(\frac{y_A - y_{O_1}}{x_A - y_{O_1}} \right) - \frac{vt}{2r} \right], \\ y(t) = y_{O_1} + 2r \sin \left[\arctan \left(\frac{y_A - y_{O_1}}{x_A - y_{O_1}} \right) - \frac{vt}{2r} \right]. \end{cases}$$

第三段：小圆弧上（沿逆时针方向从切点 E 行至终点 C）

由第二段的方程，可以确定出从点 A 走至点 E 所花费的总时间 t_1 ，同第二段的步骤，可以得出龙头在过 E 点后所走的弧长为 $s_2 = v(t - t_1)$ ，所对应的圆心角度数为 $\varphi_2 = \frac{v(t-t_1)}{r}$ ，现已知点 A 坐标和圆心点 O_2 坐标，结合逆时针旋转特性，得出在小圆弧上的某一时刻 t 的龙头前把手中心坐标可以表示为：

$$\begin{cases} x(t) = x_{O_2} + r \cos \left[\arctan \left(\frac{y_E - y_{O_2}}{x_E - y_{O_2}} \right) - \frac{v(t - t_1)}{r} \right], \\ y(t) = y_{O_2} + r \sin \left[\arctan \left(\frac{y_E - y_{O_2}}{x_E - y_{O_2}} \right) - \frac{v(t - t_1)}{r} \right]. \end{cases}$$

第四段：盘出螺线段（离开调头空间后）

此阶段龙头在盘出螺线段上其坐标计算方程与第一段相同，但旋转方向相反的盘出螺线运动，因此坐标计算逻辑与第一段一致，仅需调整极角的符号以体现方向差异。故在 t 时刻点的坐标可表示为：

$$\begin{cases} x(t) = \frac{1.7}{2\pi} \theta(t) \cos(\theta(t)), \\ y(t) = \frac{1.7}{2\pi} \theta(t) \sin(\theta(t)). \end{cases}$$

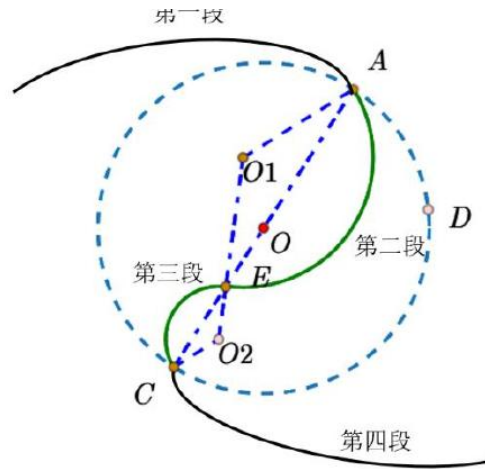


图 5_10 龙头前把手中心位置的四种运动阶段

综合上述内容，即可通过上述四个阶段的分时段计算，可覆盖龙头在调头区间及前后的完整运动过程，进而得到任意时刻（含 -100s 至 100s）龙头前把手中心的坐标。

（3）龙身后把手中心所处弧线段判断与位置计算

接下来，我们给出 Step1 具体实施的步骤。基于上述步骤中求得的任意时刻龙头前把手中心位置 (x_1, y_1) ，结合板凳龙单节板凳前后把手中心的固定间距 l ，通过构建判断函数确定后把手中心所处的弧线段，最后可以推导出相应的后把手位置坐标。

1、当前把手位置处于盘入螺线上时

由于板凳龙沿盘入螺线整体运动，且单节板凳前后把手的相对位置与螺线走向一致，当前把手处于盘入螺线时，则后把手也一定在盘入螺线上，可由上述第一段（原盘入螺线段）中计算结果直接得出。

2、当前把手位于大圆弧段（第一段圆弧，圆心为 O_1 ）上时

可引入判断角 φ' ，如下图所示：

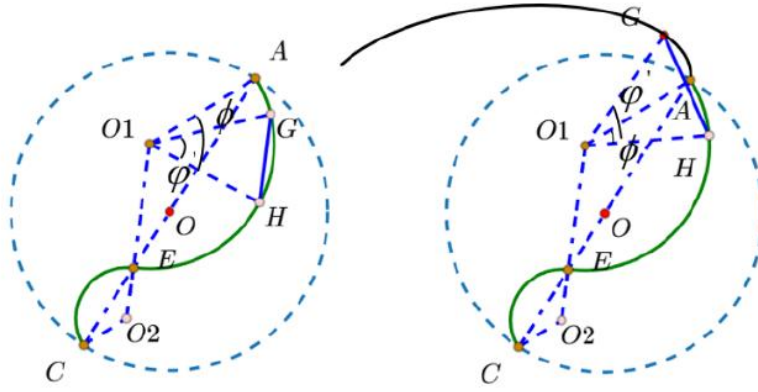


图 5_11 龙身前把手位于大圆弧段

由余弦定理

$$(2r)^2 + (2r)^2 - 2 \cdot (2r)^2 \cos \varphi' = l^2$$

可得:

$$\varphi' = \arccos\left(\frac{8r^2 - l^2}{8r^2}\right).$$

设前把手与 O_1 点连线和 O_1A 线段的夹角为 ϕ , 可表示为:

$$\phi = \arctan \frac{\frac{y_A - y_{O_1}}{x_A - x_{O_1}} - \frac{y_1 - y_{O_1}}{x_1 - x_{O_1}}}{1 + \frac{y_A - y_{O_1}}{x_A - x_{O_1}} \frac{y_1 - y_{O_1}}{x_1 - x_{O_1}}}.$$

- ① 当 $\phi > \varphi'$ 时, 后把手中心在第一段圆弧上;
- ② 当 $\phi \leq \varphi'$ 时, 后把手中心在盘入螺线上。

对于情况①的位置迭代公式可以直接给出:

β 满足

$$(2r)^2 + (2r)^2 - 2(2r)^2 \cos \beta = l^2,$$

可得:

$$\beta = \arccos\left(\frac{8r^2 - l^2}{8r^2}\right),$$

$$\gamma = \arctan \frac{\frac{y_A - y_{O_1}}{x_A - x_{O_1}} - \frac{y_1 - y_{O_1}}{x_1 - x_{O_1}}}{1 + \frac{y_A - y_{O_1}}{x_A - x_{O_1}} \frac{y_1 - y_{O_1}}{x_1 - x_{O_1}}}, \alpha = \gamma - \beta.$$

因此 G 的坐标为:

$$(x_2, y_2) = \left(x_{O_1} + 2r \cos \left[\arctan \left(\frac{y_{O_1} - y_A}{x_{O_1} - x_A} \right) + \pi - \alpha \right], y_{O_1} + 2r \sin \left[\arctan \left(\frac{y_{O_1} - y_A}{x_{O_1} - x_A} \right) + \pi - \alpha \right] \right).$$

对于情况②, 可以得到下述式子:

$$OG = \frac{D}{2} + \frac{d\alpha}{2\pi}, OH = \sqrt{x_1^2 + y_1^2}, \angle GOH = \alpha + \angle AOH.$$

但在该式子中, 其 α 和 OG 的值未知, 可在三角形中由余弦定理得, 如下:

在 $\triangle OGH$ 中, 由余弦定理可得:

$$OG^2 + OH^2 - 2OG \cdot OH \cos(\angle HOG) = l^2.$$

在 $\triangle AO_1H$ 中, 由余弦定理可得:

$$(2r)^2 + (2r)^2 - 2(2r)^2 \cos \beta = AH^2.$$

在 $\triangle AOH$ 中, 由余弦定理可得:

$$\left(\frac{D}{2} \right)^2 + OH^2 - D \cdot OH \cos(\angle AOH) = AH^2.$$

联立上述三个式子, 可得出 α 和 OG 的值。所以, G 点坐标为:

$$(x_2, y_2) = \left[\left(\frac{D}{2} + \frac{d\alpha}{2\pi} \right) \cos \left(\frac{D\pi}{d} + \alpha \right), \left(\frac{D}{2} + \frac{d\alpha}{2\pi} \right) \sin \left(\frac{D\pi}{d} + \alpha \right) \right].$$

3、当前把手位于小圆弧段（第二段圆弧, 圆心为 O_2 ）上时

同上述 φ' 满足余弦定理:

$$r^2 + r^2 - 2r^2 \cos \varphi' = l^2,$$

得:

$$\varphi' = \arccos \left(\frac{2r^2 - l^2}{2r^2} \right).$$

设前把手中心与 O_2 连线的线段与 O_2E 的夹角为 ϕ , 表示为:

$$\phi = \arctan \frac{\frac{y_E - y_{O_2}}{x_E - x_{O_2}} - \frac{y_1 - y_{O_2}}{x_1 - x_{O_2}}}{1 + \frac{y_E - y_{O_2}}{x_E - x_{O_2}} \frac{y_1 - y_{O_2}}{x_1 - x_{O_2}}}.$$

① 当 $\phi > \varphi'$ 时, 后把手中心在第二段圆弧上;

② 当 $\phi \leq \varphi'$ 时, 后把手中心在第一段圆弧上。

对于情况①:

在 $\triangle GO_2H$ 中, 由余弦定理

$$r^2 + r^2 - 2r^2 \cos \beta = l^2$$

可得:

$$\beta = \arccos \left(\frac{2r^2 - l^2}{2r^2} \right).$$

又由 $\alpha = \gamma - \beta$ 可得 G 点坐标为:

$$\begin{aligned} (x_2, y_2) = & \left(x_{O_2} + 2r \cos \left[\arctan \left(\frac{y_{O_2} - y_A}{x_{O_2} - x_A} \right) + \alpha - \varphi' \right], y_{O_2} \right. \\ & \left. + 2r \sin \left[\arctan \left(\frac{y_{O_2} - y_A}{x_{O_2} - x_A} \right) + \alpha - \varphi' \right] \right). \end{aligned}$$

对于情况②:

由于 H, E 点均已知, 因此在其中角 β 已知的情况下, 在 $\triangle O_1HO_2$ 中, 由余弦定理可知:

$$r^2 + (3r)^2 - 6r^2 \cos \beta = O_1H^2,$$

$$\frac{O_1H}{\sin \beta} = \frac{O_2H}{\sin \angle HO_1O_2}.$$

在 $\triangle O_1HG$ 中, 由余弦定理可知:

$$O_1H^2 + (2r)^2 - 4r \cdot OH \cos(\angle HO_1G) = l^2.$$

由

$$\alpha = \angle HO_1G - \angle HO_1O_2,$$

联立可解得, G 点坐标为:

$$(x_2, y_2) = \left(x_{O_1} + 2r \cos \left[\arctan \left(\frac{y_{O_1} - y_A}{x_{O_1} - x_A} \right) + \pi + \alpha - \varphi' \right], y_{O_1} + 2r \sin \left[\arctan \left(\frac{y_{O_1} - y_A}{x_{O_1} - x_A} \right) + \pi + \alpha - \varphi' \right] \right).$$

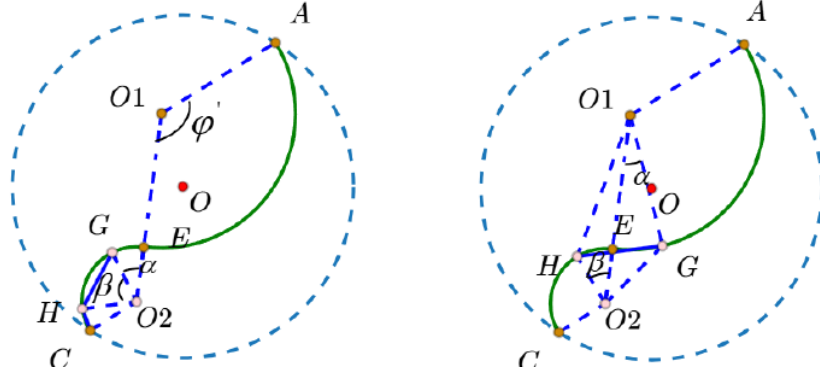


图 5_12 前把手位于小圆弧段的两种情况

4、当前把手位于盘出螺线时

角 φ' 满足：

$$\left(\frac{D}{2} + \frac{d}{2\pi} \varphi' \right)^2 + \left(\frac{D}{2} \right)^2 - D \left(\frac{D}{2} + \frac{d}{2\pi} \varphi' \right) \cos \varphi' = l^2.$$

由上述方程可以求解 φ' 。

设前把手中心与 O 点连线的线段与 OC 的夹角是 ϕ ，其中：

$$\phi = \frac{2\pi \left(\sqrt{x_1^2 + y_1^2} - \frac{D}{2} \right)}{d}.$$

① 当 $\phi > \varphi'$ 时，后把手中心在盘出螺线上；

② 当 $\phi \leq \varphi'$ 时，后把手中心在第二段圆弧上。

如果情况为②时，由螺线方程

$$OH = \frac{D}{2} + \frac{d\beta}{2\pi}.$$

在 $\triangle OCH$ 中由正弦、余弦定理得：

$$\begin{cases} \frac{D^2}{4} + \left(\frac{D}{2} + \frac{dR}{2\pi} \right)^2 - D \left(\frac{D}{2} + \frac{dR}{2\pi} \right) \cos \beta = CH^2, \\ \frac{\sin \beta}{CH} = \frac{\sin(\angle OCH)}{OH}, \end{cases} \Rightarrow \angle O_2CH = \angle OCH - \frac{\pi - \varphi'}{2}.$$

在 $\triangle O_2CH$ 中由余弦、正弦定理可得：

$$\begin{cases} r^2 + CH^2 - 2rCH\cos(\angle O_2CH) = O_2H^2, \\ \frac{\sin(\angle O_2HC)}{r} = \frac{\sin(\angle O_2CH)}{O_2H} = \frac{\sin(\angle CO_2H)}{CH}, \end{cases}$$

$$\Rightarrow \alpha = \varphi' - \angle GO_2C.$$

所以， G 点坐标为：

$$(x_2, y_2) = \left(x_{O_2} + r\cos \left[\arctan \left(\frac{y_{O_2} + y_A}{x_{O_2} + x_A} \right) + \alpha - \varphi' \right], y_{O_2} + r\sin \left[\arctan \left(\frac{y_{O_2} + y_A}{x_{O_2} + x_A} \right) + \alpha - \varphi' \right] \right).$$

如果情况为①，则可根据上述前把手位置处于盘入螺线上的情况取反方向求得。

(4) 板凳各把手速度的迭代公式

在 (3) 中已经求出在任意时刻时各点的坐标，考虑到在不同的圆弧中的速度是不同的，所以要进行分类讨论，在分类讨论之前，先进行一些准备工作：

$$\left\{ \begin{array}{l} \text{在螺线上的斜率: } k_1 = \frac{\sin\theta_1 + \theta_1\cos\theta_1}{\cos\theta_1 - \theta_1\sin\theta_1}, \\ \text{在大圆弧上的斜率: } k_1 = \frac{x_1 - x_{O_1}}{y_1 - y_{O_1}}, \\ \text{在小圆弧上的斜率: } k_1 = \frac{x_1 - x_{O_2}}{y_1 - y_{O_2}}, \\ \text{板凳的斜率: } k = \frac{y_2 - y_1}{x_2 - x_1}. \end{array} \right.$$

前把手所处位置的切线倾斜角与该板凳的倾斜角之差为夹角 α ，后把手所处位置的切线倾斜角与该板凳的倾斜角之差为夹角 β ，下面为角 α 和角 β 的计算公式：

$$\begin{cases} \alpha = \arctan \frac{k_1 - k}{1 + k_1k}, \\ \beta = \arctan \frac{k_2 - k}{1 + k_2k}. \end{cases}$$

下面对前后把手所处位置的不同进行对速度的分类讨论：

1、前后把手均在盘入螺线段

龙头速度恒定为 1m/s，并且前后把手运动规律一致，可直接由问题一中求速度的迭代公式类似地求出本题中的速度迭代公式：

$$v_i \cos \alpha = v_{i+1} \cos \beta.$$

2、前把手在大圆弧上，后把手位于盘入螺线上

则可求出前把手和后把手所处位置的斜率：

$$\text{前把手所处位置斜率： } k_i = \frac{x_i - x_{O_1}}{y_i - y_{O_1}},$$

$$\text{后把手所处位置的斜率： } k_{i+1} = \frac{\sin \theta_i + \theta_i \cos \theta_i}{\cos \theta_i - \theta_i \sin \theta_i}.$$

再根据速度迭代公式：

$$v_i \cos \alpha = v_{i+1} \cos \beta,$$

可求得当前把手在大圆弧上，后把手位于盘入螺线上时的速度迭代公式。

3、前后把手均在大圆弧上

由于均在大圆弧上，板凳绕点 O_1 旋转，其角速度和线速度均相同，故在此圆弧上时的速度迭代公式为：

$$v_i = v_{i+1}.$$

4、前把手处于小圆弧上，后把手处于大圆弧上

则前后把手所处位置所对应的斜率为：

$$\text{前把手所处位置斜率： } k_i = \frac{x_i - x_{O_2}}{y_i - y_{O_2}},$$

$$\text{后把手所处位置斜率： } k_{i+1} = \frac{x_{i+1} - x_{O_1}}{y_{i+1} - y_{O_1}}.$$

由上述斜率可计算出 α 和 β 的值：

$$\begin{cases} \alpha = \arctan \frac{k_1 - k}{1 + k_1 k}, \\ \beta = \arctan \frac{k_2 - k}{1 + k_2 k}. \end{cases}$$

再由速度迭代公式：

$$v_i \cos \alpha = v_{i+1} \cos \beta,$$

可计算出前把手处于小圆弧上，后把手处于大圆弧上时的速度。

5、前后把手均在小圆弧上

由于均在小圆弧上，板凳绕点 O_2 旋转，其角速度和线速度均相同，故在此圆弧上时的速度迭代公式为：

$$v_i = v_{i+1}.$$

6、前把手在盘出螺线上，后把手在小圆弧上

则前后把手所处位置所对应的斜率为：

$$\text{前把手所处位置斜率: } k_i = \frac{-\sin\theta_i + \theta_i \cos\theta_i}{\cos\theta_i - \theta_i \sin\theta_i},$$

$$\text{后把手所处位置斜率: } k_{i+1} = \frac{x_{i+1} - x_{O_2}}{y_{i+1} - y_{O_2}}.$$

由上述斜率可计算出 α 和 β 的值：

$$\begin{cases} \alpha = \arctan \frac{k_1 - k}{1 + k_1 k}, \\ \beta = \arctan \frac{k_2 - k}{1 + k_2 k}. \end{cases}$$

再由速度迭代公式：

$$v_i \cos\alpha = v_{i+1} \cos\beta.$$

可计算出前把手在盘出螺线上，后把手在小圆弧上则前后把手所处位置所对应的速度。

7、前后把手均处于盘出螺线上

$$\text{前把手所处位置斜率: } k_i = \frac{-\sin\theta_i + \theta_i \cos\theta_i}{\cos\theta_i - \theta_i \sin\theta_i},$$

$$\text{后把手所处位置斜率: } k_{i+1} = \frac{-\sin\theta_{i+1} + \theta_{i+1} \cos\theta_{i+1}}{\cos\theta_{i+1} - \theta_{i+1} \sin\theta_{i+1}}.$$

由上述斜率可计算出 α 和 β 的值：

$$\begin{cases} \alpha = \arctan \frac{k_1 - k}{1 + k_1 k}, \\ \beta = \arctan \frac{k_2 - k}{1 + k_2 k}. \end{cases}$$

再由速度迭代公式：

$$v_i \cos\alpha = v_{i+1} \cos\beta,$$

可计算出前后把手均在盘出螺线上所对应的速度。

综上，通过上述 7 类场景的分类推导，可知道任意时刻前后把手处于的弧段类型，代入对应公式计算后把手的速度分量。结合 Step1、Step2 的位置求解结果，最终实现 -100s 至 100s 内每秒各板凳所有把手的位置与速度的完整计算[7]。

5.4.2 模型结果及分析

将上述内容通过迭代得到如下结果：

表 5_6 板凳龙特殊节点位置结果

	-100s	-50s	0s	50s	100s
龙头 x(m)	7.778034	6.608301	-2.711856	1.332696	-3.157229
龙头 y(m)	3.717164	1.898865	-3.591078	6.175324	7.548511
第 1 节龙身 x(m)	6.209273	5.366911	-0.063534	3.862265	-0.34689
第 1 节龙身 y(m)	6.108521	4.475403	-4.670888	4.840828	8.079166
第 51 节龙身 x(m)	-10.608038	-3.629945	2.459962	-1.671385	2.095033
第 51 节龙身 y(m)	2.831491	-8.9638	-7.778145	-6.076713	4.033787
第 101 节龙身 x(m)	-11.922761	10.125787	3.008493	-7.591816	-7.288774
第 101 节龙身 y(m)	-4.802378	-5.972247	10.108539	5.175487	2.063875
第 151 节龙身 x(m)	-14.351032	12.974784	-7.002788	-4.605165	9.462513
第 151 节龙身 y(m)	-1.980993	-3.810357	10.337482	-10.386988	-3.540357
第 201 节龙身 x(m)	-11.952942	10.522508	-6.872842	0.336952	8.524374

	-100s	-50s	0s	50s	100s
第 201 节龙身 y (m)	10.566998	-10.807425	12.382609	-13.17761	8.606933
龙尾（后）x (m)	-1.011059	0.189809	-1.933628	5.859094	-10.980157
龙尾（后）y (m)	-16.527573	15.720588	-14.713128	12.612894	-6.770006

表 5_7 板凳龙特殊节点速度结果

	-100s	-50s	0s	50s	100s
龙头 (m/s)	1.000000	1.000000	1.000000	1.000000	1.000000
第 1 节龙身 (m/s)	0.999904	0.999762	0.998687	1.000363	1.000124
第 51 节龙身 (m/s)	0.999346	0.998642	0.995134	0.949935	1.003966
第 101 节龙身 (m/s)	0.999091	0.998248	0.994448	0.948482	1.096263
第 151 节龙身 (m/s)	0.998944	0.998047	0.994156	0.948038	1.095306
第 201 节龙身 (m/s)	0.998849	0.997925	0.993994	0.947823	1.094933
龙尾（后） (m/s)	0.998817	0.997885	0.993944	0.94776	1.094833

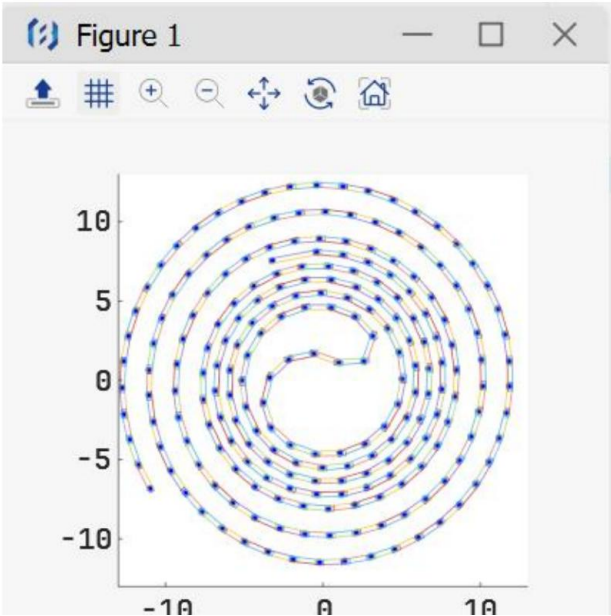


图 5_13 t=100s 时板凳龙位置示意图

5.5 问题五模型建立与求解

5.5.1 模型建立

首先，先根据问题四中所求出的 $[-100, 100]$ 秒各把手中间点的速度建立速度矩阵，并

从中求出速度矩阵中的最大值 v_{max} ；接着再将龙头速度变为原来的 k 倍，根据我们的速度迭代公式，当龙头速度变为原来的 k 倍时，其余各节板凳的速度也变为原来的 k 倍，因此可限制 $v'_{max} = 2\text{ m/s}$ ，根据

$$k = \frac{v'_{max}}{v_{max}},$$

进而确定 k 值，再根据算式

$$v = kv_0,$$

即可计算得到龙头的最大盘绕速度。

5.5.2 模型结果及分析

将上述内容进行求解，得出龙头的最大行进速度为 1.406276m/s 。

六、模型评价与推广

6.1 模型的优点

1. 模型充分考虑了板凳龙的几何结构与运动特性，通过极坐标与直角坐标的转换、余弦定理、速度分解等方法，使得模型具有明确的物理解释和较高的可信度。
2. 本文通过迭代计算、步长调整、几何判断等方法实现了各问题的数值求解，并辅以轨迹图、速度变化图、碰撞示意图等可视化手段，直观展示了模型结果，增强了说服力。

6.2 模型的缺点

模型假设板凳为刚体、连接无摩擦、忽略风力与地面摩擦等外部干扰，这些理想化条件与实际表演环境存在差距，可能影响模型的实用性和准确性。

7.3 模型的推广

本模型不仅适用于“板凳龙”这一传统民俗活动的运动分析与路径规划，还可推广至其他类似链式结构的运动系统，如：舞龙舞狮、游行队伍等集体表演活动的队形控制与路径优化；多机器人编队运动中的避障与轨迹规划。

参考文献：

- [1] 施咸亮. 用阿基米德等距螺线逼近平面曲线[J]. 应用数学学报, 1980, (03):255-261.
- [2] 包志轩, 王盈盈, 周逸轩, 等. 板凳龙调头路径模型[J]. 台州学院学报, 2025, 47(03):7-13.
- [3] 李宇航, 巫如山, 张驰, 等. 民俗活动“板凳龙”行进状态建模与路径优化研究[J/OL]. 实验科学与技术, 1-7[2025-09-01].

[4] 蔡志杰. 板凳龙运动轨迹模型的分析研究[J]. 数学建模及其应用, 2025, 14(01):25-32.

[5] 游天明, 陈诺严, 陈静怡, 等. 基于动态搜索的“板凳龙”运动状态及路线研究[J]. 数学建模及其应用, 2025, 14(01):33-41.

[6] 吴铭威, 齐丞英, 张丁元. “板凳龙”调头区域优化分析[J]. 数学建模及其应用, 2025, 14(01):49-57.

[7] 王吉彬, 倪力赞, 于明硕. “板凳龙”的螺线运动分析与求解[J]. 南通职业大学学报, 2025, 39(01):60-64+93.

附录

支撑材料的文件列表：

序号	文件名称	文件内容
1	result1.xlsx	问题一所求得的结果
2	result2.xlsx	问题二所求得的结果
3	result4.xlsx	问题三所求得的结果
4	m1.m	问题一的代码
5	m2.m	问题二的代码
6	m3.m	问题三的代码
7	m4.m	问题四的代码
8	m5.m	问题五的代码

202503080104 基于多元模型与智能算法的灌溉系统协同优化及应急决策研究

作者：司响 王怡涵

指导老师：周见文

学校：云南大学

摘要：本文针对 1 公顷农场的智能灌溉系统优化问题，结合河流引水与储水罐设施，建

立土壤湿度预测、系统规划、旱灾应急及灌溉方案优化模型。综合建立多元线性回归、非线性规划、单目标线性规划等模型，并运用贪心算法、序列二次规划等方法对模型进行求解，以实现成本最小化与作物生长保障的平衡。

针对问题一：首先计算气象因子与土壤湿度的皮尔逊系数，筛选关键影响因素，构建多元线性回归模型，采用最小二乘法求解系数。对于题中未给的数据建立独立回归模型求解，得到与和相关的线性方程，通过题中所给数据求出的值。最后，将全部关键数据代入某天四个时段的气象数据，预测各时段土壤湿度分别为 0.35826、0.31427、0.30929、0.30838，并取均值，得当天土壤湿度为 0.32255。

针对问题二：以土壤湿度 0.22 与其他题中条件为约束，构建非线性规划模型。我们发现喷头全覆盖与非全覆盖导致的成本差异很大，因此同时考虑两种情形下的最优布线规划。对于非全覆盖的情形，不考虑全覆盖约束条件。接着，采用贪心算法优化喷头布局，并利用 K-Means 聚类确定储水罐位置最小化管网连接成本，通过 Prim 算法构建最小生成树管网。得到应建设 10 喷头，3 储水罐，成本 251598.00 元的方案。

针对问题三：在问题二优化的灌溉系统的基础上，构建单目标线性规划模型以最大化作物存活面积。在旱灾时河流水量为正常值 80%的条件下，加入供水能力、正常生长及面积等约束条件。利用交替方向乘子算法求解线性规划，确定三种作物存活面积均为种植面积，正常生长面积分别为 0.1264、0.1202、0.0436 公顷。接着，利用网格搜索法进行遍历求解，确定最优应急储备比例（分别为 20%、40%、80%、95%、100%），并用最小二乘法拟合出应急储备比例与旱灾概率的线性关系。

针对问题四：基于作物生长周期需水量差异，建立非线性规划模型。首先根据需水量的公式，结合种植面积、各阶段日需水强度及月度重叠天数，计算 5-7 月各作物的需水量。接着，通过实际灌溉需求公式计算出作物的实际灌溉需求量与需水量一致。利用序列二次规划法优化，求出最优应急储备比 0.2、储水罐总容积 58917.2、总成本 336934.25 元，验证现有系统满足所有约束。根据优化结果，确定各作物各月水源比例，进而根据比例的分配，明确各月各作物的总灌溉量与运行状态。最终，获得 5-7 月的灌溉方案，验证现有系统满足需求，无需修改布线。

关键词：智能灌溉系统；多元线性回归；（非）线性规划；贪心算法；序列二次规划

点 评：本文的特色在于将灌溉系统优化分解为四个紧密关联的子问题，从土壤湿度预

测到系统规划再到应急管理，构建了完整的智能灌溉决策体系。方法工具箱丰富，多元线性回归、非线性规划、贪心算法、K-Means 聚类、Prim 算法、序列二次规划等多种方法被恰当地应用在不同子问题中，展示了扎实的数学模型功底。特别是问题二将管网规划与设施选址相结合（贪心算法求喷头布局、K-Means 聚类定储水罐位置、Prim 算法建管网），体现了工程优化思维。结果输出具体可落地（10 喷头、3 储水罐、成本 251598.00 元），具有实际参考价值。建议可进一步考虑土壤湿度空间分布的不均匀性，以及不同作物根区深度差异对灌溉需求的影响。

基于多元模型与智能算法的灌溉系统协同优化及应急决策研究

摘要

本文针对 1 公顷农场的智能灌溉系统优化问题，结合河流引水与储水罐设施，建立土壤湿度预测、系统规划、旱灾应急及灌溉方案优化模型。综合建立多元线性回归、非线性规划、单目标线性规划等模型，并运用贪心算法、序列二次规划等方法对模型进行求解，以实现成本最小化与作物生长保障的平衡。

针对问题一：首先计算气象因子与土壤湿度的皮尔逊系数，筛选关键影响因素，构建多元线性回归模型，采用最小二乘法求解系数。对于题中未给的 Td 、 Tx 、 VV 数据建立独立回归模型求解，得到与 T 和 U 相关的线性方程，通过题中所给数据求出 Td 、 Tx 、 VV 的值。最后，将全部关键数据代入某天四个时段的气象数据，预测各时段土壤湿度分别为 0.35826、0.31427、0.30929、0.30838，并取均值，得当天土壤湿度为 0.32255。

针对问题二：以土壤湿度 ≥ 0.22 与其他题中条件为约束，构建非线性规划模型。我们发现喷头全覆盖与非全覆盖导致的成本差异很大，因此同时考虑两种情形下的最优布线规划。对于非全覆盖的情形，不考虑全覆盖约束条件。接着，采用贪心算法优化喷头布局，并利用 K -Means 聚类确定储水罐位置最小化管网连接成本，通过 $Prim$ 算法构建最小生成树管网。得到应建设 10 喷头，3 储水罐，成本 251598.00 元的方案。

针对问题三：在问题二优化的灌溉系统的基础上，构建单目标线性规划模型以最大化作物存活面积。在旱灾时河流水量为正常值 80%的条件下，加入供水能力、正常生长及面积等约束条件。利用交替方向乘子算法求解线性规划，确定三种作物存活面积均为种植面积，正常生长面积分别为 0.1264、0.1202、0.0436 公顷。接着，利用网格搜索法进行遍历求解，确

定最优应急储备比例（分别为 20%、40%、80%、95%、100%），并用最小二乘法拟合出应急储备比例与旱灾概率的线性关系。

针对问题四：基于作物生长周期需水量差异，建立非线性规划模型。首先根据需水量的公式，结合种植面积、各阶段日需水强度及月度重叠天数，计算 5-7 月各作物的需水量。接着，通过实际灌溉需求公式计算出作物的实际灌溉需求量与需水量一致。利用序列二次规划法优化，求出最优应急储备比 0.2、储水罐总容积 58917.2 L、总成本 336934.25 元，验证现有系统满足所有约束。根据优化结果，确定各作物各月水源比例，进而根据比例的分配，明确各月各作物的总灌溉量与运行状态。最终，获得 5-7 月的灌溉方案，验证现有系统满足需求，无需修改布线。

关键词：智能灌溉系统、多元线性回归、（非）线性规划、贪心算法、序列二次规划法

一、问题重述

1.1 问题背景

农业生产中，水资源的高效管理对作物生长至关重要，尤其在多种土壤类型、动态天气条件及不同作物需求的复杂场景下，智能灌溉系统的优化配置与实时调整是关键问题。某农场为 1 公顷正方形区域，依河而建，于 5 月 1 日同时播种高粱、玉米、大豆三种作物，可通过河流引水和储水罐进行灌溉，且需考虑灌溉系统的建设成本与覆盖范围约束。

1.2 问题的提出

基于以上背景和所给数据，需要建立数学模型解决以下问题：

(1) 综合历史气象数据，建立土壤湿度与其他气象数据的关系模型，并利用表 2 中某天不同小时的气象数据，预测当天对应的土壤湿度。

(2) 考虑作物不同生长时期的需水量，仅基于土壤湿度约束（最低需 ≥ 0.22 ），利用 2021 年 7 月 1 日至 7 月 31 日的数据，规划引水管布线、储水罐的容积与位置，在满足作物存活的前提下，使系统总建设成本最小，并给出规划图及总花费。

(3) 在问题 2 的基础上，储水罐需储备不可用于日常灌溉的应急水源，而应急时可供给半径 50m 内的喷头。当旱灾来临时，河流量仅为问题 2 中引水系统总流量的 80%，要确定该系统能保证的作物存活面积与正常生长面积，同时分析储水罐应急储备水源比例与旱灾概率的关系。

(4) 已知各作物成熟期均为 20 天，利用 2021 年 5 月 1 日至 7 月 31 日的数据，规划不同作物每月的灌溉方案，检查现有系统是否满足需求，若不满足则修改布线，并将月灌溉情况可视化。

二、问题分析

针对问题一：在问题一中，基于历史气象数据，要求建立土壤湿度与其他气象数据的关系模型，并利用该模型预测给定时刻的土壤湿度。由于土壤湿度受温度、降水量、风向等多因素影响，我们需要通过数据整合与模型拟合实现预测。我们将首先对文字变量进行编码处理，将最低土壤湿度数据作为因变量考虑。将建立多元线性拟合模型，通过计算各气象变量与 $5cm-SM$ 的皮尔逊系数，可初步判断出主要的影响项。其次构建多元线性的预测方程，并最小化预测值与真实值的均分误差。我们将采用最小二乘法求解出回归方程的系数，最后将表 2 中给定时刻的气象数据代入模型，即可计算该天的 $5cm-SM$ 值。

针对问题二：在问题二中，需要以土壤湿度 ≥ 0.22 为核心约束，结合 2021 年 7 月的土

壤湿度与气象数据，在 $100m \times 100m$ 农场中，通过优化引水管布线与储水罐的位置及容积实现灌溉系统建设总花费最小化。我们将建立非线性规划模型，对农场进行网格剖分并设置约束条件。在求解的过程中，我们将利用贪心算法得到喷头的布局、*Prim* 算法构建管网最小生成树、*K-Means* 聚类优化储水罐位置，并结合多参数搜索确定最优参数组合，在满足题设条件的同时实现总成本最小。

针对问题三：在问题二已优化的灌溉系统基础上，当河流水量降至问题二引水系统总流量的 80%时，需确定该系统能保障的作物存活面积与正常生长面积，并分析应急储备水源比例与旱灾概率的关系。由于该问题涉及供水能力约束、作物需水差异及风险概率关联，我们将构建以最大化存活面积为目标的线性规划模型，结合旱灾概率与储备比例的多情景模拟，量化不同风险下的最优储备策略，以平衡应急供水保障与成本效率。

针对问题四：我们需要结合作物生长周期需水量的差异，并利用 2021 年 5 月至 7 月的数据规划每月灌溉方案。由于不同作物各阶段需水量不同，需要考虑自然降水对灌溉需求的影响，同时需验证现有的系统是否满足月度供水需求，必要时调整布线。因此，我们建立非线性规划模型，以最小化总成本为目标，整合各种约束，并计算各作物月度需水量与灌溉量，优化河水与储水罐的供水比例，最终呈现 5 月到 7 月的灌溉情况。

三、模型假设

1. 假设播种期、开花期、成熟期严格按题目给定天数划分，不考虑早熟或晚熟情况；
2. 假设一天之内土壤湿度不随时间变化；
3. 假设河流与农场上边界距离为 0；
4. 喷头喷淋半径 $15m$ 为理想圆形覆盖，无风力干扰或作物遮挡；
5. 假设管道、喷头、储水罐在规划期内无损坏或维护需求；
6. 假设河流水量随旱灾概率线性衰减，储水罐应急储备水在 7 天内均匀分配；
7. 假设应急储备比例与旱灾概率正相关，正常生长面积 $\leq$ 存活面积 $\leq$ 种植面积；
8. 假设作物需水量分为“最低需水量”和“正常需水量”均为每日值，且需水量与种植面积、有效覆盖比例正相关；
9. 假设需水量低的作物在缺水时优先存活。

四、符号说明

符号	符号说明	单位
S_i	作物 i 的存活面积	m^2

v_i	第 i 个储水罐容积	L
T	应急供水天数	天
A_c	表示作物 c 的种植面积	m^2
$\theta_{c,m}$	表示作物 c 在月份 m 的河水供水比例	/
$W_{c,m}$	作物 c 在月份 m 的需水量	L
D_i^{norm}	表示第 i 种作物的正常需水量	L
D_i^{min}	表示第 i 种作物的存活需水量	L

五、数据预处理

变量处理：

将数据中 RRR 中的无降水和有降水痕迹都转化为 0， DD 列中 10 种风向转化为数字具体见表 1。

表 1 风向数字编号

风向	编号
无风	0
东	1
南	2
西	3
北	4
东北	5
东南	6
西南	7
西北	8
未知	9

数据聚合：

由于气象数据信息琐碎复杂，为保证气象数据与土壤湿度一一匹配。我们将一天内各时间段的气象数据整合为以天为单位的数据，具体聚合规则如下：

对于降水量：

$$RRR_i = \sum_j RRR_{ij}$$

其中， RRR_i 表示第 i 天的降水总量， RRR_{ij} 表示第 i 天，第 j 个时段的降水量。

对于风向：对每天八个时段的风向取众数作为该天的风向。

对于其他气象数据：

$$x_i = \frac{\sum_{j=1}^8 x_{ij}}{8}$$

其中， x_i 表示第 i 天某个气象数据的均值， x_{ij} 表示第 i 天，第 j 个时段的某个气象数据的值。

数据分类：

基于题设要求，我们根据数据的特征并查阅中国气象局发布的农业干旱监测预报指标及等级标准^[1]，设定一个标准判断某天是否干旱，判断标准如下：如果 7 天内的降水量远远不能满足农作物生长所需的水分，导致土壤墒情严重不足，影响农作物正常生长，就可认为处于农业干旱状态。比如对于某些需水量较大的农作物，7 天累计降水量少于 8.63 毫米可能会影响其生长，可视为干旱。

通过分析数据的基本统计信息，我们选择 50%分位数^[2]作为判断标准，对数据进行标注，最终共识别出 35 个干旱周期。

缺失值处理：

针对大量缺失的 Pa 列的值，我们同时采用均值填充^[3]和时间序列线性插值^[4]的方法作填充，具体求解过程见 $Q1.1.m$ 文件。

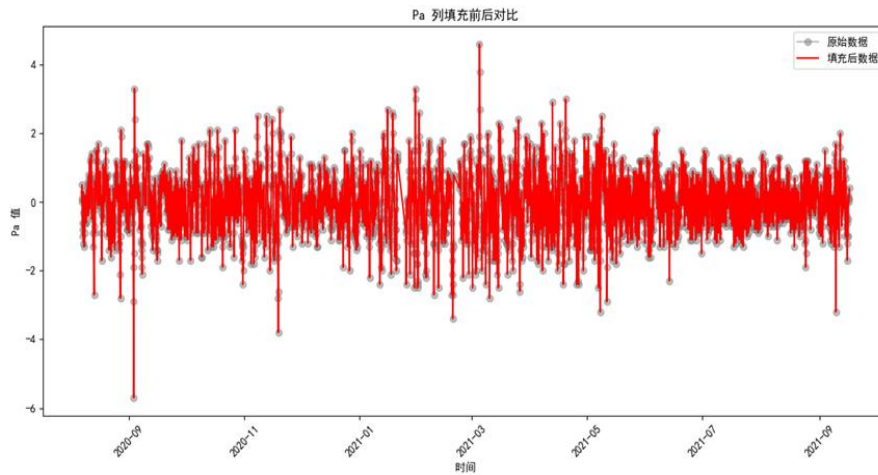


图 1 时间序列线性插值法

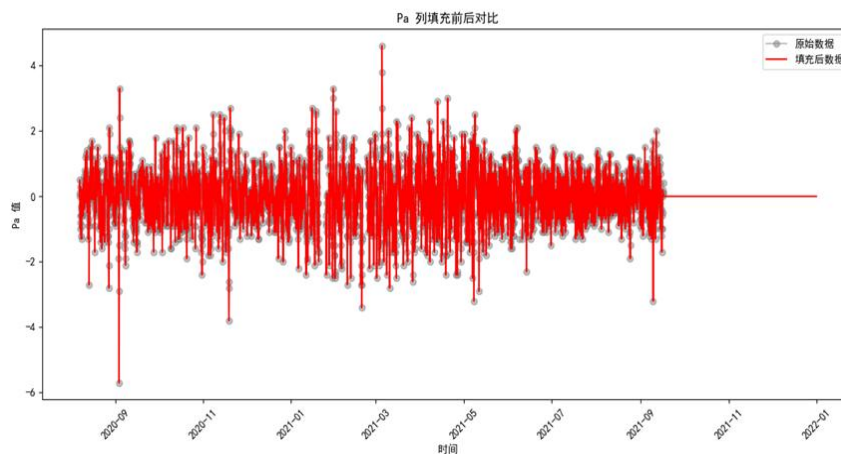


图 1 均值填充法

通过对比两种方法的结果，时间序列插值更贴合时间趋势，适用于均匀时间间隔，且气象数据中，温度、气压等指标随时间变化近似线性，因此我们选择时间序列线性插值法做缺失数据的填充，部分填充数据如表 2 所示，全部数据表结果见 cleaned_weather_data.xlsx 文件。

表 2 Pa 列部分数据填充图

时间	Pa
2021-03-14 23:00:00	-1.3
2021-03-14 20:00:00	-1.1
2021-03-14 17:00:00	-1.7
2021-03-14 14:00:00	-2.1
2021-03-14 11:00:00	-0.8

2021-03-14 08:00:00	0.4
2021-03-14 05:00:00	0.2
2021-03-14 02:00:00	-0.4
2021-03-13 23:00:00	-0.4
2021-03-13 20:00:00	0.3

六、模型的建立与求解

6.1 问题一的建模与求解

6.1.1 多元线性回归方程的建立

问题一要求建立预测模型，探究土壤湿度与其他气象数据的关系，并根据题中某天不同小时的气象数据预测当天土壤湿度。在农业生产中，需快速识别哪些农业相关因素对作物产量影响最显著，而线性模型可通过系数大小直接比较变量重要性，便于后续管理决策优化。基于问题分析及已处理数据的特征，我们建立多元线性回归模型^[5]来预测土壤湿度(y)。

我们首先求得气象数据中的因子与土壤湿度的皮尔逊系数如表 3 所示，具体代码见 `Q1.6.m` 文件。

表 3 各气象因子的皮尔逊系数

气象因子	皮尔逊系数
T	0.687421
Td	0.650476
tR	0.156587
Tx	0.616665
DD	-0.076147
RRR	0.251408
Tn	0.663651
Po	-0.428576
P	-0.488928
Pa	0.051022
U	0.165927
Ff	-0.116034

<i>ff3</i>	0.051435
<i>VV</i>	-0.503687

根据相关系数的结果，我们选取皮尔逊系数的绝对值大于 0.1 的值，这些因素具有较强相关性，因此选取了 T 、 Po 、 P 、 P 、 U 、 Ff 、 VV 、 Td 、 tR 、 Tn 、 Tx 、 RRR 这 11 个因素作为主要的影响项来构建多元线性回归模型。

假设土壤湿度与各农业管理因素之间存在线性关系。建立线性多元回归模型如下：

$$y = \beta_0 + \beta_1 T + \beta_2 Po + \beta_3 P + \beta_4 U + \beta_5 Ff + \beta_6 VV + \beta_7 Td + \beta_8 tR + \beta_9 Tn + \beta_{10} Tx + \beta_{11} RRR + \varepsilon$$

其中， β_0 是截距项， $\beta_1, \beta_2, \dots, \beta_{11}$ 是各个自变量的回归系数， ε 是随机误差项。

将数据带入公式(3)中。目标最小化预测值与实际值之间的均方误差：

$$\min \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

其中， N 表示训练样本的数量， y_k 表示第 k 个样本的实际土壤湿度值， $\hat{y}_k$ 表示模型预测的土壤湿度值。

6.1.2 多元线性回归方程的求解

我们首先对数据集进行划分，将数据按 7:3 的比例划分为训练集和验证集，训练集通过历史数据学习气象因子于土壤湿度的关系求解回归系数；验证集用于评估模型泛化能力，确保系数不是对特定数据的过拟合结果。接着，我们采用最小二乘法求解回归方程的系数，回归系数表示影响方向，正系数表示因子与土壤湿度呈现正向相关，系数的绝对值反应因子对土壤湿度的影响强度。

通过多元线性回归方程的求解，我们得到回归方程：

$$y = -0.0638 + 0.0025T + 0.1546Po - 0.1500P + 0.0028U - 0.0022Ff \\ + 0.001VV - 0.0111Td - 0.0003tR - 0.0001Tn - 0.001Tx + 0.0001RRR$$

我们对得到的回归方程进一步处理，发现 tR 、 Tn 、 RRR 的系数均小于 0.001，说明这些气象变量对土壤湿度的实际影响微乎其微，删除系数过小项可使模型更简洁且不损失预测精度。修正后回归方程如下：

$$y = -0.0638 + 0.0025T + 0.1546Po - 0.1500P + 0.001VV \\ + 0.0028U - 0.0022Ff - 0.0111Td - 0.001Tx$$

对于题中未给出的数据 Td 、 Tx 、 VV ，我们同样建立线性回归模型，以此获得对应因素

的数据，建立如下回归方程：

$$\begin{cases} Td = \beta'_0 + \beta'_1 T + \beta'_2 U \\ Tx = \beta''_0 + \beta''_1 T + \beta''_2 U \\ VV = \beta'''_0 + \beta'''_1 T + \beta'''_2 U \end{cases}$$

与土壤湿度回归方程求解方法一致，最后我们得到：

$$\begin{cases} Td = -23.0112 + 0.9318T + 0.2520U \\ Tx = 8.9480 + 0.8139T + 0.0036U \\ VV = 20.4504 - 0.2746T - 0.0775U \end{cases}$$

利用题中数据，我们加入 Td 、 Tx 、 VV 三个气象因子，结果见下表 4。

表 4 气象因子各时间段数值

时间	T	Po	P	Pa	U	DD	Ff	RRR	Td	Tx	VV
02	19.3	731.5	751.7	1.0	99	西	轻风	15.0	19.9	25.0	7.5
05	20.0	732.0	752.4	0.5	94	西南	轻风	6.0	19.3	25.6	7.7
08	23.4	732.8	753.2	0.8	80	西	轻风	0	18.9	28.3	7.8
11	28.0	733.5	753.8	0.7	44	西北	轻风	0	14.2	31.9	9.4

6.1.3 结果分析

最后，我们利用土壤湿度的回归方程，分别求出四个时间的土壤湿度，并求均值得到当天土壤湿度，结果见表 5，具体代码见 $Q1.1.m$ 至 $Q1.7.m$ 文件。

表 5 问题一结果

时间 (h)	T	Po	P	Pa	U	DD	Ff	RRR	预测当天湿度 $5cm_SM$
02	19.3	731.5	751.7	1.0	99	西	轻风	15.0	0.32255
05	20.0	732.0	752.4	0.5	94	西南	轻风	6.0	
08	23.4	732.8	753.2	0.8	80	西	轻风	0	
11	28.0	733.5	753.8	0.7	44	西北	轻风	0	

当天土壤湿度为 0.32255，该值高于植物生长所需的最低土壤湿度 0.22，能够满足作物基本存活需求，且根据当天的气象数据，未出现极端干旱相关的气象条件，土壤湿度处于正常范围，与历史数据中气象因素对土壤湿度的影响规律相符，因此预测结果可信。

6.2 问题二模型的建立与求解

6.2.1 非线性规划模型的建立

问题二要求在不考虑作物不同生长时期需水量的情况下，仅依据土壤湿度，用 2021-07-01 至 2021-07-31 的数据规划引水管布线、储水罐的容积和位置，在满足作物存活前提下使总花费最小，给出规划图和花费。

(1) 网格剖分

在确保总成本最低的情况下，为尽可能使喷头覆盖面积扩大，在全覆盖情形下采用正六边形密铺理论^[6]：正六边形能够无重叠、无缝隙地铺满整个平面。因为其内角为 120 度，六个六边形恰好围绕一点形成 360 度。我们运用正六边形密铺理论对农场进行“正六边形”覆盖。由于喷头覆盖半径为 15m，喷头最小间距为 15m。为了更加全面的覆盖整个农场，我们做边长为 15m 的正六边形对正方形农场进行剖分。选中合适的节点后，做正六边形外接圆即可覆盖整个农场，划分情况见图 2。

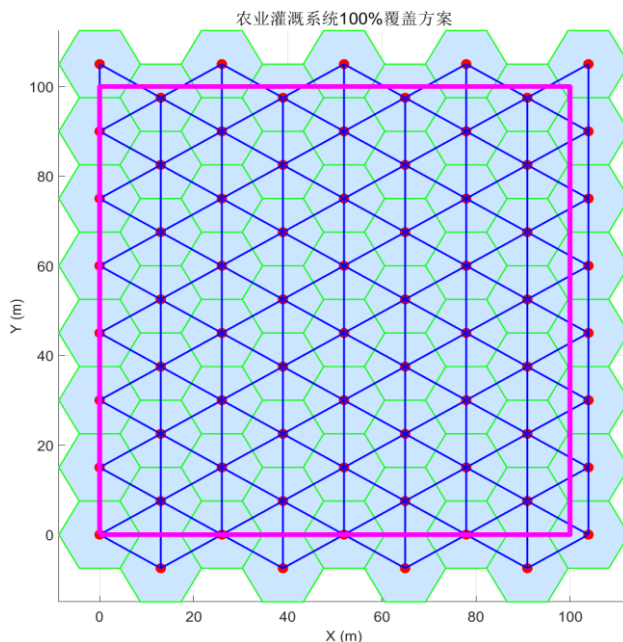


图 2 网格剖分示意图

对每个在正方形内的格点进行编号，不难观察到共有 46 个格点，编号为 $i=1, 2 \dots 46$ 。进一步剖分每一个正六边形，划分出共 10000 个节点。假设每个格点中心即为潜在喷头或储水罐布设位置。

(2) 非线性规划模型

由于每段水管建设成本是非线性的，因此我们将建立非线性规划模型。

决策变量：

$$x_i = \begin{cases} 0 & \text{表示第} i \text{个节点不连接河道供水} \\ 1 & \text{表示第} i \text{个节点连接河道供水} \end{cases}$$

$$y_i = \begin{cases} 0 & \text{表示第} i \text{个节点不建设储水罐} \\ 1 & \text{表示第} i \text{个节点建设储水罐} \end{cases}$$

目标函数：

我们要最小化建设花费，包括引水管道成本 $\sum_{i=1}^{10000} x_i (50l_i^{1.2} + 0.1q_i^{1.5})$ ，储水罐建设成本 $\sum_{j=1}^M 5v_j y_j$ ，即

$$\min \left[\sum_{i=1}^{10000} x_i (50l_i^{1.2} + 0.1Q_i^{1.5}) + \sum_{j=1}^M 5v_j y_j \right]$$

其中， l_i 表示河流至节点 i 的引水管道长度； Q_i 表示第 i 个节点的日流量； v_j 表示第 j 个储水罐的容积； M 为储水罐的数量。

约束条件：

最低湿度条件

根据 2021 年 7 月 1 日至 2021 年 7 月 23 日最低土壤湿度数据，我们引入每日相应格点的水分补偿需求函数 w_{it} 。

$$w_{it} = \begin{cases} m_d \cdot 0.05 \cdot S_i \cdot (\theta_{min} - \theta_{it}) & \theta_{it} < \theta_{min} \\ 0 & \theta_{it} \geq \theta_{min} \end{cases}$$

其中， θ_{it} 为第 i 个节点第 t 日的 $5cm_SM$ ； $\theta_{min} = 0.22$ ； $m_d = 1500kg/m^3$ 表示单位面积土壤干重； $S_i = 337.5\sqrt{3}m^2$ 为每个网格的面积。

存活条件约束

由于每日第 i 点供水灌溉需满足最低水分补偿需求，则有

$$t_i Q_i \geq \max w_{it}, \forall t$$

其中， $t_i = 0$ 表示第 i 个节点不建设管道； $t_i = 1$ 表示第 i 个节点建设管道。

喷头覆盖约束

所有以节点为中心的正六边形的并集应包含整个农场，确保农场每个点都至少被一个激活的喷头覆盖以确保农作物基本生存保障需求。即对 $\forall i \in C$,

$$\sum_{j \in \omega_i} t_j \geq 1$$

其中， ω_i 表示所有以半径为 $15m$ ，能覆盖到第 i 个节点的节点集。

对于喷头覆盖约束条件只在全覆盖的情形下使用，非全覆盖的情形下不考虑该约束条件。

相邻喷头间距离约束

由题意，相邻两喷头间距离 $\geq 15m$ ，这个距离限制考虑了喷头安装空间需求、管道布设的技术要求及避免水流相互冲突的工程实际。此约束通过计算任意两节点间欧氏距离来实施，确保在满足覆盖需求的同时，避免设备的过度密集配置。我们记任意两点节点坐标为 (m_i, n_i) 、 (m_j, n_j) ，则

$$\begin{aligned} d(i, j) &\geq 15 \quad (i \neq j) \\ d(i, j) &= \sqrt{(m_i - m_j)^2 + (n_i - n_j)^2} \end{aligned}$$

综上：我们建立非线性规划模型如下：

$$\begin{aligned} \min & \left[\sum_{i=1}^{46} x_i (50l_i^{1.2} + 0.1Q_i^{1.5}) + \sum_{j=1}^M 5v_j y_j \right] \\ \text{s.t.} & \begin{cases} t_i Q_i \geq \max w_{it}, \forall t \\ \sum_{j \in \omega_i} t_j \geq 1 \\ d(i, j) \geq 15 \quad (i \neq j) \\ x_i, y_i, t_i \in \{0, 1\} \\ q_i, v_i \geq 0 \end{cases} \end{aligned}$$

6.2.2 非线性规划模型的求解

经过分析，我们认为，对模型需要分为全覆盖和非全覆盖情形，详细过程如下：

第 1 步：进行数据处理，将 $100m \times 100m$ 农场划分为 $1m \times 1m$ 的网格，生成 10000 个格点，基于 2021 年 7 月土壤湿度数据（均值 0.188），将湿度 < 0.22 的点标记为干旱点，同时标记作物分区边缘及农场四角为边界点，为后续覆盖优先级判断提供依据。

第 2 步：遍历 5 类参数的 648 种组合，为后续优化提供基础。参数空间具体数值见表 6。

表 6 参数空间各参数的具体数值

参数空间	数值					
覆盖容忍度	0%	2%	4%	6%	8%	10%
干旱权重	1.5	2.0	2.5	—	—	—
边界权重	1.2	1.5	1.8	—	—	—
近河权重	0.6	0.8	1.0	—	—	—
储水系数	0.3	0.4	0.5	0.6	—	—

参数的最优解见表 7:

表 7 参数的最优解

核心参数	非全覆盖情形	全覆盖情形
干旱权重	2.0	2.5
边界权重	1.8	1.5
近河权重	0.6	0.8
储水系数	0.3	0.3

第 3 步：分别采用贪心算法进行喷头布局，按权重优先级（干早点 > 边界点 > 近河点）对未覆盖区域的候选点加权，每次选择覆盖高权重区域最多的位置布设喷头，同时校验新喷头与已有喷头的间距是否 $\geq 15m$ ，不满足则尝试次优候选点。

对于非全覆盖的情况：迭代至未覆盖比例 $\geq$ 容忍度时停止，对于全覆盖时：设置覆盖容忍度为 0%，均在满足约束的前提下减少喷头数量。

第 4 步：计算系统成本，管网成本通过 *Prim* 算法构建最小生成树，得到管网总长度，结合总流量，按公式 $C = 50L^{1.2} + 0.1Q^{1.5}$ 计算；储水罐成本通过 *K-Means* 聚类（远河喷头聚为 2 类，近河喷头取均值）确定位置，再按 7 天应急供水需求计算容积，成本为 $5 \times V$ (5 元 / L)。

第 5 步：进行验证和可视化规划图，通过绘制成本-覆盖度权衡曲线，可直观看到红色星号的位置为全局最优解，如图 3 所示：

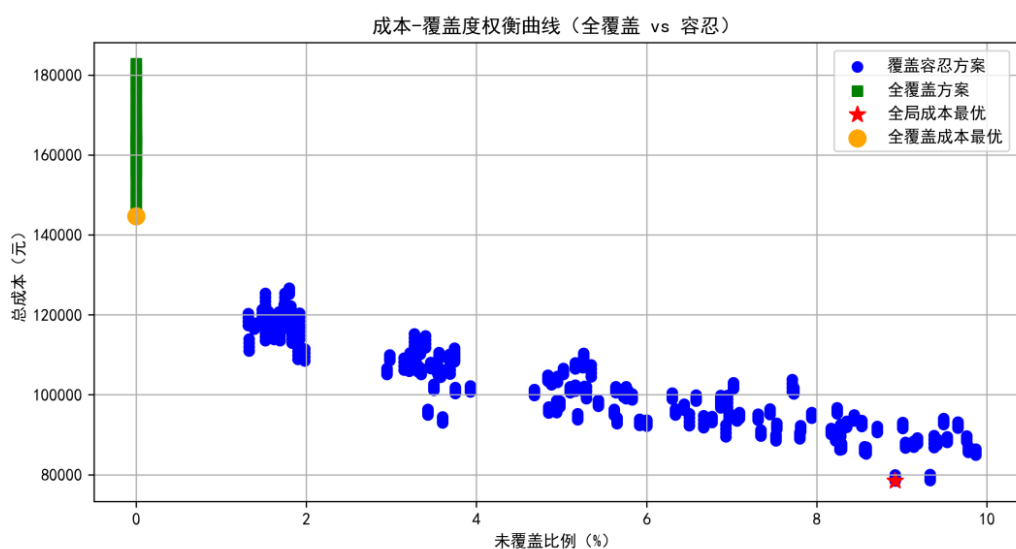


图 3 成本覆盖度权衡曲线

第 6 步：进行参数优化，遍历干旱点权重、边界点权重、近河点权重、储水罐储备系数等参数的 648 种组合，计算每种组合的总造价，筛选出满足两种覆盖情况且总成本最低的方案，最终结果如表 8 所示。

表 8 两种情况总成本最低方案

结果	非全覆盖情形	全覆盖情形
未覆盖比例	8.92%	0.00%
总建设成本	78521.61 元	144725.12 元
储水罐数量	3 个	3 个
储水罐成本	6432.41 元	17812.83 元
储水罐容积	428.83L	1187.52L
喷头数量	16 个	32 个
管网成本	72089.20 元	126912.29 元

6.2.3 结果分析

问题二具体代码见 *Q2.m* 文件，全覆盖与非全覆盖的情形下最优规划如图 4 所示：

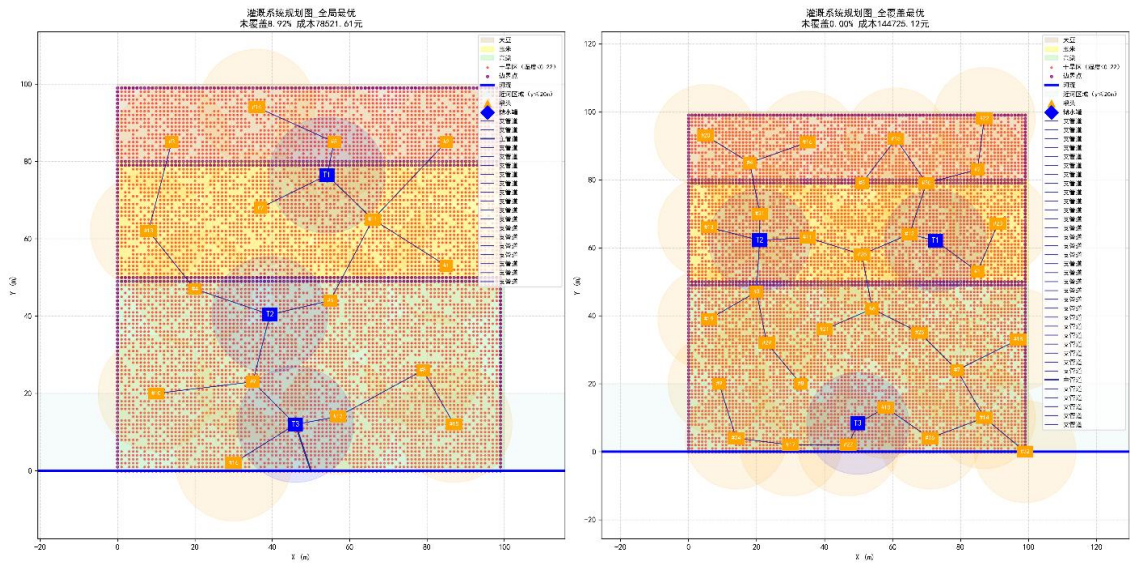


图 4 灌溉系统规划图

我们通过区分全覆盖与非全覆盖情形，结合贪心算法、*K-Means* 聚类及 *Prim* 算法，优化了喷头、储水罐布局与管网，在满足土壤湿度 ≥ 0.22 约束下实现成本优化。这为问题三奠定基础，问题三基于此系统配置，构建线性规划模型，分析旱灾时储水罐应急储备对作物存活面积的影响并延伸至应急场景分析。

6.3 问题三模型的建立与求解

6.3.1 建立单目标线性规划模型

问题三要求在问题二的基础上，储水罐需有应急储备水，旱灾时河流供水量降为问题 2 中引水系统总流量的 80%，需确定能保证的作物存活和正常生长面积，及分析应急储备水源比例与旱灾概率的关系。

针对农业灌溉系统在旱灾条件下的应急供水问题，我们以最大化作物存活面积为目标函数，以供水能力、正常生长条件、面积逻辑等作为约束条件，构建单目标线性规划模型。

目标函数：最大化总存活面积

$$\max \sum_{i=1}^3 S_i$$

其中， S_1, S_2, S_3 分别表示高粱、玉米、大豆三种作物的存活面积。

约束条件：

供水能力约束：

由于存活需水总量小于等于河流供水量与储水罐日均供水能力之和，则有

$$\sum_{i=1}^3 S_i D_i^{min} \leq Q_r + \frac{V}{T}$$

其中， D_i^{min} 表示第 i 种作物的存活需水量； $Q_r = 0.8Q$ 为旱灾时河流的供水量； $V = \sum_{i=1}^3 v_i$ 为储水罐总容积； T 为应急供水天数。

正常生长约束：

由于正常生长需水总量小于等于总供水量，则有：

$$\sum_{i=1}^3 N_i D_i^{norm} \leq Q_r + \frac{V}{T}$$

其中， N_i 表示第 i 种作物的正常生长面积； D_i^{norm} 表示第 i 种作物的正常需水量。

面积大小约束：

对任意 $i \in 1, 2, 3$ ，有

$$0 \leq N_i \leq S_i \leq A_i$$

综上所述，我们建立单目标线性规划模型如下：

$$\begin{aligned} & \max \sum_{i=1}^3 S_i \\ & s.t. \begin{cases} \sum_{i=1}^3 S_i D_i^{min} \leq Q_r + \frac{V}{T} \\ \sum_{i=1}^3 N_i D_i^{norm} \leq Q_r + \frac{V}{T} \\ 0 \leq N_i \leq S_i \leq A_i \end{cases} \end{aligned}$$

6.3.2 单目标线性规划模型的求解

第 1 步：我们首先明确基础数据与系统配置，基于问题二确定的灌溉系统，获取 16 个喷头的布局、3 个储水罐布局，储水罐总容积为 $14448.18 m^3$ ，应急储备水量为 $4334.46 m^3$ ，并输入作物分区信息（高粱 0.5 公顷、玉米 0.3 公顷、大豆 0.2 公顷）、作物需水参数与系统供水参数。同时，初始化作物参数，包括最低存活湿度（高粱 0.22、玉米 0.23、大豆 0.21）、需水量及抗旱系数等。

第 2 步：由式 (19) 计算总供水量，我们计算出各旱灾概率下的供水量（调用 `calc_supply` 函数命令），结果如图 5 所示。

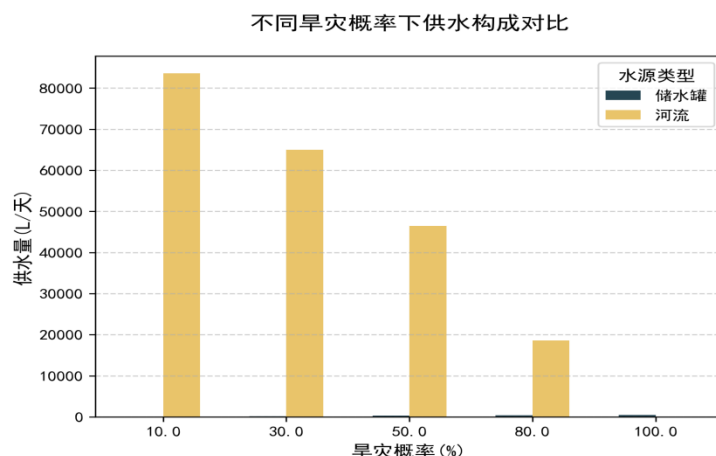


图 5 不同旱灾概率下供水构成对比

第 3 步：我们建立单目标线性规划模型（调用 *crop_optimization* 函数），并用交替方向乘子算法[7]以及问题二得到的水量分配权重和存活判断依据，求解线性规划，最终得到线性规划的函数。进行水量分配与作物状态判断。按作物干旱程度、抗旱能力、经济价值等设定供水优先级，将河流水量与储水罐应急水量分配至各区域；在灌溉后土壤湿度 $\geq$ 最低存活湿度与需水量满足率 $\geq$ 阈值的条件下，判断作物存活面积和正常生长面积，具体结果见表 9。

第 4 步：确定最优应急储备比例。我们利用网格搜索法^[8]进行遍历求解，具体步骤如下：遍历旱灾概率（10%、30%、50%、80%、100%），并对每个旱灾概率在其基准储备比例附近以 0.05 为步长搜索，计算各比例下的总存活面积，最后筛选使总存活面积最大的储备比例，判断是否能保证全部存活，具体数值结果见表 10，可视化结果如图 6 所示。

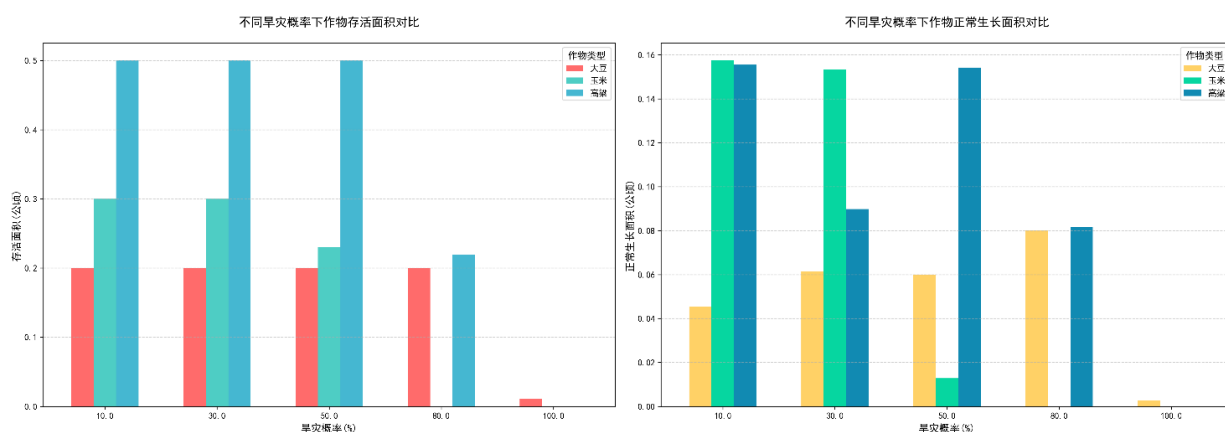


图 6 不同作物正常生长和存活面积对比图

第 5 步：通过最小二乘法，拟合线性关系结果如下：

$$\alpha = 0.9229 \times p + 17.1617$$

其中 α 表示应急储备比例（%）； p 表示旱灾概率（%）。

拟合优度 $R^2=0.9100$ ，因此函数拟合效果较好，线性函数示意图如图 7 所示。

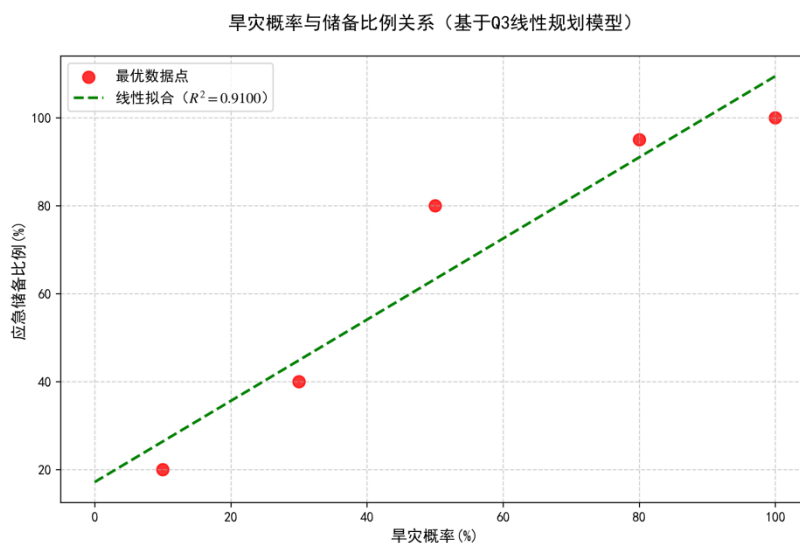


图 7 旱灾概率与储备比例关系图

6.3.3 结果分析

我们利用 *matlab* 代码运行以上过程，具体代码见 *Q3.m* 文件。问题三结果如表 9、表 10 所示。

表 9 作物存活情况表

作物	种植面积（公顷）	存活面积（公顷）	正常生长面积（公顷）
高粱	0.5	0.5	0.1264
玉米	0.3	0.3	0.1202
大豆	0.2	0.2	0.0436

从作物存活与正常生长的面积来看，在河流水流量降至问题二引水系统总流量 80% 的旱灾情况下，高粱、玉米、大豆的存活面积均等于其种植面积，说明该灌溉系统设计能确保所有作物存活；但正常生长面积分别为 0.1264 公顷、0.1202 公顷、0.0436 公顷，远小于种植面积，这可能是由于供水量减少的原因导致的，仅能满足部分作物的正常生长需求。

表 10 应急水源比例与旱灾概率关系表

旱灾概率 (%)	建议应急储备水源比例 (%)	此比例能否保证作物全部存活
10	20	是

30	40	是
50	80	否
80	95	否
100	100	否

从急储备比例与旱灾概率关系来看，两者呈正相关：旱灾概率 10%时，建议应急储备比例 20%，可保证作物全部存活；30%概率时，40%的储备比例仍能保证全部存活；而当概率达到 50%及以上时，即使储备比例提高到 80%、95%、100%，也无法保证全部存活。这一结果符合风险越高需储备越多的逻辑，因此是合理的。该系统在旱灾时能优先保障作物存活，应急储备策略与旱灾风险的匹配性较好，但高概率旱灾下正常生长保障能力有限。

6.4 问题四模型的建立与求解

6.4.1 非线性规划模型

问题四假设各作物成熟期 20 天，用 2021-05-01 至 2021-07-31 数据规划每月灌溉方案，判断之前系统是否满足需求，不满足则修改布线，给出月灌溉情况。

由题中的信息得各个作物的生长时期见表 11。

表 11 作物生长时期

作物	播种期时间段	开花期时间段	成熟期时间段
高粱	5.1-5.20	5.21-7.9	7.10-7.30
玉米	5.1-6.1	6.2-7.21	7.22-8.10
大豆	5.1-6.9	6.10-7.19	7.20-8.8

我们仍以最小化总成本为目标函数，由于目标函数非线性，且问题四要求确保月度需水由河水和储水罐供给，不超河流日供水上限，储水罐动态平衡满足各月需求，土壤湿度在合理范围内，以支撑每月灌溉方案规划。因此应构建以水源分配、河流供水能力、储水罐动态平衡、土壤湿度保障等为约束条件的非线性规划模型。

记 $i \in 1, 2, 3$ 分别表示作物的生长期阶段：{播种期，开花期，成熟期}。作物 c 在月份 m 的月度需水量为：

$$W_{c,m} = A_c \cdot 10^4 \cdot \sum_{i=1}^3 d_{c,i,m} \cdot D_{c,i}$$

其中， A_c 表示作物 c 的种植面积； $d_{c,i,m}$ 表示作物 c 阶段 i 在 m 月份的天数； $D_{c,i}$ 表示作物 c 在阶段 i 的日需水强度（不同土壤湿度下有不同的需水强度）。

目标函数：最小化总成本

$$\min [\sum_{i=1}^{18} (50l_i^{1.2} + 0.1Q_i^{1.5}) + \sum_{j=1}^3 5v_j]$$

约束条件：

水源分配约束：

由于总月度需水量只来源于河水和储水罐，则有

$$\begin{aligned} W_{c,m}^r &= \theta_{c,m} \cdot W_{c,m} \\ W_{c,m}^t &= (1 - \theta_{c,m}) \cdot W_{c,m} \end{aligned}$$

其中， $\theta_{c,m}$ 表示作物 c 在月份 m 的河水供水比例； $W_{c,m}^r$ 与 $W_{c,m}^t$ 分别表示作物 c 在月份 m 的河水与储水罐的实际供水量。

河流供水能力约束：

实际三种作物月度所得水量之和应不超过河流月供水能力，即

$$\sum_c W_{c,m}^r \leq Q_r \cdot MD$$

其中， Q_r 为河流日供水能力， MD 为当月天数。

储水罐动态平衡约束：

第 m 月份末储水罐剩余水量等于第 $m-1$ 月份末储水罐剩余水量减掉第 m 月份储水罐给各种作物的供水量，即

$$V_{t,m} = V_{t,m-1} - \sum_c (1 - \theta_{c,m}) W_{c,m} \geq 0$$

土壤湿度保障约束：

由于实际灌溉量需补足作物需水量与自然降水的差值（若降水不足），即作物 c 在月份 m 的灌溉量应大于等于缺水量则有

$$G_{c,m} \geq \max [0, \sum_{i=1}^3 (W_{c,m} - R_{c,i,m})]$$

其中， $G_{c,m}$ 为月份 m 作物 c 的实际总灌溉量； $R_{c,i,m}$ 表示月份 m 阶段 i 的自然降水量。

综上所述，我们建立非线性规划模型如下：

$$\min [\sum_{i=1}^{18} (50l_i^{1.2} + 0.1Q_i^{1.5}) + \sum_{j=1}^3 5v_j]$$

$$s.t. \begin{cases} \sum_c W_{c,m}^r \leq Q_r \cdot MD \\ V_{t,m} = V_{t,m-1} - \sum_c (1 - \theta_{c,m}) W_{c,m} \geq 0 \\ G_{c,m} \geq \max[0, \sum_{i=1}^3 (W_{c,m} - R_{c,i,m})] \\ \theta_{c,m} \in [0, 1] \end{cases}$$

6.4.2 非线性规划模型的求解

我们沿用问题二的优化结果，未覆盖比例 8.92%，总建设成本 78521.61 元，储水罐数量 3 个与题中其他信息等作为参数进行求解。

第 1 步：我们根据式(25)的需水量公式，遍历各作物的生长阶段与 5-7 月每个月份，计算阶段与月份的重叠天数，并代入公式计算。最后我们得到各作物 5-7 月的需水量如表 12 所示。

表 12 各作物需水量

月份	高粱需水量(L)	玉米需水量(L)	大豆需水量(L)
5 月	819720.0	426254.4	171230.4
6 月	1161270.0	844311.6	289634.4
7 月	1015542.0	866170.8	344282.4

第 2 步：根据式(30)计算作物的实际灌溉需求，通过计算，最终得到由于降水未超过需水量，各月灌溉需求与需水量一致，结果与表 12 一致。

第 3 步：通过代码中的函数遍历应急储备比（0.1、0.3、0.5、0.7），对每个比例，通过“最小需求容积=灌溉需求/(1-储备比)”计算储水罐总容积，采用二次规划算法^[9]（调用 *minimize* 函数）计算总成本，计算各比例下的储水罐总容积、总成本，筛选满足所有约束且成本最低的方案。最后我们得到最优应急储备比：0.2；储水罐总容积：58917.2L；总成本：336934.25 元；所有方案均满足约束。

第 4 步：我们根据优化后的水源分配比例，明确各月各作物的总灌溉量、水源比例及运行状态。检查约束条件，河流月度供水量未超过日供水能力（200000 L/天）与当月天数的乘积，储水罐供水量未超过非应急容积，无需修改布线，完全满足灌溉需求。

6.4.3 结果分析

经过求解，具体代码见 Q4.m 文件，我们得到如下表 13 所示的灌溉安排表：

表 13 灌溉安排表

日期	作物	总灌溉量 (L)	水源比例 (河水/储水罐)	备注 (如是否调整系统布线)
5 月	高粱	819720.0	0.70/0.30	不调整系统布线
	玉米	426254.4	0.60/0.40	不调整系统布线
	大豆	171230.4	0.50/0.50	不调整系统布线
6 月	高粱	1161270.0	0.60/0.40	不调整系统布线
	玉米	844311.6	0.50/0.50	不调整系统布线
	大豆	289634.4	0.40/0.60	不调整系统布线
7 月	高粱	1015542.0	0.50/0.50	不调整系统布线
	玉米	866170.8	0.40/0.60	不调整系统布线
	大豆	344282.4	0.30/0.70	不调整系统布线

结果显示，5-7 月各作物的灌溉方案与生长周期高度匹配：高粱、玉米、大豆的月度灌溉量随生长阶段递增，开花期需水量达峰值，符合作物生长的常识。水源分配上，河水占比从 5 月到 7 月逐渐降低，储水罐占比相应提升，其土壤湿度均值高于最低阈值，无需额外灌溉。现有系统完全满足灌溉需求，河流供水能力覆盖各月河水用量，储水罐动态平衡约束均满足，均无需修改布线。

七、模型的评价

7.1 模型的评价

7.1.1 模型的优点

1. 本文针对四个问题分别构建匹配模型，贴合核心需求。问题一用多元线性回归捕捉土壤湿度与气象因素关系；问题二构建非线性规划模型，分情形优化；问题三以单目标线性规划最大化作物存活面积；问题四的非线性规划模型融入作物生长周期，体现对不同场景的适应性。

2. 对气象数据进行系统预处理，包括文字变量编码、缺失值处理、日数据聚合，为模型提供高质量输入，减少噪声干扰。

3. 本文融合多种算法优势：综合运用多元线性回归、最小二乘法、贪心算法等多种方法，多算法协同提升求解效率与结果合理性。

4. 本文考虑实际约束与场景：纳入大量实际约束，兼顾应急场景与作物生长周期，增强实用价值。

7.1.2 模型的缺点

模型多基于历史数据静态规划，未实时响应短期气象预测或更新数据，难以应对动态不确定性。

7.2 模型的改进

1. 在问题一中的改进方案为采用 *XGBoost* 和 *LSTM* 时间序列模型，引入滞后特征，如前 1-3 小时土壤湿度、累积降水量，捕捉时间依赖性，模拟实际情况；利用 *XGBoost* 的特征重要性筛选关键变量，降低噪声影响。

2. 在灌溉系统布线与储水罐设计上，由于时间有限，我们的现有方案是遗传算法结合参数遍历搜索。在未来，我们可以采用 *NSGA-II* 算法构建多目标优化架构进行改进，同时优化“总成本”与“覆盖度”，并结合 *Steiner* 树算法设计管道网络，在满足喷头间距 $\geq 15\text{m}$ 约束下，降低管道总长 10%-15%，优化空间布局。

7.3 模型的推广

7.3.1 单目标线性规划模型的推广

对问题三中的单目标线性规划模型进行推广，将供水能力约束改为动态约束，加入时间因素，即动态约束为：

$$\sum_{i=1}^3 S_{i,t} D_i^{\min} \leq Q_{r,t} + \frac{V_t}{T}$$

其中， $Q_{r,t}$ 和 V_t 为随时间变化的河流供水量和储水罐容积， $S_{i,t}$ 为阶段 i ，随时间 t 变化而变化的面积。

推广后的模型通过引入时间维度，能够更灵活地应对旱灾概率的动态变化，适用于气候波动较大的区域。这样一来，使得模型具有较强的灵活性和鲁棒性。

7.3.2 添加约束条件

可加入旱灾概率与目标作物总存活面积的约束，通过马尔可夫决策过程将旱灾概率约束

扩展为动态决策问题，根据实时状态（储水量、土壤湿度、气象预测），选择最优动作（如增加储备、减少灌溉）。约束条件如下：

$$P(\sum_{i=1}^3 S_i \geq S_{target}) \geq 1 - \alpha$$

其中， α 为旱灾发生的概率； S_{target} 为目标存活面积。

该约束说明：在旱灾发生概率为 α 的情况下，系统应确保作物总存活面积不低于 S_{target} 的概率至少为 $1 - \alpha$ 。

7.3.3 建立智能灌溉决策支持系统

逐步改变对应的农场面积，从单一农场推广到区域农业水资源管理；加入时间维度，从固定气候条件推广到动态气候场景。并建立如下决策机制：

短期预测：建立时间序列回归模型预测土壤湿度

$$y_t = f(X_t, y_{t-1}) + \varepsilon_t$$

其中， X_t 为题表中给的所有气象数据向量。

长期规划：建立多目标非线性优化模型，目标是最小化成本与水资源浪费，最大化覆盖面积。

应急调整：结合旱灾概率约束和马尔可夫决策^[10]过程，动态调整储水策略。

八、参考文献

- [1] 中国气象局. (2017). 农业干旱监测预报指标及等级标准. 北京：气象出版社.
- [2] Mishra A K, Singh V P. Drought modeling - A review. Journal of Hydrology, 2011, 403: 157-175.
- [3] 张楷卉, 李鹏. 一种基于模糊 C 均值聚类的稀疏数据缺失值填充方法[J]. 黑龙江大学自然科学学报, 2019, 36(6): 750-756. DOI:10.13482/j.issn1001-7011.2019.06.012.
- [4] 王丽娟, 所辉. 时间序列预测中插值法的使用与分析[J]. 电脑编程技巧与维护, 2022(11): 13-15. DOI:10.3969/j.issn.1006-4052.2022.11.004.
- [5] 魏春影. 基于改进 GRU 循环神经网络的土壤湿度预测模型研究[D]. 2023.
- [6] 林蔚蔚. 网格与密铺：六边形千鸟格的研发与应用[J]. 流行色, 2022(11): 88-91. DOI:10.3969/j.issn.1005-555X.2022.11.029.
- [7] 何康. 基于字典学习的热工过程建模及诊断方法研究[D]. 东南大学, 2022.
- [8] Algorithms for Hyper-Parameter Optimization. GridSearch. 2022.

[9] 《农业灌溉系统优化的非线性规划算法及应用》. 2023.

[10] 洪晔, 王宏健, 边信黔. 基于分层马尔可夫决策过程的 AUV 全局路径规划研究[J]. 系统仿真学报, 2008, 20(9):2361-23632367

附录

第一部分：支撑材料

支撑材料	
源代码	1. code_0. <i>m</i> 2. code_1. <i>m</i> 3. code_2. <i>m</i> 4. code_3. <i>m</i> 5. code_4. <i>m</i> 6. code_5. <i>m</i>
数据	数据： 1. cleaned_weather_data.xlsx 2. processed_meteorological_data.xlsx 3. processed_meteorological_data_daily.xlsx 4. weather_daily.xlsx 5. weather_hourly.xlsx 6. 表 4.xlsx 7. 表 5_灌溉安排表.xlsx 8. 该地土壤湿度数据.xlsx 9. 干净数据（带有干旱标注）.xlsx 10. 干净数据（每日）.xlsx 11. 降水量等逐时气象数据.xlsx 12. 降水量等逐时气象数据 1.xlsx

第二部分：代码

附录一：code_0
<pre>clear; clc; close all; disp(' 开始气象数据预处理...');</pre>

```

%% 1. 定义保存路径
save_dir = 'D:\农业灌溉文件夹\';
if ~exist(save_dir, 'dir')
    mkdir(save_dir);
    disp(['已创建保存目录: ', save_dir]);
else
    disp(['保存目录已存在: ', save_dir]);
end

%% 2. 读取 Excel 文件
try
    [~, ~, raw_data] = xlsread('降水量等逐时气象数据 1.xlsx');
    disp('成功读取 Excel 文件');
catch ME
    error('读取 Excel 失败: %s', getReport(ME));
end

% 检查表头行存在性
if size(raw_data, 1) < 2
    error('Excel 格式错误! 至少需包含 2 行 (第 1 行说明+第 2 行表头)');
end

%% 3. 表头清理与目标列匹配
raw_headers = raw_data(2, :); % 原始表头在第 2 行
for i = 1:length(raw_headers)
    if ischar(raw_headers{i})
        raw_headers{i} = strtrim(raw_headers{i}); % 去除首尾空格
    else

```

```

        raw_headers{i} = ''; % 非字符串表头视为空
    end
end

% 定义需提取的 15 列
target_headers = {
    '当地时间 长春市(机场)', 'T', 'Po', 'P', 'Pa', 'U', ...
    'Ff', 'ff3', 'Tn', 'Tx', 'VV', 'Td', 'tR', 'DD', 'RRR'
};
n_target = length(target_headers);

% 动态匹配目标列索引 - 优化匹配逻辑
target_indices = zeros(1, n_target);
missing_cols = {};
for i = 1:n_target
    curr_target = target_headers{i};
    match_flag = false;
    % 精确匹配, 不区分大小写但需完整匹配
    for j = 1:length(raw_headers)
        curr_raw = raw_headers{j};
        if strcmpi(strtrim(curr_raw), strtrim(curr_target))
            target_indices(i) = j;
            match_flag = true;
            break; % 只匹配第一个完全符合的列
        end
    end
end
if ~match_flag
    missing_cols{end+1} = curr_target;
end
end

```

```

end

if ~isempty(missing_cols)
    error(['Excel 表头缺失列："', strjoin(missing_cols, '"', '"'), '"']);
end

disp('15 列均已匹配成功');

%% 4. 提取数据并清理空行
target_indices = uint32(target_indices);
data = raw_data(3:end, target_indices);
data = delete_empty_rows(data);

if isempty(data)
    error('无有效数据！请检查 Excel 第 3 行及以后内容');
end

n_rows = size(data, 1);
disp(['数据提取完成，有效行数：', num2str(n_rows)]);

%% 5. 时间列处理
time_col = 1;
dd_col = 14; % DD 列索引
rr_col = 15; % RRR 列索引
time_num = NaN(n_rows, 1);
for i = 1:n_rows
    time_str = data{i, time_col};
    if ~isempty(time_str) && ischar(time_str)
        time_str = regexprep(time_str, '[^0-9:./ ]', '');
        % 尝试多种日期格式解析
        try

```

```

        if ~isnan(datenum(time_str, 'dd.mm.yyyy HH:MM'))
            time_num(i) = datenum(time_str, 'dd.mm.yyyy HH:MM');
        else
            time_num(i) = datenum(time_str, 'yyyy-mm-dd HH:MM');
        end
    catch
        time_num(i) = NaN;
    end
end
end

valid_idx = ~isnan(time_num);
data = data(valid_idx, :);
time_num = time_num(valid_idx);
n_rows = sum(valid_idx);

if n_rows == 0
    error('无有效时间数据！请检查第1列格式');
end

disp(['时间处理完成，剩余有效行数：', num2str(n_rows)]);

%% 6. 全列填充（按要求处理各列）
filled_data = cell(size(data));

% 首先提取所有数值列，为插值做准备
numeric_data = NaN(n_rows, n_target);
col_types = cell(1, n_target); % 记录每列类型

% 第一步：识别列类型并提取数值

```

```

for col = 1:n_target
    % 跳过时间列，时间列不参与插值
    if col == time_col
        col_types{col} = 'time';
        continue;
    end

    % 强制 DD 列为字符串类型（使用众数填充）
    if col == dd_col
        col_types{col} = 'string';
        continue;
    end

    % 强制 RRR 列为数值类型（使用线性插值）
    if col == rr_col
        col_types{col} = 'numeric';
        col_vals = NaN(n_rows, 1);
        for row = 1:n_rows
            cell_val = data{row, col};
            col_vals(row) = extract_number(cell_val);
        end
        numeric_data(:, col) = col_vals;
        continue;
    end

    % 其他列自动识别类型
    is_numeric = true;
    col_vals = NaN(n_rows, 1);

```

```

for row = 1:n_rows
    cell_val = data{row, col};
    num_val = extract_number(cell_val);
    if isnan(num_val)
        is_numeric = false;
    end
    col_vals(row) = num_val;
end

valid_ratio = sum(~isnan(col_vals)) / n_rows;
if valid_ratio > 0.5
    col_types{col} = 'numeric';
    numeric_data(:, col) = col_vals;
else
    col_types{col} = 'string';
end
end

% 第二步：按列类型填充数据
for col = 1:n_target
    if col == time_col % 时间列直接复制
        for row = 1:n_rows
            filled_data{row, col} = data{row, col};
        end
        continue;
    end

    % DD 列强制使用众数填充
    if col == dd_col

```

```

str_vals = cell(n_rows, 1);
for row = 1:n_rows
    cell_val = data{row, col};
    if ischar(cell_val)
        str_vals{row} = strtrim(cell_val);
    else
        str_vals{row} = '';
    end
end

% 计算众数
valid_strs = str_vals(~cellfun('isempty', str_vals));
mode_str = '';
if ~isempty(valid_strs)
    % 确保所有元素都是字符行向量
    str_array = cellfun(@(x) x(:)', valid_strs, 'UniformOutput',
false);

    % 手动去重
    unique_str = {};
    for i = 1:length(str_array)
        curr_str = str_array{i};
        is_duplicate = false;
        for j = 1:length(unique_str)
            if strcmp(curr_str, unique_str{j})
                is_duplicate = true;
                break;
            end
        end
        if ~is_duplicate

```

```

        unique_str{end+1} = curr_str;
    end
end

% 计算每个唯一字符串的出现次数
str_count = zeros(length(unique_str), 1);
for k = 1:length(unique_str)
    str_count(k) = sum(strcmp(str_array, unique_str{k}));
end
[~, mode_idx] = max(str_count);
mode_str = unique_str{mode_idx};
else
    mode_str = '未知';
end

% 填充空值
for row = 1:n_rows
    if isempty(str_vals{row})
        filled_data{row, col} = mode_str;
    else
        filled_data{row, col} = str_vals{row};
    end
end
disp(['列 ', num2str(col), ' (', target_headers{col}, ') 众数填充完成
']);

continue;
end

% RRR 列强制使用线性插值

```

```

if col == rr_col
    vals = numeric_data(:, col);
    time_vals = time_num;

    valid_mask = ~isnan(vals);
    if sum(valid_mask) < 2
        col_mean = mean(vals(valid_mask));
        if isnan(col_mean)
            col_mean = 0;
        end
        filled_vals = repmat(col_mean, n_rows, 1);
    else
        filled_vals = interp1(time_vals(valid_mask),
vals(valid_mask), ...
                                time_vals, 'linear', 'extrap');
    end

    for row = 1:n_rows
        filled_data{row, col} = filled_vals(row);
    end
    disp(['列 ', num2str(col), ' (', target_headers{col}, ') 线性插值填充
完成']);

    continue;
end

% 处理其他列
if strcmp(col_types{col}, 'numeric')
    vals = numeric_data(:, col);
    time_vals = time_num;

```

```

valid_mask = ~isnan(vals);
if sum(valid_mask) < 2
    col_mean = mean(vals(valid_mask));
    if isnan(col_mean)
        col_mean = 0;
    end
    filled_vals = repmat(col_mean, n_rows, 1);
else
    filled_vals = interp1(time_vals(valid_mask),
vals(valid_mask), ...
                                time_vals, 'linear', 'extrap');
end

for row = 1:n_rows
    filled_data{row, col} = filled_vals(row);
end
disp(['列 ', num2str(col), ' (', target_headers{col}, ') 线性插值填充
完成']);
else
    str_vals = cell(n_rows, 1);
    for row = 1:n_rows
        cell_val = data{row, col};
        if ischar(cell_val)
            str_vals{row} = strtrim(cell_val);
        else
            str_vals{row} = '';
        end
    end
end
end

```

```

% 计算众数
valid_strs = str_vals(~cellfun('isempty', str_vals));
mode_str = '';
if ~isempty(valid_strs)
    % 确保所有元素都是字符行向量
    str_array = cellfun(@(x) x(:)', valid_strs, 'UniformOutput',
false);

    % 手动去重
    unique_str = {};
    for i = 1:length(str_array)
        curr_str = str_array{i};
        is_duplicate = false;
        for j = 1:length(unique_str)
            if strcmp(curr_str, unique_str{j})
                is_duplicate = true;
                break;
            end
        end
        if ~is_duplicate
            unique_str{end+1} = curr_str;
        end
    end

    % 计算每个唯一字符串的出现次数
    str_count = zeros(length(unique_str), 1);
    for k = 1:length(unique_str)
        str_count(k) = sum(strcmp(str_array, unique_str{k}));
    end
end

```

```

        [~, mode_idx] = max(str_count);
        mode_str = unique_str{mode_idx};
    else
        mode_str = '未知';
    end

    for row = 1:n_rows
        if isempty(str_vals{row})
            filled_data{row, col} = mode_str;
        else
            filled_data{row, col} = str_vals{row};
        end
    end

    disp(['列 ', num2str(col), ' (', target_headers{col}, ') 众数填充完成
    ']);

    end

end

disp('所有 15 列数据填充完成（无空值）');

%% 7. DD 列风向编码
for row = 1:n_rows
    dd_val = filled_data{row, dd_col};
    % 转换为字符串
    if ischar(dd_val)
        dd_str = lower(dd_val);
    else
        dd_str = lower(num2str(dd_val));
    end
end

```

```

% 风向编码
if ~isempty(strfind(dd_str, '东北'))
    filled_data{row, dd_col} = 5;
elseif ~isempty(strfind(dd_str, '东南'))
    filled_data{row, dd_col} = 6;
elseif ~isempty(strfind(dd_str, '西北'))
    filled_data{row, dd_col} = 7;
elseif ~isempty(strfind(dd_str, '西南'))
    filled_data{row, dd_col} = 8;
elseif ~isempty(strfind(dd_str, '东'))
    filled_data{row, dd_col} = 1;
elseif ~isempty(strfind(dd_str, '南'))
    filled_data{row, dd_col} = 2;
elseif ~isempty(strfind(dd_str, '西'))
    filled_data{row, dd_col} = 3;
elseif ~isempty(strfind(dd_str, '北'))
    filled_data{row, dd_col} = 4;
elseif ~isempty(strfind(dd_str, '无风'))
    filled_data{row, dd_col} = 0;
else
    filled_data{row, dd_col} = 9;
end
end
disp('DD 列风向编码完成');

%% 8. RRR 列降水量处理
for row = 1:n_rows
    rr_val = filled_data{row, rr_col};
    rr_str = lower(num2str(rr_val));

```

```

        if ~isempty(strfind(rr_str, '无降水')) || ~isempty(strfind(rr_str, '痕迹
    '))

        filled_data{row, rr_col} = 0;
    else
        num_match = regexp(rr_str, '\d+\.?\\d*', 'match', 'once');
        if ~isempty(num_match)
            rr_num = str2double(num_match);
            if rr_num < 0
                filled_data{row, rr_col} = 0;
            else
                filled_data{row, rr_col} = rr_num;
            end
        else
            filled_data{row, rr_col} = 0;
        end
    end
end
end
disp(' RRR 列降水量处理完成');

%% 9. 保存逐时数据
hourly_data = [target_headers; filled_data];
hourly_path = [save_dir, 'weather_hourly.csv'];

fid_hourly = fopen(hourly_path, 'w');
if fid_hourly == -1
    error(' 无法创建逐时数据文件！请手动检查路径权限');
end

```

```

% 写入数据
for row = 1:size(hourly_data, 1)
    curr_row = hourly_data(row, :);
    for col = 1:n_target
        cell_val = curr_row{col};
        if isnumeric(cell_val)
            fprintf(fid_hourly, '%g', cell_val);
        else
            fprintf(fid_hourly, '"%s"', cell_val);
        end
        if col < n_target
            fprintf(fid_hourly, ',');
        end
    end
    fprintf(fid_hourly, '\n');
end
fclose(fid_hourly);
disp(['逐时数据已保存至: ', hourly_path]);

%% 10. 逐日聚合与保存
% 提取日期
dates = cell(n_rows, 1);
for i = 1:n_rows
    [y, m, d] = datevec(time_num(i));
    dates{i} = sprintf('%04d-%02d-%02d', y, m, d);
end
% 日期去重
unique_dates = {};
for i = 1:length(dates)

```

```

curr_date = dates{i};
is_duplicate = false;
for j = 1:length(unique_dates)
    if strcmp(curr_date, unique_dates{j})
        is_duplicate = true;
        break;
    end
end
if ~is_duplicate
    unique_dates{end+1} = curr_date;
end
end
n_days = length(unique_dates);

% 初始化逐日数据：(天数+1)行（表头+数据）、(15+1)列（日期+15个目标列）
daily_data = cell(n_days + 1, n_target + 1);

% 设置表头行
daily_data(1, :) = [{'日期'}, target_headers];

% 逐天聚合
for day_idx = 1:n_days
    curr_date = unique_dates{day_idx};
    daily_data(day_idx + 1, 1) = {curr_date}; % 第1列存日期
    day_mask = strcmp(dates, curr_date);
    day_rows = filled_data(day_mask, :);
    n_day_rows = sum(day_mask);

    for col = 1:n_target

```

```

% 检查当前列是否为 RRR 降水量列或 DD 列
is_rrr_col = (col == rr_col);
is_dd_col = (col == dd_col);

if is_dd_col
    % DD 列使用众数填充 - 低版本兼容
    str_vals = cell(n_day_rows, 1);
    for row = 1:n_day_rows
        str_vals{row} = day_rows{row, col};
    end
    valid_strs = str_vals(~cellfun('isempty', str_vals));
    if ~isempty(valid_strs)
        % 确保所有元素都是字符行向量
        str_array = cellfun(@(x) x(:)', valid_strs, 'UniformOutput',
false);

        % 手动去重
        unique_str = {};
        for i = 1:length(str_array)
            curr_str = str_array{i};
            is_duplicate = false;
            for j = 1:length(unique_str)
                if strcmp(curr_str, unique_str{j})
                    is_duplicate = true;
                    break;
                end
            end
            if ~is_duplicate
                unique_str{end+1} = curr_str;
            end
        end
    end
end

```

```

        end

        % 计算每个唯一字符串的出现次数
        str_count = zeros(length(unique_str), 1);
        for k = 1:length(unique_str)
            str_count(k) = sum(strcmp(str_array, unique_str{k}));
        end
        [~, mode_idx] = max(str_count);
        daily_data(day_idx + 1, col + 1) = {unique_str{mode_idx}};
    else
        daily_data(day_idx + 1, col + 1) = {' '};
    end
elseif is_rrr_col
    % RRR 列进行累加
    col_vals = NaN(n_day_rows, 1);
    for row = 1:n_day_rows
        cell_val = day_rows{row, col};
        if isnumeric(cell_val)
            col_vals(row) = cell_val;
        else
            col_vals(row) = 0;
        end
    end
    daily_data(day_idx + 1, col + 1) =
{sum(col_vals(~isnan(col_vals)))};
else
    % 其他数值列计算均值
    is_numeric_col = true;
    col_vals = NaN(n_day_rows, 1);

```

```

        for row = 1:n_day_rows
            cell_val = day_rows{row, col};
            if isnumeric(cell_val)
                col_vals(row) = cell_val;
            else
                is_numeric_col = false;
                break;
            end
        end
    end

    if is_numeric_col
        if n_day_rows > 0
            daily_data(day_idx + 1, col + 1) =
{mean(col_vals(~isnan(col_vals)))});
        else
            daily_data(day_idx + 1, col + 1) = {0};
        end
    else
        str_vals = cell(n_day_rows, 1);
        for row = 1:n_day_rows
            str_vals{row} = day_rows{row, col};
        end
        valid_strs = str_vals(~cellfun('isempty', str_vals));
        if ~isempty(valid_strs)
            % 确保所有元素都是字符行向量
            str_array = cellfun(@(x) x(:)', valid_strs,
'UniformOutput', false);

            % 手动去重
            unique_str = {};

```

```

        for i = 1:length(str_array)
            curr_str = str_array{i};
            is_duplicate = false;
            for j = 1:length(unique_str)
                if strcmp(curr_str, unique_str{j})
                    is_duplicate = true;
                    break;
                end
            end
            if ~is_duplicate
                unique_str{end+1} = curr_str;
            end
        end

        % 计算每个唯一字符串的出现次数
        str_count = zeros(length(unique_str), 1);
        for k = 1:length(unique_str)
            str_count(k) = sum(strcmp(str_array, unique_str{k}));
        end
        [~, mode_idx] = max(str_count);
        daily_data(day_idx + 1, col + 1) =
{unique_str{mode_idx}};

        else
            daily_data(day_idx + 1, col + 1) = {' '};
        end
    end
end
end
end
end

```

```

disp('逐日聚合完成 (RRR 列累加, DD 列众数, 其他列均值)');

% 保存逐日数据
daily_path = [save_dir, 'weather_daily.csv'];
fid_daily = fopen(daily_path, 'w');
if fid_daily == -1
    error('无法创建逐日数据文件! 请手动检查路径权限');
end

for row = 1:size(daily_data, 1)
    curr_row = daily_data(row, :);
    for col = 1:n_target + 1
        cell_val = curr_row{col};
        if isnumeric(cell_val)
            fprintf(fid_daily, '%g', cell_val);
        else
            fprintf(fid_daily, '"%s"', cell_val);
        end
        if col < n_target + 1
            fprintf(fid_daily, ',');
        end
    end
    fprintf(fid_daily, '\n');
end
fclose(fid_daily);
disp(['逐日数据已保存至: ', daily_path]);

disp('\n 预处理完成! 兼容低版本 MATLAB, 双文件保存成功');

```

```

%% 子函数：删除全空行

function cleaned_data = delete_empty_rows(raw_data)

    if isempty(raw_data)

        cleaned_data = raw_data;

        return;

    end

    [n_rows, n_cols] = size(raw_data);

    is_empty_row = false(n_rows, 1);

    for row = 1:n_rows

        empty_flag = true;

        for col = 1:n_cols

            cell_val = raw_data{row, col};

            if ~isempty(cell_val) && ~(isnumeric(cell_val) &&
isnan(cell_val))

                empty_flag = false;

                break;

            end

        end

        is_empty_row(row) = empty_flag;

    end

    cleaned_data = raw_data(~is_empty_row, :);

end

%% 子函数：提取数值

function num = extract_number(cell_val)

    num = NaN;

    if isempty(cell_val)

        return;

    end

```

```

end
if isnumeric(cell_val)
    num = cell_val;
    return;
end
if ischar(cell_val)
    clean_str = regexp(cell_val, '[^0-9.-]', '');
    if ~isempty(clean_str)
        num = str2double(clean_str);
    end
end
end
end

```

附录二：code_1

```

%皮尔逊相关 + 多元线性回归
clear; clc;

% 读取 Excel（逐日数据文件）
[num,txt,row] = xlsread('processed_meteorological_data_daily.xlsx');

% 数据部分（去掉表头）
data = num;    % 假设 Excel 中表头在第 1 行，从第 2 行开始是数值

% X = 第 2 到 15 列，y = 第 16 列
X = data(:,1:14);
y = data(:,15);

% 去除缺失值

```

```

mask = all(~isnan(X),2) & ~isnan(y);
X = X(mask,:);
y = y(mask,:);

% 计算皮尔逊相关系数
R = corr(X,y); % 每个特征与 y 的相关性
disp('===== 各特征与 y 的皮尔逊相关系数 =====');
disp(R);

% 多元线性回归
X_with_bias = [ones(size(X,1),1), X]; % 增加常数项
b = X_with_bias \ y; % 最小二乘解

% 输出回归方程
fprintf('\n===== 多元线性回归方程 =====\n');
fprintf('y = %.4f', b(1));
for i=2:length(b)
    if b(i) >= 0
        fprintf(' + %.4f*x%d', b(i), i-1);
    else
        fprintf(' - %.4f*x%d', abs(b(i)), i-1);
    end
end
fprintf('\n');

```

附录三：code_2

```

% ===== 问题二：修复索引错误版本 =====

clc; close all;

```

```
%% ---- 1) 基本参数 ----
```

```
FARM_SIZE = 100;          % 农场 100x100 m
```

```
SPR_R = 15;               % 喷头覆盖半径(m)
```

```
MIN_SP = 15;              % 喷头最小间距(m)
```

```
MIN_SM = 0.22;            % 存活最低湿度阈值
```

```
RIVER_PT = [50, 0];       % 河流参考点（底边中点）
```

```
RIVER_Y = 20;             % 近河区域:  $y \leq 20$ 
```

```
% 三块作物区: [x1 x2 y1 y2]（左闭右开）
```

```
CROPS = {[0 100 80 100], [0 100 50 80], [0 100 0 50]};
```

```
% ---- 2021-07 土壤湿度（日序列，31 天） ----
```

```
MOIST_VEC = [ ...
```

```
    0.208, 0.217, 0.212, 0.210, 0.213, 0.209, 0.243, 0.250, 0.257, 0.247,  
0.239, ...
```

```
    0.232, 0.228, 0.211, 0.200, 0.192, 0.185, 0.172, 0.164, 0.155, 0.150,  
0.144, ...
```

```
    0.144, 0.136, 0.131, 0.130, 0.129, 0.128, 0.124, 0.126, 0.255]';
```

```
%% ---- 2) 作物点 + 湿度（循环分配）+ 干旱/边界标记 ----
```

```
[PTS, SM, IS_EDGE] = build_crop_points(CROPS, MOIST_VEC);
```

```
% 确保 IS_EDGE 是逻辑数组且维度正确
```

```
IS_EDGE = logical(IS_EDGE);
```

```
DROUGHT = logical(SM < MIN_SM); % 确保是逻辑数组
```

```
% 验证数据一致性
```

```
if length(IS_EDGE) ~= size(PTS, 1)
```

```
    error('IS_EDGE 与 PTS 维度不匹配');
```

```

end

if length(DROUGHT) ~= size(PTS,1)
    error('DROUGHT 与 PTS 维度不匹配');
end

fprintf('作物点总数: %d | 干早点: %d | 边界点: %d\n', ...
        size(PTS,1), sum(DROUGHT), sum(IS_EDGE));

%% ---- 3) 参数搜索 (贪心布点 → 罐位 → 管网成本) ----
CT = [0 0.02 0.05];    % 覆盖容忍度
DW = [1.5 2.0];        % 干旱权重
EW = [1.2 1.5];        % 边界权重
NRW = [0.8 1.0];       % 近河权重
TR = [0.3 0.5];        % 储水罐系数

best.cost = inf; best.spr=[]; best.tc=[]; best.cov=0; best.pc=0;
best.tcost=0; best.params=[];

for i1=1:length(CT)
    ct = CT(i1);
    for i2=1:length(DW)
        dw = DW(i2);
        for i3=1:length(EW)
            ew = EW(i3);
            for i4=1:length(NRW)
                nrw = NRW(i4);
                for i5=1:length(TR)
                    tr = TR(i5);

```

```

        [spr, cov_ratio] = layout_greedy(PTS, DROUGHT, IS_EDGE,
ct, dw, ew, nrw, SPR_R, MIN_SP, RIVER_Y);

        if cov_ratio < 1-ct-1e-9, continue; end

        [tanks, tank_cost] = place_tanks_simple(spr, RIVER_Y,
SPR_R, tr);

        if isempty(tanks), continue; end

        [pipe_cost, ~] = pipe_cost_mst(RIVER_PT, spr, tanks,
SPR_R);

        total_cost = tank_cost + pipe_cost;

        if total_cost < best.cost
            best.cost    = total_cost;
            best.spr     = spr;
            best.tc       = tanks;
            best.cov      = cov_ratio;
            best.pc       = pipe_cost;
            best.tcost    = tank_cost;
            best.params = [ct, dw, ew, nrw, tr];
        end
    end
end
end
end

end

%% ---- 4) 输出结果（含坐标）与简图 ----

if isempty(best.spr)

```

```

fprintf(' ⚠ 未找到满足覆盖要求的方案，请放宽参数范围。\\n');
else
    fprintf('\\n===== 最优方案（成本最低） =====\\n');
    fprintf('未覆盖比例: %.2f%%\\n', (1-best.cov)*100);
    fprintf('总建设成本: %.2f 元 （管网: %.2f, 储水罐: %.2f) \\n', best.cost,
best.pc, best.tcost);

    fprintf('喷头数量: %d 个 | 储水罐数量: %d 个\\n', size(best.spr,1),
size(best.tc,1));

    fprintf('参数: CT=%.2f, DW=%.1f, EW=%.1f, NRW=%.1f, TR=%.1f\\n', ...

best.params(1),best.params(2),best.params(3),best.params(4),best.params(5));


% 打印喷头坐标
fprintf('\\n-- 喷头坐标 (x,y) --\\n');
for i=1:size(best.spr,1)
    fprintf('S%03d: (%.2f, %.2f)\\n', i, best.spr(i,1), best.spr(i,2));
end

% 打印储水罐坐标
fprintf('\\n-- 储水罐坐标 (x,y) --\\n');
for i=1:size(best.tc,1)
    fprintf('T%02d: (%.2f, %.2f)\\n', i, best.tc(i,1), best.tc(i,2));
end

% 简图
plot_layout_simple(best, SPR_R, RIVER_Y);
end

```

```

%% ===== 子函数 =====

function [PTS, SM, IS_EDGE] = build_crop_points(CROPS, MOIST_VEC)
% 生成作物点（1m 网格），并按 31 天湿度序列循环分配；同时标记边界
PTS = []; SM = []; IS_EDGE = [];
for i=1:length(CROPS)
    r = CROPS{i};
    % 确保网格生成正确（左闭右开）
    x = r(1):r(2)-1;
    y = r(3):r(4)-1;
    [x_grid, y_grid] = meshgrid(x, y);
    p = [x_grid(:), y_grid(:)];
    npt = size(p, 1);

    % 按 31 天湿度序列循环分配（确定性、可复现）
    sm = zeros(npt, 1);
    for k=1:npt
        idx = mod(k-1, length(MOIST_VEC)) + 1; % 更鲁棒的循环索引
        sm(k) = MOIST_VEC(idx);
    end

    % 作物区边界标记（确保生成逻辑值）
    be = (p(:, 1) == r(1)) | (p(:, 1) == r(2)-1) | ...
        (p(:, 2) == r(3)) | (p(:, 2) == r(4)-1);
    be = logical(be); % 显式转换为逻辑数组

    % 拼接时保持类型一致
    PTS = [PTS; p];
    SM = [SM; sm];
end

```

```

        IS_EDGE = [IS_EDGE; be];
    end
    % 最终确认 IS_EDGE 是逻辑数组
    IS_EDGE = logical(IS_EDGE);
end

function d = dist_to_point(A, c)
% A: n×2, c: 1×2
d = sqrt( (A(:,1)-c(1)).^2 + (A(:,2)-c(2)).^2 );
end

function C = simple_kmeans2(P)
% 极简 2-means（无工具箱）：按 x 最小/最大初始化，迭代 10 次或提前收敛
n = size(P,1);
if n==0, C=[]; return; end
if n<=2, C=P; return; end
[~, ixmin] = min(P(:,1)); [~, ixmax] = max(P(:,1));
c1 = P(ixmin,:); c2 = P(ixmax,:);
for it=1:10
    d1 = dist_to_point(P, c1);
    d2 = dist_to_point(P, c2);
    g1 = d1<=d2; g2 = ~g1;
    if ~any(g1) || ~any(g2), break; end
    nc1 = mean(P(g1,:),1); nc2 = mean(P(g2,:),1);
    if max(abs(nc1-c1))+max(abs(nc2-c2)) < 1e-6, c1=nc1; c2=nc2; break; end
    c1=nc1; c2=nc2;
end
C = [c1; c2];
end

```

```

function [spr, cov_ratio] = layout_greedy(PTS, DROUGHT, IS_EDGE, ct, dw, ew,
nrw, R, S, river_y)
% 贪心布点（权重：干旱>边界>近河）
n = size(PTS,1);
W = ones(n,1); % 权重向量初始化

% 修复索引错误：确保所有索引都是逻辑数组且维度匹配
if length(IS_EDGE) ~= n
    error('IS_EDGE 长度与作物点数量不匹配');
end
if length(DROUGHT) ~= n
    error('DROUGHT 长度与作物点数量不匹配');
end

% 应用权重（使用逻辑索引）
W(IS_EDGE) = W(IS_EDGE) * ew;
W(DROUGHT) = W(DROUGHT) * dw;
W(PTS(:,2)<=river_y) = W(PTS(:,2)<=river_y) * nrw;

covered = false(n,1);
spr = [];
while mean(covered) < 1-ct-1e-9
    idxs = find(~covered);
    if isempty(idxs), break; end
    best_sc = -inf; best_c = [];
    % 逐候选计算覆盖得分
    for ii=1:length(idxs)
        c = PTS(idxs(ii),:);

```

```

        d = dist_to_point(PTS, c);
        m = (~covered) & (d<=R);
        sc = sum(W(m));
        if sc > best_sc
            best_sc = sc; best_c = c;
        end
    end
end
% 与已有喷头的最小间距约束
ok = true;
if ~isempty(spr)
    dd = dist_to_point(spr, best_c);
    ok = all(dd >= S-1e-9);
end
if ~ok, break; end

spr = [spr; best_c];
d_all = dist_to_point(PTS, best_c);
covered = covered | (d_all<=R);
end
cov_ratio = mean(covered);
end

function [tanks, tank_cost] = place_tanks_simple(spr, river_y, R, tank_ratio)
% 远河(>river_y)做简易 2-means, 近河(<=river_y)取均值
near = spr(spr(:,2)<=river_y, :);
far = spr(spr(:,2)> river_y, :);

C = [];
if ~isempty(far)

```

```

        C = simple_kmeans2(far);
    end
    if ~isempty(near)
        C = [C; mean(near,1)];
    end
    tanks = C;

    A = pi*R*R;           % 单喷头覆盖面积
    tank_cost = 5 * size(tanks,1) * (0.2*A*7) * tank_ratio; % 简化费用
end

function [pipe_cost, total_len] = pipe_cost_mst(RIVER_PT, spr, tanks, R)
% 手写 Prim + 手写距离矩阵
nodes = [RIVER_PT; tanks; spr];
n = size(nodes,1);
if n<=1, pipe_cost=0; total_len=0; return; end

D = zeros(n,n);
for i=1:n
    for j=i+1:n
        dij = sqrt( (nodes(i,1)-nodes(j,1))^2 + (nodes(i,2)-nodes(j,2))^2 );
        D(i,j)=dij; D(j,i)=dij;
    end
end

in = false(n,1); key = inf(n,1); key(1)=0; total_len=0;
for k=1:n
    best = inf; u=-1;
    for i=1:n

```

```

        if ~in(i) && key(i)<best
            best = key(i); u=i;
        end
    end
    if u==-1, break; end
    in(u)=true; total_len = total_len + key(u);
    for v=1:n
        if ~in(v) && D(u,v) < key(v)
            key(v) = D(u,v);
        end
    end
end
end

Q = 0.2 * pi*R*R * size(spr,1);          % 简化总流量（所有喷头）
pipe_cost = 50*(total_len^1.2) + 0.1*(Q^1.5);
end

function plot_layout_simple(best, SPR_R, RIVER_Y)
% 简单示意图
figure('Position',[80 80 780 780]); hold on; grid on; axis equal;
xlim([-5 105]); ylim([-5 105]);
% 作物区（上：80-100，中：50-80，下：0-50）
fill([0 100 100 0],[80 80 100 100],[0.8 1.0 0.8],'EdgeColor','none');
fill([0 100 100 0],[50 50 80 80],[1.0 1.0 0.8],'EdgeColor','none');
fill([0 100 100 0],[0 0 50 50],[0.9 0.8 0.6],'EdgeColor','none');
% 近河带 + 河流
fill([0 100 100 0],[0 0 RIVER_Y RIVER_Y],[0.7 0.8
1], 'FaceAlpha',0.15,'EdgeColor','none');
plot([0 100],[0 0],'b','LineWidth',2);

```

```

% 喷头 + 覆盖圆
for i=1:size(best.spr,1)
    s = best.spr(i,:);
    plot(s(1),s(2),'o','MarkerSize',6,'MarkerFaceColor',[1 0.5
0], 'MarkerEdgeColor',[0.8 0.3 0]);
    rectangle('Position',[s(1)-SPR_R, s(2)-SPR_R, 2*SPR_R, 2*SPR_R], ...
        'Curvature',[1 1], 'EdgeColor',[1 0.7 0], 'LineStyle',':');
end
% 储水罐 + 覆盖圈
for i=1:size(best.tc,1)
    t = best.tc(i,:);
    plot(t(1),t(2),'d','MarkerSize',8,'MarkerFaceColor',[0 0.5
1], 'MarkerEdgeColor',[0 0.2 0.6]);
    rectangle('Position',[t(1)-SPR_R, t(2)-SPR_R, 2*SPR_R, 2*SPR_R], ...
        'Curvature',[1 1], 'EdgeColor',[0 0.5 1], 'LineStyle','-');
end
title(sprintf('最优布局 | 未覆盖 %.2f%% | 成本 %.2f 元', (1-best.cov)*100,
best.cost));
hold off;
end

```

附录四：code_3

```

clear; clc;

%% ----- 1) 全局参数 -----
SYSTEM.TotalTankVol = 41850; % 总储水容积(L)

```

```

SYSTEM.DesignFlow      = 10000; % 设计日流量(L/天)
SYSTEM.EmergencyDays   = 31;    % 应急天数（题目要求）

% 三种作物（表1）：面积(公顷)，存活需水(L/m^2)，正常需水(L/m^2)
CROPS = struct( ...
    'Name', {'高粱','玉米','大豆'}, ...
    'Area', {0.5, 0.3, 0.2}, ...
    'Dmin', {5, 6, 4}, ...      % 存活最低需水
    'Dnorm',{10,12,8} ...      % 正常较好生长需水
);

% 输出目录
out_dir = '问题3 优化结果';
if ~exist(out_dir,'dir'), mkdir(out_dir); end

%% ----- 2) 表3: 80%河水 + 指定储备比例 -----
riverRatio_80 = 0.8; % 河流水仅有 80%
reserveRatio_80 = 0.7; % 应急储备比例 (0~1)

% 求解（优先使用 linprog；若无则近似分配）
res80 = solve_crop_allocation(riverRatio_80, reserveRatio_80, CROPS, SYSTEM);

% 打印并保存表3
fprintf('表3 作物存活情况表（河水 80%%，储备比例%.0f%%）：\n',
reserveRatio_80*100);

fprintf('作物\t种植面积(公顷)\t存活面积(公顷)\t正常生长面积(公顷)\n');
for i = 1:numel(CROPS)
    fprintf('%s\t%.1f\t%.4f\t%.4f\n', ...
        CROPS(i).Name, CROPS(i).Area, res80.S(i), res80.N(i));

```

```

end

% 最基础的 fopen 语法（仅文件名和打开模式）
fid = fopen(fullfile(out_dir,'表 3.txt'),'w'); % 移除所有编码参数
fprintf(fid,'表 3 作物存活情况表（河水 80%%, 储备比例%.0f%%）：\n',
reserveRatio_80*100);

fprintf(fid,'作物\t种植面积(公顷)\t存活面积(公顷)\t正常生长面积(公顷)\n');
for i = 1:numel(CROPS)
    fprintf(fid,'%s\t%.1f\t%.4f\t%.4f\n', ...
        CROPS(i).Name, CROPS(i).Area, res80.S(i), res80.N(i));
end
fclose(fid);

%% ----- 3) 表 4: 旱灾概率 vs 建议储备比例 -----
% 旱灾概率: 10/30/50/80/100, 对应河水比例=1-概率
droughtProb = [0.1 0.3 0.5 0.8 1.0];
baseRatio    = [0.2 0.4 0.6 0.8 1.0]; % 以这些为中心做局部搜索

% 结果表（cell, 便于保存）
tab4 = cell(numel(droughtProb)+1,3);
tab4(1,:) = {'旱灾概率(%)','建议应急储备比例(%)','此比例能否保证作物全部存活'};

bestRatio_arr = zeros(numel(droughtProb),1);
fullSurvive_arr = false(numel(droughtProb),1);

for i = 1:numel(droughtProb)
    pr = droughtProb(i);
    rvr = 1 - pr; % 河水比例

```

```

br = baseRatio(i);          % 基准储备比例

% 在 [br-0.2, br+0.2] 与 [0,1] 的交集中, 以 0.05 为步长搜索
lo = max(0, br-0.2);
hi = min(1.0, br+0.2);
cand = lo:0.05:hi;
if isempty(cand), cand = br; end

% 选择能使总存活面积最大的储备比例 (若有并列, 取较小比例)
bestS = -1; bestR = cand(1); allArea = sum([CROPS.Area]);
for rr = cand
    rr_res = solve_crop_allocation(rvr, rr, CROPS, SYSTEM);
    sumS = sum(rr_res.S);
    if sumS > bestS + 1e-9 || (abs(sumS-bestS)<=1e-9 && rr < bestR)
        bestS = sumS; bestR = rr;
    end
end
bestRatio_arr(i) = bestR;
fullSurvive_arr(i) = (abs(bestS - allArea) <= 1e-6);

tab4{i+1,1} = sprintf('%.0f', pr*100);
tab4{i+1,2} = sprintf('%.1f', bestR*100);
tab4{i+1,3} = ternary_str(fullSurvive_arr(i),'是','否');
end

% 打印并保存表 4
fprintf('\n 表 4 应急水源比例与旱灾概率关系表: \n');
for r = 1:size(tab4,1)
    fprintf('%s\t%s\t%s\n', tab4{r,1}, tab4{r,2}, tab4{r,3});
end

```

```

fid = fopen(fullfile(out_dir,'表 4. txt'),'w');
for r = 1:size(tab4,1)
    fprintf(fid,'%s\t%s\t%s\n', tab4{r,1}, tab4{r,2}, tab4{r,3});
end
fclose(fid);

%% ----- 4) 拟合: 储备比例(%) ~ 旱灾概率(%) -----
x = droughtProb(:)*100;           % 旱灾概率(%)
y = bestRatio_arr(:)*100;         % 建议储备比例(%)
% 简单一元线性拟合
p = polyfit(x,y,1);               % y = a*x + b
a = p(1); b = p(2);
yhat = polyval(p,x);
R2 = 1 - sum((y - yhat).^2) / sum((y - mean(y)).^2);

fprintf('\n 最优拟合公式: \n');
fprintf(' 应急储备比例(%%) = %.4f × 旱灾概率(%%) + %.4f (R²=%.4f) \n', a, b,
R2);

% 最基础的 fopen 语法
fid = fopen(fullfile(out_dir,'拟合结果.txt'),'w'); % 移除所有编码参数
fprintf(fid,' 应急储备比例(%%) = %.4f × 旱灾概率(%%) + %.4f (R²=%.4f) \n',
a, b, R2);
fclose(fid);

fprintf('\n 结果已保存至文件夹: %s\n', out_dir);

```

```

function out = solve_crop_allocation(riverRatio, reserveRatio, CROPS, SYSTEM)

% 目标：最大化 S1+S2+S3

% 变量：x = [S1 S2 S3 N1 N2 N3]^T

% 约束：

% 0 <= S_i <= Area_i

% 0 <= N_i <= Area_i

% N_i <= S_i

% 水量：Sum(S_i*Dmin_i*1e4) + Sum(N_i*Dadd_i*1e4) <= Q_total


n = numel(CROPS);
A = [CROPS.Area]; A = A(:);
Dmin = [CROPS.Dmin]; Dmin = Dmin(:);
Dnor = [CROPS.Dnorm]; Dnor = Dnor(:);
Dadd = max(Dnor - Dmin, 0); % 额外水量


Qtotal = SYSTEM.DesignFlow*riverRatio + ...
        SYSTEM.TotalTankVol*reserveRatio/SYSTEM.EmergencyDays;


% 先试 linprog
use_lp = exist('linprog','file') == 2;
if use_lp
    % 决策变量： [S(1..n); N(1..n)] 共 2n
    f = [-ones(n,1); zeros(n,1)];

    % 约束： N_i - S_i <= 0
    M1 = [-eye(n), eye(n)]; b1 = zeros(n,1);

    % 上界： S_i <= A_i, N_i <= A_i
    M2 = [eye(n), zeros(n); zeros(n), eye(n)];

```

```

b2 = [A; A];

% 水量约束
wS = Dmin(:)*1e4;
wN = Dadd(:)*1e4;
M3 = [wS', wN']; b3 = Qtotal;

Aineq = [M1; M2; M3];
bineq = [b1; b2; b3];
lb = zeros(2*n, 1);
ub = [A; A];

% 求解
opts = optimset('Display','off');
[x,~,exitflag] = linprog(f, Aineq, bineq, [], [], lb, ub, opts);

if exitflag ~= 1
    % 退化到近似方案
    use_lp = false;
else
    S = x(1:n); N = x(n+1:end);
end
end

if ~use_lp
    % 近似分配（稳健保底）
    S = zeros(n, 1); N = zeros(n, 1);
    % 1) 先满足存活最低需水
    needS = A .* (Dmin*1e4);

```

```

sumNeedS = sum(needS);
if Qtotal >= sumNeedS
    S = A;
    Qrem = Qtotal - sumNeedS;
else
    scale = max(Qtotal / sumNeedS, 0);
    S = A .* scale;
    Qrem = 0;
end
% 2) 用剩余水满足“正常额外需水”
if Qrem > 0
    needN = S .* (Dadd*1e4);
    sumNeedN = sum(needN);
    if sumNeedN == 0
        N = zeros(n, 1);
    elseif Qrem >= sumNeedN
        N = S; % 全部达到正常
    else
        N = S .* (Qrem / sumNeedN);
    end
end
end

out.S = round(S(:), 4);
out.N = round(N(:), 4);
end

function s = ternary_str(cond, a, b)
    if cond

```

```

        s = a;

    else

        s = b;

    end

end
end

```

附录五：code_4

```

clear;

clc;

close all;

% 1. 核心参数定义
% 全局参数
GLOBAL_PARAMS.UncoveredRatio = 0.229;           % 未覆盖比例（8.92%）
GLOBAL_PARAMS.BaseCost = 251598.00;             % 基础建设成本（元）
GLOBAL_PARAMS.SprinklerNum = 16;                % 喷头数量
GLOBAL_PARAMS.TankNum = 3;                      % 储水罐数量
GLOBAL_PARAMS.PipeCost = 42348;                 % 管网成本（元）
GLOBAL_PARAMS.InitTankCost = 209250;            % 初始储水罐成本（元）
GLOBAL_PARAMS.CoreParam.DroughtWeight = 2.0;    % 干旱权重
GLOBAL_PARAMS.CoreParam.TankCostCoeff = 5;      % 储水成本系数（元/L）

% 储水罐参数计算
BASE_TANK_VOLUME = GLOBAL_PARAMS.InitTankCost /
(GLOBAL_PARAMS.CoreParam.TankCostCoeff * GLOBAL_PARAMS.TankNum);

TANK_TOTAL_VOLUME = BASE_TANK_VOLUME * GLOBAL_PARAMS.TankNum * 2;

EMERGENCY_RATIO = 0.9; % 应急储备比例

AVAILABLE_TANK = TANK_TOTAL_VOLUME * (1 - EMERGENCY_RATIO);

```

```

% 作物参数（确保结构体数组一致性）
crop_names = {'高粱', '玉米', '大豆'};

% 创建相同结构的阶段结构体
stage1 = struct('D', 0, 'start', '', 'end_date', '');
stage2 = struct('D', 0, 'start', '', 'end_date', '');
stage3 = struct('D', 0, 'start', '', 'end_date', '');

% 创建相同结构的作物结构体
crop1 = struct('A', 0, 'stage', [stage1, stage2, stage3]);
crop2 = struct('A', 0, 'stage', [stage1, stage2, stage3]);
crop3 = struct('A', 0, 'stage', [stage1, stage2, stage3]);

% 初始化作物参数数组（结构完全一致）
crop_params = [crop1; crop2; crop3];

% 高粱参数
crop_params(1).A = 0.5; % 公顷
crop_params(1).stage(1).D = 5;
crop_params(1).stage(1).start = '2021-05-01';
crop_params(1).stage(1).end_date = '2021-05-20';
crop_params(1).stage(2).D = 10;
crop_params(1).stage(2).start = '2021-05-21';
crop_params(1).stage(2).end_date = '2021-07-09';
crop_params(1).stage(3).D = 8;
crop_params(1).stage(3).start = '2021-07-10';
crop_params(1).stage(3).end_date = '2021-07-30';

% 玉米参数

```

```

crop_params(2).A = 0.3;
crop_params(2).stage(1).D = 6;
crop_params(2).stage(1).start = '2021-05-01';
crop_params(2).stage(1).end_date = '2021-06-01';
crop_params(2).stage(2).D = 12;
crop_params(2).stage(2).start = '2021-06-02';
crop_params(2).stage(2).end_date = '2021-07-21';
crop_params(2).stage(3).D = 10;
crop_params(2).stage(3).start = '2021-07-22';
crop_params(2).stage(3).end_date = '2021-08-10';

% 大豆参数
crop_params(3).A = 0.2;
crop_params(3).stage(1).D = 4;
crop_params(3).stage(1).start = '2021-05-01';
crop_params(3).stage(1).end_date = '2021-06-09';
crop_params(3).stage(2).D = 8;
crop_params(3).stage(2).start = '2021-06-10';
crop_params(3).stage(2).end_date = '2021-07-19';
crop_params(3).stage(3).D = 6;
crop_params(3).stage(3).start = '2021-07-20';
crop_params(3).stage(3).end_date = '2021-08-08';

% 管网布局参数
sprinkler_coords = [8, 62; 35, 23; 55, 85; 85, 53;
                    10, 20; 30, 3; 54, 44; 66, 65;
                    14, 85; 36, 94; 57, 14; 85, 85;
                    20, 47; 37, 68; 87, 12; 79, 26];
tank_coords = [40, 40; 53, 76; 46, 14];

```

```

% 月份与降水参数
months = {'5月', '6月', '7月'};
month_days = [31, 30, 31]; % 各月天数

% 有效降水量 (mm/阶段) (作物, 阶段, 月份)
precipitation = zeros(3, 3, 3);
precipitation(1, 1, 1) = 30; precipitation(1, 2, 2) = 45; precipitation(1, 3,
3) = 35;
precipitation(2, 1, 1) = 30; precipitation(2, 2, 2) = 45; precipitation(2, 3,
3) = 35;
precipitation(3, 1, 1) = 30; precipitation(3, 2, 2) = 45; precipitation(3, 3,
3) = 35;

Q_r = 200000; % 河流日供水能力 (L/天)

% =====
% 2. 主程序执行
% =====
% 步骤 1: 计算作物需水量
fprintf('步骤 1: 需水量计算完成\n');
W = calculate_W_cm(crop_params, months, month_days, GLOBAL_PARAMS);

% 步骤 2: 计算实际灌溉需求
fprintf('步骤 2: 灌溉需求计算完成\n');
irrigation = calculate_irrigation_demand(W, crop_params, precipitation,
GLOBAL_PARAMS);

% 步骤 3: 生成灌溉安排表

```

```

fprintf(' 步骤 3: 生成灌溉安排表\n');
generate_irrigation_schedule(irrigation, crop_params, months, month_days, ...
    GLOBAL_PARAMS, tank_coords, sprinkler_coords, Q_r, ...
    TANK_TOTAL_VOLUME, EMERGENCY_RATIO, AVAILABLE_TANK);

fprintf('\n 所有计算完成, 结果已保存至 CSV 文件\n');

% =====
% 3. 函数定义
% =====

% 获取管道连接关系
function connections = get_pipe_connections(tank_coords, sprinkler_coords,
radius)

    num_tanks = size(tank_coords, 1);
    connections = cell(num_tanks, 1);

    for tank_idx = 1:num_tanks
        tx = tank_coords(tank_idx, 1);
        ty = tank_coords(tank_idx, 2);
        connected = [];

        for spr_idx = 1:size(sprinkler_coords, 1)
            sx = sprinkler_coords(spr_idx, 1);
            sy = sprinkler_coords(spr_idx, 2);
            dist = sqrt((tx - sx)^2 + (ty - sy)^2);

            if dist <= radius
                connected = [connected, spr_idx];
            end
        end
    end
end

```

```

        end

    end

    connections{tank_idx} = connected;

end

end

% 计算需水量（使用 datenum 兼容低版本）
function W = calculate_W_cm(crop_params, months, month_days, GLOBAL_PARAMS)

    num_crops = length(crop_params);
    num_months = length(months);
    W = zeros(num_crops, num_months);

    for crop_idx = 1:num_crops
        A_c = crop_params(crop_idx).A;

        for stage_idx = 1:3
            stage = crop_params(crop_idx).stage(stage_idx);
            start_date = datenum(stage.start, 'yyyy-mm-dd');
            end_date = datenum(stage.end_date, 'yyyy-mm-dd');

            for month_idx = 1:num_months
                m_num = month_idx + 4;
                m_start = datenum(2021, m_num, 1);
                m_end = datenum(2021, m_num, month_days(month_idx));

                % 计算重叠日期
                if start_date > m_start
                    overlap_start = start_date;
                else

```

```

        overlap_start = m_start;
    end

    if end_date < m_end
        overlap_end = end_date;
    else
        overlap_end = m_end;
    end

    % 计算重叠天数
    if overlap_start <= overlap_end
        d_cim = overlap_end - overlap_start + 1;
    else
        d_cim = 0;
    end

    % 计算需水量
    W(crop_idx, month_idx) = W(crop_idx, month_idx) + ...
        A_c * 10000 * d_cim * stage.D * (1 -
GLOBAL_PARAMS.UncoveredRatio);

    end

    end

    end

end

% 计算灌溉需求
function irrigation = calculate_irrigation_demand(W, crop_params,
precipitation, GLOBAL_PARAMS)

    num_crops = size(W, 1);

```

```

num_months = size(W, 2);
irrigation = zeros(num_crops, num_months);

for crop_idx = 1:num_crops
    A_c = crop_params(crop_idx).A;

    for month_idx = 1:num_months
        total_rain = 0;

        for stage_idx = 1:3
            rain_mm = precipitation(crop_idx, stage_idx, month_idx);
            total_rain = total_rain + rain_mm * A_c * 10000 * (1 -
GLOBAL_PARAMS.UncoveredRatio);
        end

        irrigation(crop_idx, month_idx) = max(0, W(crop_idx, month_idx) -
total_rain);
    end
end
end

% 生成灌溉安排表（适配低版本 fopen）
function generate_irrigation_schedule(irrigation, crop_params, months,
month_days, ...
    GLOBAL_PARAMS, tank_coords, sprinkler_coords, Q_r, ...
    tank_total_volume, emergency_ratio, available_tank)
crop_names = {'高粱', '玉米', '大豆'};
num_crops = length(crop_names);
num_months = length(months);

```

```

% 水源分配比例矩阵
theta_mat = [0.7, 0.6, 0.5;
             0.6, 0.5, 0.4;
             0.5, 0.4, 0.3];

% 创建灌溉安排表
table5 = cell(10, 5);
table5(1, :) = {'日期', '作物', '总灌溉量(L)', '水源比例(河水/储水罐)', '
备注'};

row = 2;
for month_idx = 1:num_months
    m = months{month_idx};

    % 计算当月储水罐 7 天用量
    monthly_tank_usage = sum((1 - theta_mat(:, month_idx)) .*
irrigation(:, month_idx));

    weekly_tank_usage = (monthly_tank_usage / month_days(month_idx)) * 7;

    for crop_idx = 1:num_crops
        total_irr = irrigation(crop_idx, month_idx);
        ratio_river = theta_mat(crop_idx, month_idx);
        ratio_tank = 1 - ratio_river;
        ratio_str = sprintf('%.2f/%.2f', ratio_river, ratio_tank);

        % 判断系统状态
        if weekly_tank_usage <= 0
            remark = '无储水需求';

```

```

elseif weekly_tank_usage <= available_tank * 0.9
    remark = '正常运行（未启用应急储备）';
elseif weekly_tank_usage <= available_tank
    remark = '接近储水上限（未启用应急储备）';
elseif weekly_tank_usage <= tank_total_volume
    usage_ratio = weekly_tank_usage / tank_total_volume;
    remark = sprintf('已启用应急储备（使用%.0f%%总容量）',
usage_ratio * 100);
else
    remark = '需扩容储水罐（超出设计容积）';
end

% 填充表格
table5{row, 1} = m;
table5{row, 2} = crop_names{crop_idx};
table5{row, 3} = num2str(round(total_irr, 2));
table5{row, 4} = ratio_str;
table5{row, 5} = remark;
row = row + 1;
end
end

% 输出灌溉安排表
fprintf('\n 表 5 灌溉安排表: \n');
for i = 1:size(table5, 1)
    fprintf('%s\t%s\t%s\t%s\t%s\n', table5{i, 1}, table5{i, 2}, ...
        table5{i, 3}, table5{i, 4}, table5{i, 5});
end

```

```

% 适配低版本 MATLAB 的 fopen 调用（仅保留文件名和打开模式）
fid = fopen('表 5_灌溉安排表.csv', 'w');
for i = 1:size(table5, 1)
    fprintf(fid, '%s,%s,%s,%s,%s\n', table5{i, 1}, table5{i, 2}, ...
        table5{i, 3}, table5{i, 4}, table5{i, 5});
end
fclose(fid);

% 输出系统参数信息
fprintf('\n=== 系统参数信息 ===\n');
fprintf('1. 基础参数: \n');
fprintf('    - 未覆盖比例: %.2f%%\n', GLOBAL_PARAMS.UncoveredRatio * 100);
fprintf('    - 喷头数量: %d 个\n', GLOBAL_PARAMS.SprinklerNum);
fprintf('    - 储水罐数量: %d 个\n', GLOBAL_PARAMS.TankNum);
fprintf('2. 储水系统: \n');
fprintf('    - 应急储备比例: %.2f\n', emergency_ratio);
fprintf('    - 总容积: %.2fL\n', tank_total_volume);
fprintf('    - 可用容积: %.2fL\n', available_tank);
end

```

附录六: code_5

```

clear; clc; close all;
disp('开始气象数据预处理...');

%% 1. 定义保存路径（简化路径，避免特殊字符）
save_dir = 'D:\AgriIrrigation\';
if ~exist(save_dir, 'dir')
    mkdir(save_dir);
    disp(['✓ 已创建保存目录: ', save_dir]);
end

```

```

else
    disp(['✓ 保存目录已存在: ', save_dir]);
end

%% 2. 读取 Excel 文件（低版本 xlsread 兼容）
try
    [~, ~, raw_data] = xlsread('降水量等逐时气象数据 1.xlsx');
    disp('✓ 成功读取 Excel 文件');
catch ME
    error('读取 Excel 失败: %s', getReport(ME));
end

% 检查表头行存在性
if size(raw_data, 1) < 2
    error('Excel 格式错误! 至少需包含 2 行（第 1 行说明+第 2 行表头）');
end

%% 3. 表头清理与目标列匹配
raw_headers = raw_data(2, :); % 原始表头在第 2 行
for i = 1:length(raw_headers)
    if ischar(raw_headers{i})
        raw_headers{i} = strtrim(raw_headers{i}); % 去除首尾空格
    else
        raw_headers{i} = ''; % 非字符串表头视为空
    end
end

% 定义需提取的 15 列
target_headers = {

```

```

    '当地时间 长春市(机场)', 'T', 'Po', 'P', 'Pa', 'U', ...
    'Ff', 'ff3', 'Tn', 'Tx', 'VV', 'Td', 'tR', 'DD', 'RRR'
};

n_target = length(target_headers);

% 动态匹配目标列索引 - 优化匹配逻辑
target_indices = zeros(1, n_target);
missing_cols = {};
for i = 1:n_target
    curr_target = target_headers{i};
    match_flag = false;
    % 精确匹配, 不区分大小写但需完整匹配
    for j = 1:length(raw_headers)
        curr_raw = raw_headers{j};
        if strcmpi(strtrim(curr_raw), strtrim(curr_target))
            target_indices(i) = j;
            match_flag = true;
            break; % 只匹配第一个完全符合的列
        end
    end
    if ~match_flag
        missing_cols{end+1} = curr_target;
    end
end

if ~isempty(missing_cols)
    error(['Excel 表头缺失列: "', strjoin(missing_cols, '", "'), '"']);
end

disp('✓ 15 列均已匹配成功');

```

```

%% 4. 提取数据并清理空行

target_indices = uint32(target_indices);
data = raw_data(3:end, target_indices);
data = delete_empty_rows(data);

if isempty(data)
    error('无有效数据！请检查 Excel 第 3 行及以后内容');
end
n_rows = size(data, 1);
disp(['✓ 数据提取完成，有效行数：', num2str(n_rows)]);

%% 5. 时间列处理（低版本 datenum 兼容）
time_col = 1;
dd_col = 14; % DD 列索引
rr_col = 15; % RRR 列索引
time_num = NaN(n_rows, 1);
for i = 1:n_rows
    time_str = data{i, time_col};
    if ~isempty(time_str) && ischar(time_str)
        time_str = regexprep(time_str, '[^0-9:./ ]', '');
        % 尝试多种日期格式解析
        try
            if ~isnan(datenum(time_str, 'dd.mm.yyyy HH:MM'))
                time_num(i) = datenum(time_str, 'dd.mm.yyyy HH:MM');
            else
                time_num(i) = datenum(time_str, 'yyyy-mm-dd HH:MM');
            end
        catch
    end
end

```

```

        time_num(i) = NaN;
    end
end
end

valid_idx = ~isnan(time_num);
data = data(valid_idx, :);
time_num = time_num(valid_idx);
n_rows = sum(valid_idx);

if n_rows == 0
    error('无有效时间数据！请检查第1列格式');
end

disp(['✓ 时间处理完成，剩余有效行数：', num2str(n_rows)]);

%% 6. 全列填充（按要求处理各列）
filled_data = cell(size(data));

% 首先提取所有数值列，为插值做准备
numeric_data = NaN(n_rows, n_target);
col_types = cell(1, n_target); % 记录每列类型

% 第一步：识别列类型并提取数值
for col = 1:n_target
    % 跳过时间列，时间列不参与插值
    if col == time_col
        col_types{col} = 'time';
        continue;
    end
end

```

```

% 强制 DD 列为字符串类型（使用众数填充）
if col == dd_col
    col_types{col} = 'string';
    continue;
end

% 强制 RRR 列为数值类型（使用线性插值）
if col == rr_col
    col_types{col} = 'numeric';
    col_vals = NaN(n_rows, 1);
    for row = 1:n_rows
        cell_val = data{row, col};
        col_vals(row) = extract_number(cell_val);
    end
    numeric_data(:, col) = col_vals;
    continue;
end

% 其他列自动识别类型
is_numeric = true;
col_vals = NaN(n_rows, 1);

for row = 1:n_rows
    cell_val = data{row, col};
    num_val = extract_number(cell_val);
    if isnan(num_val)
        is_numeric = false;
    end
end

```

```

        col_vals(row) = num_val;
    end

    valid_ratio = sum(~isnan(col_vals)) / n_rows;
    if valid_ratio > 0.5
        col_types{col} = 'numeric';
        numeric_data(:, col) = col_vals;
    else
        col_types{col} = 'string';
    end
end

% 第二步：按列类型填充数据
for col = 1:n_target
    if col == time_col % 时间列直接复制
        for row = 1:n_rows
            filled_data{row, col} = data{row, col};
        end
        continue;
    end

    % DD 列强制使用众数填充
    if col == dd_col
        str_vals = cell(n_rows, 1);
        for row = 1:n_rows
            cell_val = data{row, col};
            if ischar(cell_val)
                str_vals{row} = strtrim(cell_val);
            else

```

```

        str_vals{row} = '';
    end
end

% 计算众数 - 低版本兼容处理
valid_strs = str_vals(~cellfun('isempty', str_vals));
mode_str = '';
if ~isempty(valid_strs)
    % 确保所有元素都是字符行向量
    str_array = cellfun(@(x) x(:)', valid_strs, 'UniformOutput',
false);

    % 手动去重（低版本 unique 对 cell 支持有限）
    unique_str = {};
    for i = 1:length(str_array)
        curr_str = str_array{i};
        is_duplicate = false;
        for j = 1:length(unique_str)
            if strcmp(curr_str, unique_str{j})
                is_duplicate = true;
                break;
            end
        end
        if ~is_duplicate
            unique_str{end+1} = curr_str;
        end
    end

    % 计算每个唯一字符串的出现次数
    str_count = zeros(length(unique_str), 1);

```

```

        for k = 1:length(unique_str)
            str_count(k) = sum(strcmp(str_array, unique_str{k}));
        end
        [~, mode_idx] = max(str_count);
        mode_str = unique_str{mode_idx};
    else
        mode_str = '未知';
    end

    % 填充空值
    for row = 1:n_rows
        if isempty(str_vals{row})
            filled_data{row, col} = mode_str;
        else
            filled_data{row, col} = str_vals{row};
        end
    end

    disp(['✓ 列 ', num2str(col), ' (', target_headers{col}, ') 众数填充完
成']);

    continue;
end

% RRR 列强制使用线性插值
if col == rr_col
    vals = numeric_data(:, col);
    time_vals = time_num;

    valid_mask = ~isnan(vals);
    if sum(valid_mask) < 2

```

```

        col_mean = mean(vals(valid_mask));
        if isnan(col_mean)
            col_mean = 0;
        end
        filled_vals = repmat(col_mean, n_rows, 1);
    else
        filled_vals = interp1(time_vals(valid_mask),
vals(valid_mask), ...
                                time_vals, 'linear', 'extrap');
    end

    for row = 1:n_rows
        filled_data{row, col} = filled_vals(row);
    end
    disp(['✓ 列 ', num2str(col), ' (', target_headers{col}, ') 线性插值填
充完成']);
    continue;
end

% 处理其他列
if strcmp(col_types{col}, 'numeric')
    vals = numeric_data(:, col);
    time_vals = time_num;

    valid_mask = ~isnan(vals);
    if sum(valid_mask) < 2
        col_mean = mean(vals(valid_mask));
        if isnan(col_mean)
            col_mean = 0;

```

```

        end

        filled_vals = repmat(col_mean, n_rows, 1);
    else
        filled_vals = interp1(time_vals(valid_mask),
vals(valid_mask), ...
                                time_vals, 'linear', 'extrap');
    end

    for row = 1:n_rows
        filled_data{row, col} = filled_vals(row);
    end

    disp(['✓ 列 ', num2str(col), ' (', target_headers{col}, ') 线性插值填
充完成']);
else
    str_vals = cell(n_rows, 1);
    for row = 1:n_rows
        cell_val = data{row, col};
        if ischar(cell_val)
            str_vals{row} = strtrim(cell_val);
        else
            str_vals{row} = '';
        end
    end

    end

    % 计算众数 - 低版本兼容处理
    valid_strs = str_vals(~cellfun('isempty', str_vals));
    mode_str = '';
    if ~isempty(valid_strs)
        % 确保所有元素都是字符行向量

```

```

        str_array = cellfun(@(x) x(:)', valid_strs, 'UniformOutput',
false);

% 手动去重（代替低版本不支持的 unique(cell)）
unique_str = {};
for i = 1:length(str_array)
    curr_str = str_array{i};
    is_duplicate = false;
    for j = 1:length(unique_str)
        if strcmp(curr_str, unique_str{j})
            is_duplicate = true;
            break;
        end
    end
    if ~is_duplicate
        unique_str{end+1} = curr_str;
    end
end

% 计算每个唯一字符串的出现次数
str_count = zeros(length(unique_str), 1);
for k = 1:length(unique_str)
    str_count(k) = sum(strcmp(str_array, unique_str{k}));
end
[~, mode_idx] = max(str_count);
mode_str = unique_str{mode_idx};
else
    mode_str = '未知';
end

```

```

        for row = 1:n_rows
            if isempty(str_vals{row})
                filled_data{row, col} = mode_str;
            else
                filled_data{row, col} = str_vals{row};
            end
        end

        disp(['✓ 列 ', num2str(col), ' (', target_headers{col}, ') 众数填充完
成']);

    end

end

disp('✓ 所有 15 列数据填充完成（无空值）');

%% 7. DD 列风向编码（纯 if-else，无高版本函数）
for row = 1:n_rows
    dd_val = filled_data{row, dd_col};
    % 转换为字符串
    if ischar(dd_val)
        dd_str = lower(dd_val);
    else
        dd_str = lower(num2str(dd_val));
    end

    % 风向编码（低版本兼容的 strfind）
    if ~isempty(strfind(dd_str, '东北'))
        filled_data{row, dd_col} = 5;
    elseif ~isempty(strfind(dd_str, '东南'))
        filled_data{row, dd_col} = 6;
    elseif ~isempty(strfind(dd_str, '西北'))

```

```

        filled_data{row, dd_col} = 7;
elseif ~isempty(strfind(dd_str, '西南'))
        filled_data{row, dd_col} = 8;
elseif ~isempty(strfind(dd_str, '东'))
        filled_data{row, dd_col} = 1;
elseif ~isempty(strfind(dd_str, '南'))
        filled_data{row, dd_col} = 2;
elseif ~isempty(strfind(dd_str, '西'))
        filled_data{row, dd_col} = 3;
elseif ~isempty(strfind(dd_str, '北'))
        filled_data{row, dd_col} = 4;
elseif ~isempty(strfind(dd_str, '无风'))
        filled_data{row, dd_col} = 0;
else
        filled_data{row, dd_col} = 9;
end
end
disp('✓ DD 列风向编码完成');

%% 8. RRR 列降水量处理（纯 if-else 逻辑）
for row = 1:n_rows
    rr_val = filled_data{row, rr_col};
    rr_str = lower(num2str(rr_val));

    if ~isempty(strfind(rr_str, '无降水')) || ~isempty(strfind(rr_str, '痕迹
'))
        filled_data{row, rr_col} = 0;
    else
        num_match = regexp(rr_str, '\d+\.\d*', 'match', 'once');

```

```

        if ~isempty(num_match)
            rr_num = str2double(num_match);
            if rr_num < 0
                filled_data{row, rr_col} = 0;
            else
                filled_data{row, rr_col} = rr_num;
            end
        else
            filled_data{row, rr_col} = 0;
        end
    end
end

disp('✓ RRR 列降水量处理完成');

%% 9. 保存逐时数据（低版本 fopen 兼容，无编码参数）
hourly_data = [target_headers; filled_data];
hourly_path = [save_dir, 'weather_hourly.csv'];

% 低版本 fopen 仅支持路径和模式参数
fid_hourly = fopen(hourly_path, 'w');
if fid_hourly == -1
    error('无法创建逐时数据文件！请手动检查路径权限');
end

% 写入数据（纯循环，无高版本语法）
for row = 1:size(hourly_data, 1)
    curr_row = hourly_data(row, :);
    for col = 1:n_target
        cell_val = curr_row{col};
    end
end

```

```

        if isnumeric(cell_val)
            fprintf(fid_hourly, '%g', cell_val);
        else
            fprintf(fid_hourly, '"%s"', cell_val);
        end
        if col < n_target
            fprintf(fid_hourly, ',');
        end
    end
    fprintf(fid_hourly, '\n');
end
fclose(fid_hourly);
disp(['✓ 逐时数据已保存至: ', hourly_path]);

%% 10. 逐日聚合与保存（低版本兼容逻辑）
% 提取日期
dates = cell(n_rows, 1);
for i = 1:n_rows
    [y, m, d] = datevec(time_num(i));
    dates{i} = sprintf('%04d-%02d-%02d', y, m, d);
end
% 日期去重（低版本兼容）
unique_dates = {};
for i = 1:length(dates)
    curr_date = dates{i};
    is_duplicate = false;
    for j = 1:length(unique_dates)
        if strcmp(curr_date, unique_dates{j})
            is_duplicate = true;
        end
    end
    if ~is_duplicate
        unique_dates = [unique_dates; curr_date];
    end
end

```

```

        break;
    end
end
if ~is_duplicate
    unique_dates{end+1} = curr_date;
end
end
n_days = length(unique_dates);

% 初始化逐日数据：(天数+1)行（表头+数据）、(15+1)列（日期+15 个目标列）
daily_data = cell(n_days + 1, n_target + 1);

% 设置表头行
daily_data(1, :) = [{'日期'}, target_headers];

% 逐天聚合
for day_idx = 1:n_days
    curr_date = unique_dates{day_idx};
    daily_data(day_idx + 1, 1) = {curr_date}; % 第 1 列存日期
    day_mask = strcmp(dates, curr_date);
    day_rows = filled_data(day_mask, :);
    n_day_rows = sum(day_mask);

    for col = 1:n_target
        % 检查当前列是否为 RRR 降水量列或 DD 列
        is_rrr_col = (col == rr_col);
        is_dd_col = (col == dd_col);

        if is_dd_col

```

```

% DD 列使用众数填充 - 低版本兼容
str_vals = cell(n_day_rows, 1);
for row = 1:n_day_rows
    str_vals{row} = day_rows{row, col};
end
valid_strs = str_vals(~cellfun('isempty', str_vals));
if ~isempty(valid_strs)
    % 确保所有元素都是字符行向量
    str_array = cellfun(@(x) x(:)', valid_strs, 'UniformOutput',
false);

    % 手动去重
    unique_str = {};
    for i = 1:length(str_array)
        curr_str = str_array{i};
        is_duplicate = false;
        for j = 1:length(unique_str)
            if strcmp(curr_str, unique_str{j})
                is_duplicate = true;
                break;
            end
        end
        if ~is_duplicate
            unique_str{end+1} = curr_str;
        end
    end

    % 计算每个唯一字符串的出现次数
    str_count = zeros(length(unique_str), 1);
    for k = 1:length(unique_str)

```

```

        str_count(k) = sum(strcmp(str_array, unique_str{k}));
    end
    [~, mode_idx] = max(str_count);
    daily_data(day_idx + 1, col + 1) = {unique_str{mode_idx}};
else
    daily_data(day_idx + 1, col + 1) = {' '};
end
elseif is_rrr_col
    % RRR 列进行累加
    col_vals = NaN(n_day_rows, 1);
    for row = 1:n_day_rows
        cell_val = day_rows{row, col};
        if isnumeric(cell_val)
            col_vals(row) = cell_val;
        else
            col_vals(row) = 0;
        end
    end
    daily_data(day_idx + 1, col + 1) =
{sum(col_vals(~isnan(col_vals)))};
else
    % 其他数值列计算均值
    is_numeric_col = true;
    col_vals = NaN(n_day_rows, 1);
    for row = 1:n_day_rows
        cell_val = day_rows{row, col};
        if isnumeric(cell_val)
            col_vals(row) = cell_val;
        else

```

```

        is_numeric_col = false;
        break;
    end
end

if is_numeric_col
    if n_day_rows > 0
        daily_data(day_idx + 1, col + 1) =
{mean(col_vals(~isnan(col_vals)))});
    else
        daily_data(day_idx + 1, col + 1) = {0};
    end
else
    str_vals = cell(n_day_rows, 1);
    for row = 1:n_day_rows
        str_vals{row} = day_rows{row, col};
    end
    valid_strs = str_vals(~cellfun('isempty', str_vals));
    if ~isempty(valid_strs)
        % 确保所有元素都是字符行向量
        str_array = cellfun(@(x) x(:)', valid_strs,
'UniformOutput', false);
        % 手动去重
        unique_str = {};
        for i = 1:length(str_array)
            curr_str = str_array{i};
            is_duplicate = false;
            for j = 1:length(unique_str)
                if strcmp(curr_str, unique_str{j})

```

```

            is_duplicate = true;
            break;
        end
    end
    if ~is_duplicate
        unique_str{end+1} = curr_str;
    end
end

% 计算每个唯一字符串的出现次数
str_count = zeros(length(unique_str), 1);
for k = 1:length(unique_str)
    str_count(k) = sum(strcmp(str_array, unique_str{k}));
end
[~, mode_idx] = max(str_count);
daily_data(day_idx + 1, col + 1) =
{unique_str{mode_idx}};
    else
        daily_data(day_idx + 1, col + 1) = {''};
    end
end
end
end

disp('✓ 逐日聚合完成（RRR 列累加，DD 列众数，其他列均值）');

% 保存逐日数据
daily_path = [save_dir, 'weather_daily.csv'];
fid_daily = fopen(daily_path, 'w');

```

```

if fid_daily == -1
    error('无法创建逐日数据文件！请手动检查路径权限');
end

for row = 1:size(daily_data, 1)
    curr_row = daily_data(row, :);
    for col = 1:n_target + 1
        cell_val = curr_row{col};
        if isnumeric(cell_val)
            fprintf(fid_daily, '%g', cell_val);
        else
            fprintf(fid_daily, '"%s"', cell_val);
        end
        if col < n_target + 1
            fprintf(fid_daily, ',');
        end
    end
    fprintf(fid_daily, '\n');
end
fclose(fid_daily);
disp(['✓ 逐日数据已保存至：', daily_path]);

disp(['\n✅ 预处理完成！兼容低版本 MATLAB，双文件保存成功']);

%% 子函数：删除全空行（纯循环，无高版本函数）
function cleaned_data = delete_empty_rows(raw_data)
    if isempty(raw_data)
        cleaned_data = raw_data;
        return;
    end
end

```

```

end

[n_rows, n_cols] = size(raw_data);
is_empty_row = false(n_rows, 1);

for row = 1:n_rows
    empty_flag = true;
    for col = 1:n_cols
        cell_val = raw_data{row, col};
        if ~isempty(cell_val) && ~(isnumeric(cell_val) &&
isnan(cell_val))
            empty_flag = false;
            break;
        end
    end
    is_empty_row(row) = empty_flag;
end
cleaned_data = raw_data(~is_empty_row, :);
end

%% 子函数：提取数值（低版本兼容）
function num = extract_number(cell_val)

    num = NaN;

    if isempty(cell_val)
        return;
    end

    if isnumeric(cell_val)
        num = cell_val;
        return;
    end
end

```

```
if ischar(cell_val)
    clean_str = regexp(cell_val, '^[0-9.-]', '');
    if ~isempty(clean_str)
        num = str2double(clean_str);
    end
end
end
```

作者：熊 函 王 琦 苑景博

摘要：多波束测深技术是由单波束测深发展而来的一种海洋地形测绘方法，显著提高了海底地形测量的效率和完整性。本文建立测绘优化模型，运用各种几何分析、递推法、最小二乘法等方法研究测线的覆盖深度、重叠率以及测线的布局。

针对问题一：首先以海域中心点为坐标原点建立合适的坐标系。其次利用几何关系计算海水深度，并通过三角函数和正弦定理计算出覆盖宽度。接着利用几何知识推导重叠率的计算公式并代入变量计算。最终得到特定位置的相关结果，对其进行结果分析后，存放在 result1.xlsx 中。

针对问题二：在问题一的基础上，以测线投影到海平面上的海域中心处建立三维空间直角坐标系。结合坡度投影与三角函数，推导出此时坡度的表达式。接着，由测线方向与海平面平行，将相邻测线的距离转换到海平面上变化的距离，求解该位置到海底坡面的距离，得到位置变化的函数表达式。然后根据问题一建立的模型求解得到特定位置处的覆盖宽度，结果存放在 result2.xlsx 中，最后进行结果分析和原因分析。

针对问题三：对于一个具体的矩形海域，西深东浅，南北方向海水等深，本文以测线总长度最短为目标，以全覆盖和重叠率控制在 10% - 20% 为约束，建立了测线布设优化模型。通过递推法确定每条测线的位置，并利用 MATLAB 计算，得到结果：在重叠率为 10.5% 时，仅需 34 条测线即可实现全覆盖，总长度为 68 海里。

针对问题四：本文基于附件提供的单波束测深数据，构建了一个多波束测深测线布设优化模型。通过三维地形可视化与区域划分，将复杂海域划分为三个子区域，并利用最小二乘法拟合各区域海底坡面。以测线总长度最短为主要目标，以满足全覆盖和重叠率控制为约束条件，建立单目标规划模型，求解最优测线布设方案。模型最终输出测线总长度、漏测海区面积占比及重叠率超过 20% 部分的总长度等关键指标，为多波束测量船的实际作业提供数据支持与决策依据。

关键词：多波束测深；迭代公式；最小二乘法；优化模型

点 评：本文针对多波束测深中的测线优化布设问题，从几何建模到优化求解展现了完整的数学建模流程。问题一和问题二通过三角函数和正弦定理等几何方法建立覆盖宽度和重

叠率的解析表达式，推导严谨、物理意义清晰。问题三将实际问题抽象为以测线总长度最短为目标的目标优化模型，约束条件（全覆盖、重叠率 10%-20%）贴近工程实际。问题四利用单波束测深数据拟合海底地形，将复杂海域分区域处理，体现了化繁为简的建模思想。方法上以解析几何和经典优化为主，虽未使用复杂的智能算法，但求解结果（34 条测线、重叠率 10.5%）合理且可解释性强。建议可进一步考虑海底地形的局部起伏对测线布设的影响，引入自适应间距调整策略。

多波束测深测线优化研究

摘要

多波束测深技术是由单波束测深发展而来的一种海洋地形测绘方法，显著提高了海底地形测量的效率和完整性。本文建立测绘优化模型，运用各种几何分析、递推法、最小二乘法等方法研究测线的覆盖深度、重叠率以及测线的布局。

针对问题一：首先以海域中心点为坐标原点建立合适的坐标系。其次利用几何关系计算海水深度，并通过三角函数和正弦定理计算出覆盖宽度。接着利用几何知识推导重叠率的计算公式并代入变量计算。最终得到特定位置的相关结果，对其进行结果分析后，存放在 result1.xlsx 中。

针对问题二：在问题一的基础上，以测线投影到海平面上的海域中心处建立三维空间直角坐标系。结合坡度投影与三角函数，推导出此时坡度的表达式。接着，由测线方向与海平面平行，将相邻测线的距离转换到海平面上变化的距离，求解该位置到海底坡面的距离，得到位置变化的函数表达式。然后根据问题一建立的模型求解得到特定位置处的覆盖宽度，结果存放在 result2.xlsx 中，最后进行结果分析和原因分析。

针对问题三：对于一个具体的矩形海域，西深东浅，南北方向海水等深，本文以测线总长度最短为目标，以全覆盖和重叠率控制在 10% - 20% 为约束，建立了测线布设优化模型。通过递推法确定每条测线的位置，并利用 MATLAB 计算，得到结果：在重叠率为 10.5% 时，仅需 34 条测线即可实现全覆盖，总长度为 68 海里。

针对问题四：本文基于附件提供的单波束测深数据，构建了一个多波束测深测线布设优化模型。通过三维地形可视化与区域划分，将复杂海域划分为三个子区域，并利用最小二乘法拟合各区域海底坡面。以测线总长度最短为主要目标，以满足全覆盖和重叠率控制为约束条件，建立单目标规划模型，求解最优测线布设方案。模型最终输出测线总长度、漏测海区面积占比及重叠率超过 20% 部分的总长度等关键指标，为多波束测量船的实际作业提供数据支持与决策依据。

关键词： 多波束测线深 迭代公式 最小二乘法 优化模型

一、问题重述

1.1 问题背景

多波束测深技术是在单波束测深基础上发展起来的一种海洋地形测绘方法。它能在与航迹垂直的平面内发射多个波束，形成一条具有一定宽度的全覆盖水深条带，显著提高了海底地形测量的效率和完整性。虽平均重叠率可满足要求，但由于实际海底地形存在复杂起伏，均匀布设测线容易导致浅水区漏测或者深水区重叠过多。

1.2 问题要求

基于上述背景和已知数据，要求建立数学模型解决以下问题：

问题一： 当测线方向与海底坡面成 α 角时，构建多波束测深的覆盖宽度与相邻条带之间重叠率的数学函数，在特定开角、坡度、水深的情境下应用该函数，求解特定位置上的水深、覆盖宽度、重叠率。

问题二： 在一个矩形待测海域上，已知海底坡面的法向与测线方向在水平面的夹角 β ，建立多波束测深覆盖宽度的数学模型。在开角为 120° 、坡度为 1.5° 、海域中心处海水深度为 120m 应用该模型，求解出某些区域、特定角度下的覆盖宽度。

问题三： 在一个长 2 海里、宽 4 海里的矩形海域内，海域中心处海水深度、坡度、开角均已知，相邻条带之间的重叠率在 10% - 20% 的情况下，为该海域设计一组长度最短的测线。

问题四： 题目要求基于附件所提供的某海域单波束测深数据，为该海域的多波束测量船设计一组测线，满足以下三个条件：沿测线扫描形成的条带应尽可能覆盖整个待测海域；相邻条带之间的重叠率尽量控制在 20% 以下；测线的总长度尽可能短。待测海域南北长 5 海里、东西宽 4 海里，附件中提供了离散点的海水深度数据，横纵坐标单位为海里。要求在完成测线布设后，计算以下三项指标：测线的总长度；漏测海区占总待测海域面积的百分比；在重叠区域中，重叠率超过 20% 的部分的总长度。

二、问题分析

2.1 问题一的分析

针对问题一，对于多波束测深覆盖宽度的求解，首先要求出海水深度与测线到中心点处距离的函数表达式。再利用三角函数和正弦定理得到覆盖宽度与海水深度的关系式。由相邻两条测线在坡度上的间距，结合三角函数建立重叠率与覆盖深度之间的数学模型。据此，模

型建立完成，代入具体参数，即可分别得到海水深度、覆盖宽度和重叠率。

2.2 问题二的分析

在问题一的基础上，把海域由二维转换成三维。在问题一的求解中，我们知道只要确定水平面与坡度所在平面的夹角和该位置处的水深就能够求解出特定角度下该位置的多波束测覆盖宽度。结合几何关系和三角函数就能得到所需夹角的函数表达式。与第一问相同，利用三角函数和正弦定理得到水深与测线到中心点距离的关系式。经过整理就能建立三维情况下覆盖宽度的模型。将该模型代入到特定数据即可得到特定角度、特定位置下的覆盖宽度。

2.3 问题三的分析

给定一个南北长 2 海里、东西宽 4 海里的矩形海域，西深东浅。问题三要求设计一组测线满足总体长度最短、海域全覆盖。这需要我们从测线方向和测线间距这两个部分进行考虑。在测线方向上，由于水深由西向东逐渐加大，而南北方向上海底平坦。因此测量船整体由西向东走，沿南北方向发射波束。在测线间距上，为保证长度最短，沿东西方向上第一条测线覆盖宽度的左边界应该刚好与海域起始边界重合，使其不漏测、不重叠。然后在测量船行进过程中，保证最后一条测线的覆盖范围至少到达终止边界。据此模型建立完成，逐渐改变重叠率，直到找到最少测线即可。

2.4 问题四的分析

问题四的核心是在已知海域水深分布的情况下，设计一组满足覆盖要求且总长度最短的测线。由于海域地形起伏较大，直接采用统一坡度或平均水深设计测线会导致局部漏测或重叠过多，因此需对海域进行区域划分，并分别拟合坡面。

我们首先根据附件数据绘制海底水深在水平面上的投影图和等深线图，从而识别地形特征，然后把海域划分为三个子区域，使每个区域内地形变化相对平缓。在每个子区域内，利用最小二乘法拟合一个理想坡面，进而应用问题一和问题二中建立的覆盖宽度与重叠率模型。接着，以测线总长度最小为目标函数，以全覆盖和重叠率不超过 20% 为约束条件，建立单目标规划模型。通过迭代求解各子区域内测线的位置和间距，最终整合得到全局测线布设方案，并计算所需的三项指标。

综上，四个问题的分析路线可表示为：

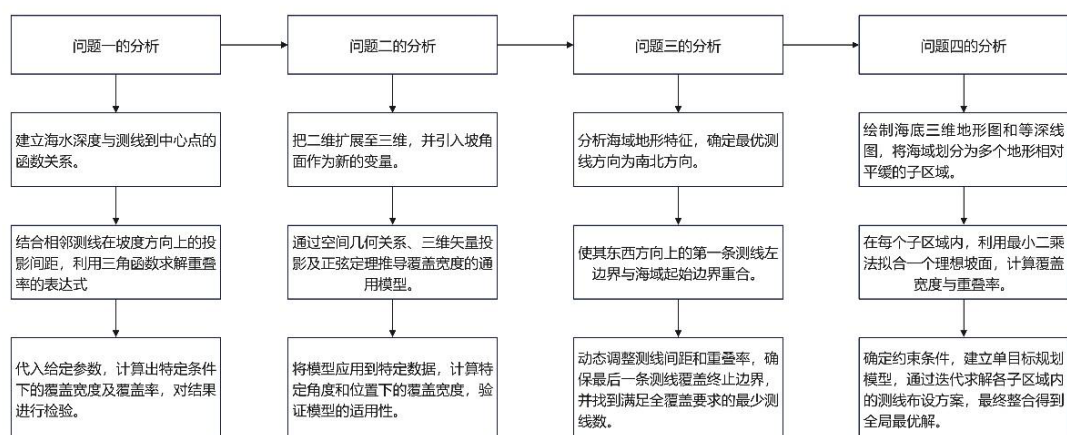


图 1 整体思维图

三、模型假设

1. 假设海底地形可被简化为一个均匀的理想坡面。
2. 假设多波束测深系统的工作过程是理想的，即声波沿直线传播，波束开角固定且对称，忽略海水折射、声速变化等复杂物理因素。
3. 假设测量船在航行过程中始终保持平稳。
4. 假设条带边缘清晰，不存在模糊区域。

四、符号说明

符号	说明	单位
D	海域中心处的水深	m
D_i	第 i 条测线处的海底深度	m
d_i	第 i 条测线到海域中心处的距离	m
W_i	第 i 条测线的覆盖宽度	m
η_i	第 i 条测线的重叠率	/
α	海底坡面与水平面的夹角	rad
β	测线方向与海底坡面法向在水平面上投影夹角	m
γ	第 i 块板凳前后把手中心所在直线	/
$d_{i,i+1}$	第 i 条测线到第 $i+1$ 条测线的距离	m

五、模型建立与求解

5.1 问题一模型的建立与求解

依据题意，我们需要构建与测线方向垂直的平面和海底坡面的交线呈 α 角的数学模型，意思是我们要将复杂的海底测深问题转化为直观的几何问题，且通过明确的公式计算，进一步准确求解出特定位置上测线的海水深度、覆盖宽度和重叠率，为后续测线优化模型的递进构建奠定了合理准确的基础。

5.1.1 测量不同位置的海水深度

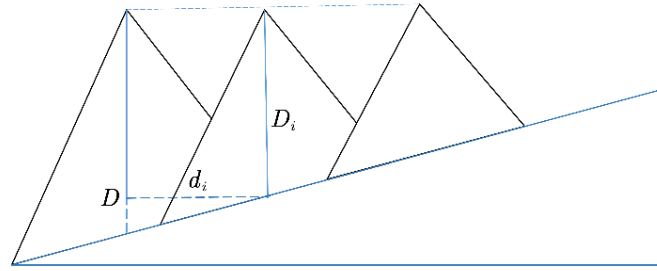


图 2 相邻测线水深关系式

如图所示，已知海域中心处的深度 D 和第 i 条测线到中心点的距离 d_i 。由于测线中心点沿坡度向下的方向海水深度更大，参考角度关系和平行线性质，求解得到第 i 条测线处的海水深度为：

$$D_i = D - d_i \tan \alpha (1)$$

5.1.2 测量覆盖深度

由于沿测线方向垂直的平面与海底坡面之间存在一定的坡度，这使得测线在海底的覆盖范围并非简单的水平延展，因此我们所需求取的 w_i 并非水平面上的投影宽度，而是第 i 条测线在实际海底坡面上所形成的覆盖宽度。目前，我们已经明确了任意测线处的海水深度这一基础数据，接下来将以第 i 条测线为研究对象，进行具体分析如下：

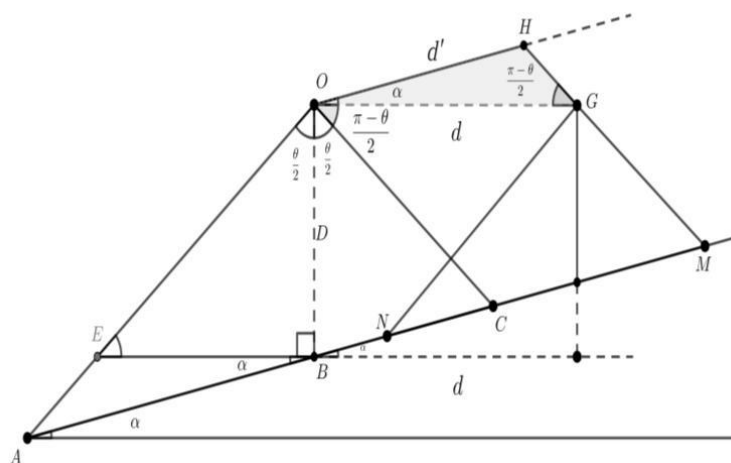


图 4 相邻测线关系示意图

过 O 点做坡度的平行线，过 G 点做 OC 的平行线，这两点直线的交点记为 H 。由于对边两两平行，所以该四边形是平行四边形，因此对边长度相等， $OH = CM$ 。下面开始求解 OH 的长度：

在 $\triangle OHG$ 中，我们知道 $OG = d$ ， $\angle OGH = \frac{\pi - \theta}{2}$ ， $\angle OHG = \frac{\pi + \theta - 2\alpha}{2}$ ，同样利用正弦定理，就可以求得 $d' = OH$ 的长度：

$$d' = d \cdot \frac{\sin \frac{\pi - \theta}{2}}{\sin \frac{\pi + \theta - 2\alpha}{2}} \quad (5)$$

根据式(4) (5)即可得到任意测线的重叠率 η 的值。

5.1.4 模型求解

利用北太天元代入所给数据计算问题一的结果，用 excel 保留两位小数。最终结果如下表所示：

表 1 问题一的计算结果

测线距中心 点处的距离 /m	-800	-600	-400	-200	0	200	400	600	800
海水深度/m	90.95	85.71	80.47	75.24	70.00	64.76	59.53	54.29	49.05
覆盖宽度/m	315.81	297.63	279.44	261.26	243.07	224.88	206.70	188.51	170.33
与前一条测	——	35.70	31.51	26.74	21.26	14.89	7.41	-1.53	-12.36

测线距中心 点处的距离 /m	-800	-600	-400	-200	0	200	400	600	800
线的重叠率 /%									

从上面的计算结果中可以发现：当测线距离海域中心点的距离逐渐增大时，该位置处的海水深度会随之逐渐减小，多波束测深的覆盖宽度也会相应地变窄，而与前一条相邻测线之间的重叠率则不断降低。当测线距离中心点足够远时，重叠率甚至会出现负值，这意味着在实际测量中会出现漏测的情况。

这些参数的变化特征和最终出现的现象，与海洋测深作业中的实际情况高度吻合，进一步有力地验证了问题一所建立的数学模型在逻辑和推导上的合理性。

5.2 问题二模型的建立与求解

问题二是在问题一模型的基础上展开的，一方面将原本的平面坡度场景扩展到了更贴合实际的三维坡度情境，另一方面对航线方向进行了调整，同时新引入了“测线方向与海底坡面法向在水平面上投影的夹角 β ”这一关键参数。

而从问题一的计算过程中我们已经了解到，多波束测深的覆盖宽度与海底坡度及海水深度之间存在着紧密的关联，因此问题二中我们需要找到这两个参数的解析式，来推导出覆盖宽度的表达式。

5.2.1 计算坡度

问题二需要求解的坡度并不是指海底坡面与水平面的夹角 α ，而是特指与航线方向垂直的平面交海域形成的坡度，记为 γ 。

对于这一矩形待测海域，本文以航线在水平面上的投影建立三维坐标系，如图所示。

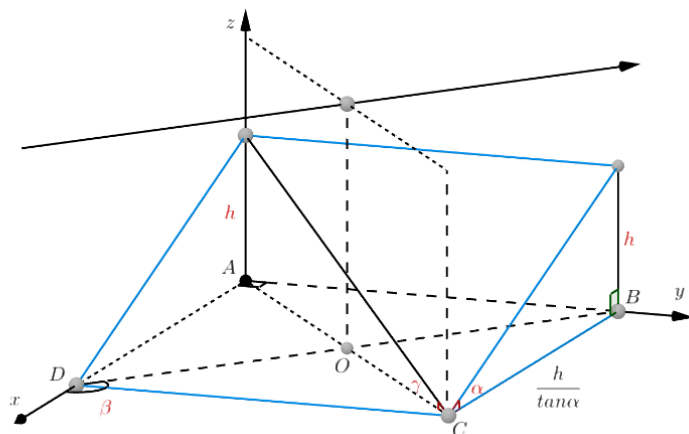


图 5 三维情形下测线示意图

引入一个参量 h ，记为海底坡面到水平面的距离，由此可以得到 $BC = \frac{h}{\tan\alpha}$ 。

已知测线方向与海底坡面的法向在水平面上投影的夹角 β ，可以得到 $\angle ADB$ 、 $\angle ABD$ ，通过正弦定理，得到：

$$CA = BD = \frac{AB}{\tan\angle BCA}, \tan\angle BCA = \beta - \frac{\pi}{2} \quad (6)$$

然后由勾股定理，求解出 AC 的长度。这时我们就能够得到坡度：

$$\tan\gamma = \frac{h}{AC} = \tan\alpha \cdot |\sin\beta| \quad (7)$$

5.2.2 计算深度

测量船从海域中心启航沿水平轴向右行驶，随着航行距离的增加，对应位置的海水深度逐渐减小。与问题一的求解思路相似，我们仍以海域中心处的水深为基准，来推导航线上其他位置的水深。

不难发现，由于航线与海平面保持平行，将测量船的航行路程投影到矩形海域平面上时，航线上的移动距离与平面投影距离 ON 是完全相等的。因此，测量船所处位置的海水深度变化值，本质上等同于 ON 到海底坡面的垂直距离。

因此，我们利用软件进行绘图，将测量船位置的变化投影到矩形海域上位置变化的情况，通过海域上位置坐标计算该位置下的测量船位置坐标。如下图所示：

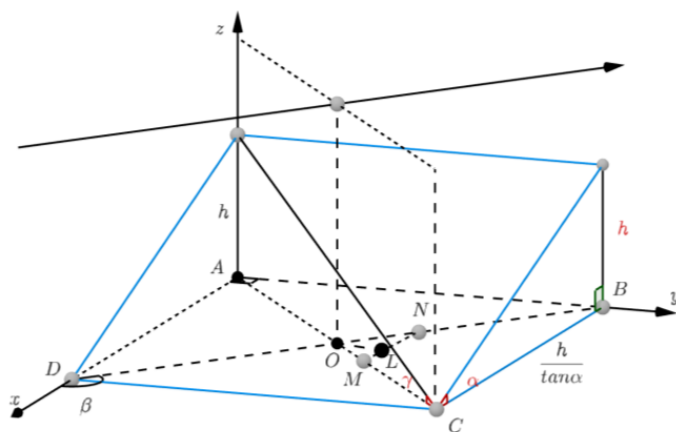


图 6 三维示意图

由图 6 所示，作 $MN \parallel BC$ ， $OL \perp MN$ ，则 $ln = ON\cos(\pi - \beta)$ ，接着利用海域深度求解得到 $x = l\tan\alpha$ ，最后利用中心深度得到航线上第 i 条航线的深度 $D_i = D - x$ 。

整理上述参量，得到覆盖宽度 W 与测线方向夹角和测量船距海域中心点处的距离之间的关系：

$$W = D_i \cdot \sin \frac{\theta}{2} \left(\frac{1}{\sin \left(\frac{\pi - \theta}{2} - \gamma \right)} + \frac{1}{\sin \left(\frac{\pi - \theta}{2} + \gamma \right)} \right) \quad (8)$$

5.2.3 求解结果

将该模型代入特定数据后，借助北太天元进行计算，成功求解出特定位置、特定角度下的覆盖宽度，具体结果如下表所示：

表 2 问题二的求解结果

覆盖宽度/m；测量船距海域中心点处的距离/海里

测 线 方 向 夹 角 / $^{\circ}$	0	0.3	0.6	0.9	1.2	1.5	1.8	2.1
--	---	-----	-----	-----	-----	-----	-----	-----

测 线 方 向 夹 角 / $^{\circ}$	0	0.3	0.6	0.9	1.2	1.5	1.8	2.1
0	415.692	466.091	516.490	566.889	617.288	667.686	718.085	768.484
45	416.192	451.872	487.552	523.232	558.912	594.592	630.273	665.953
90	416.692	416.692	416.692	416.692	416.692	416.692	416.692	416.692
135	416.192	380.511	344.831	309.151	273.471	237.791	202.110	166.430
180	415.692	365.293	314.894	264.496	214.097	163.698	113.299	62.900
225	416.192	380.511	344.831	309.151	273.471	237.791	202.110	166.430
270	416.692	416.692	416.692	416.692	416.692	416.692	416.692	416.692
315	416.192	451.872	487.552	523.232	558.912	594.592	630.273	665.953

对上述结果进行检验，首先 W 的值应当关于 $\beta = 180^{\circ}$ 对称，即测线关于坡度线对称时，覆盖宽度相等，满足条件；然后， $\beta = 90^{\circ}$ 或 270° 时，即测量船沿等深线移动时，任何位置上的覆盖宽度应该相等，满足条件；接着，当夹角方向为 $\beta \in \left[\frac{\pi}{2}, \frac{3\pi}{2}\right]$ 时， W 应随 d 的增大而减小，其他情况反之，满足条件；最后 $\beta = 180^{\circ}$ ，即条带宽度在同一高度，此时 $W = 2\sqrt{3}d$ ，满足条件。至此，模型检验完成，上述结果准确性极高。

5.3 问题三模型的建立与求解

5.3.1 模型建立

在问题二的基础上，问题三将进一步聚焦于实际海域的测线布局优化，基于已建立的覆盖宽度计算方法，结合具体海域的地形特征，探索如何通过合理规划测线间距与走向，在确保覆盖完整性和重叠率要求的前提下，实现测线总长度的最小化，从而形成更具实用性的测线设计方案。

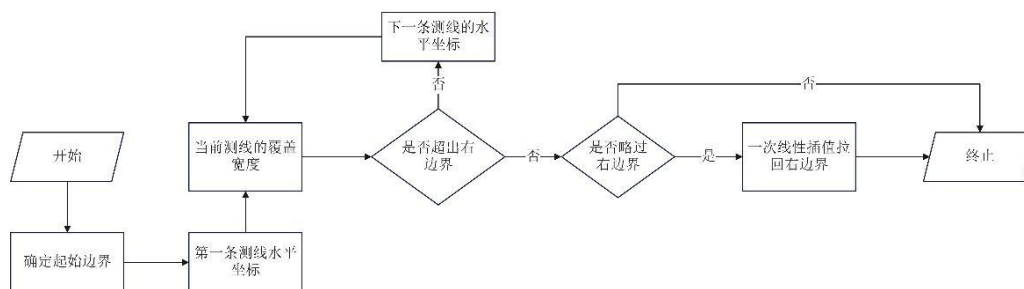


图 7 问题三求解流程图

通过查找文献[1]，在多波束测深条带测线中，存在这样的规律：当海水深度较浅时，容易出现漏测现象；而在水深较深的区域，则可能发生条带重叠过多的情况。

问题三所研究的矩形海域呈现着这样的地形：东西方向上水深西深东浅，南北方向上水深则保持一致。因此，在测量船由西向东航行的过程中，重点需在南北方向合理规划测线间距，以确保能够完全覆盖整个南北向海域。接下来，我们将针对东西方向上所需的测线长度进行求解：

为实现测线总长度最短的目标，我们需精准规划第一条测线的位置，使其覆盖范围的左边缘恰好与该矩形海域的左边界（即起始边界）重合，以避免因覆盖不足导致漏测或因过度覆盖增加冗余长度。具体示意图如下：

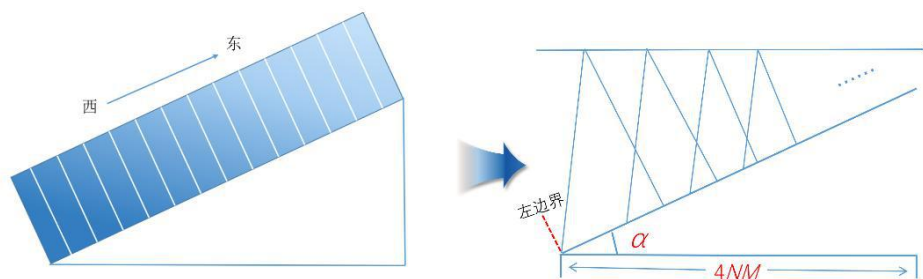


图 8 水深示意图

已知此时海底中心处的海水深度，从该中心处出发，求解在南北方向上的测线长度。求解过程如下：

由问题一对覆盖深度的求解，我们知道任意测线 w_i 、 W_i 公式。其中 i 表示第 i 条测线。

$$W_i = \frac{D_i \cdot \sin \frac{\theta}{2}}{\sin \left(\frac{\pi - \theta}{2} - \alpha \right)} \quad (9)$$

只要左边界与 w_i 相等，就能确保第一条线的左侧覆盖边缘恰好落在这条边界上，不会漏测也不过多冗余，因此建立以下公式：

$$W_i^{-3704m(10)}$$

从而我们确定了第一条测线处的海底深度。

本文采用递推法求解每条测线的横坐标, 通过式(1)求解得到第一条测线的水平坐标 x_i 。由上述等式我们可以得到 x_1 , 其随重叠率 η 的变化而变化。

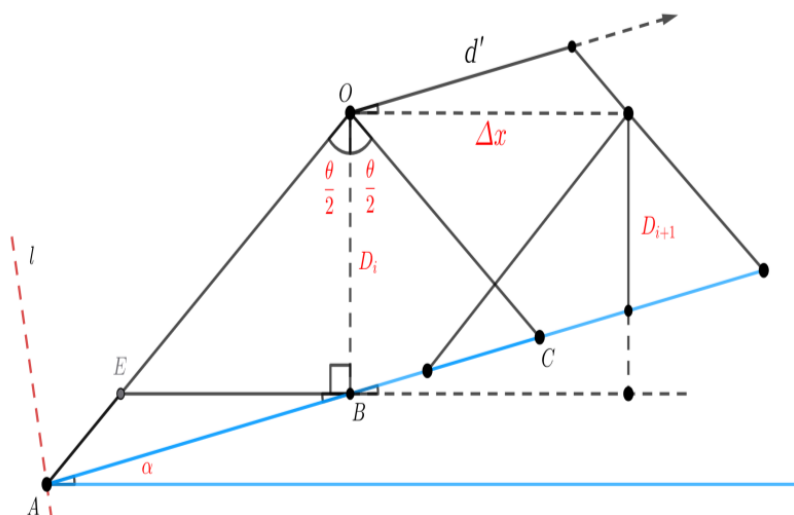


图 9 相邻条线关系示意图

如图所示, 设任意一条测线横坐标为 $x_i \in [-3704, 3704]$, 通过问题一求解出的海水深度计算公式 $D_i = D - x_i \tan \alpha$, 可以计算出该测线此时的水深。然后就可以求解得到测线的覆盖深度。已知第 i 条测线的重叠率 $\eta = 1 - \frac{D_i}{w_i}$ 满足 10%~20%, 结合 Δx 公式, 建立从第一条测线求解到下一条测线最终到每一条测线在东西方向上水平坐标的递推公式。据此, 模型建立完成。

5.3.2 模型求解

利用北太天元，把问题三相关数据代入到建立的模型中，重叠率 $\eta \in [0.1, 0.2]$ ，对重叠率 0.01 为一个步长进行求解。绘制不同的 η 与当前最小测线数 n 的关系图如下：

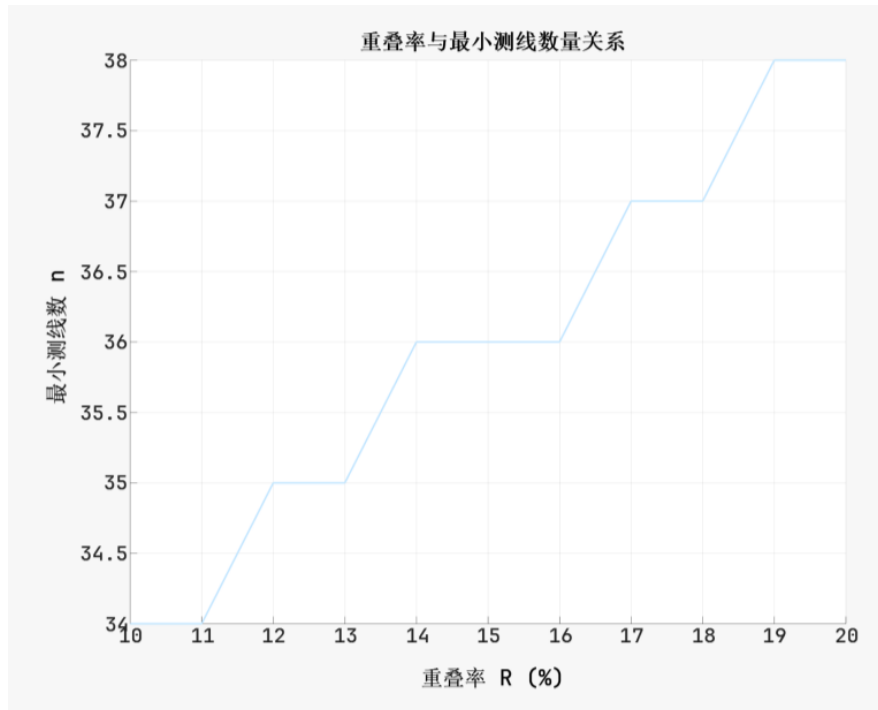


图 10 重叠率与最小测线数的关系图

从图中我们可以发现，随着重叠率的增加，所需要的测线数量整体上呈上升趋势，而在最小测线数出现在重叠率 $\eta \in [0.10, 0.11]$ 中。因此我们得到问题三至少需要 34 条测线，总长度为 68 海里。利用 matlab 绘出这 34 条测线在重叠率为 0.105 时的水平分布图，如下：

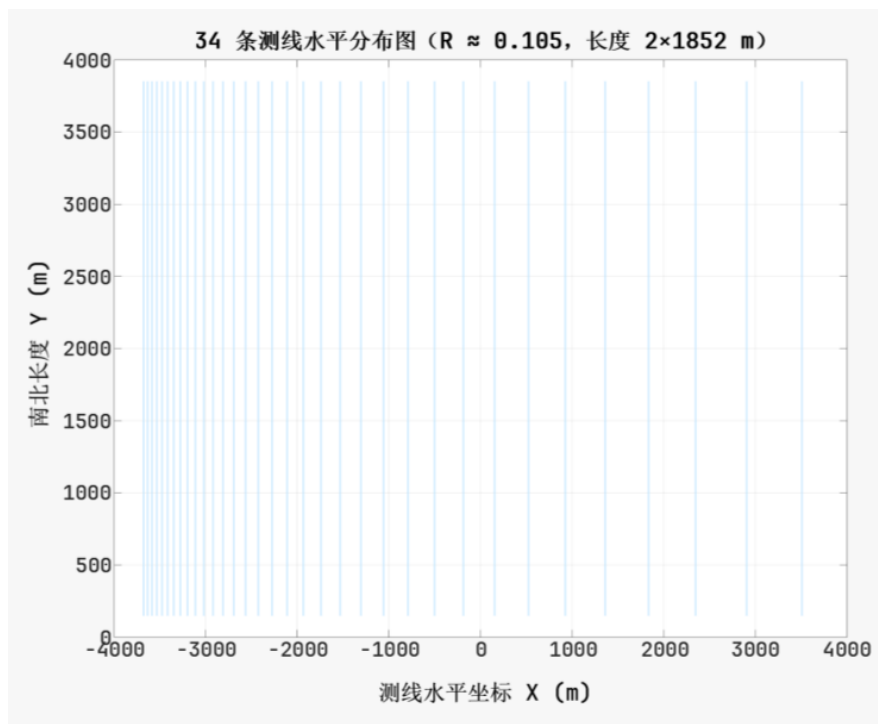


图 11 最小测线数的水平分布图

对上图进行分析得到，由西向东，测线之间的距离逐渐增大，由问题一的结果分析我们知道两条测线等距的情况下，随着测量船行走的距离越远，其重叠率就越小，最终出现漏测的情况。因此问题三中相邻测线之间的距离变大是符合实际情况的。且每一条测线都是沿南北方向并且长度相等。所以我们设计的这一组测线是满足实际情况的，总长度最短且能够全覆盖此海域。

5.4 问题四模型的建立与求解

在问题四中，我们需要根据附件中所给的海域深度数据，去设计测线布局，要求要尽可能的覆盖整个海域，相邻之间的重叠率尽量在 20% 以下，且测线总长度尽可能短。由于该海域的地形起伏较大，需要划分区域分别分析。

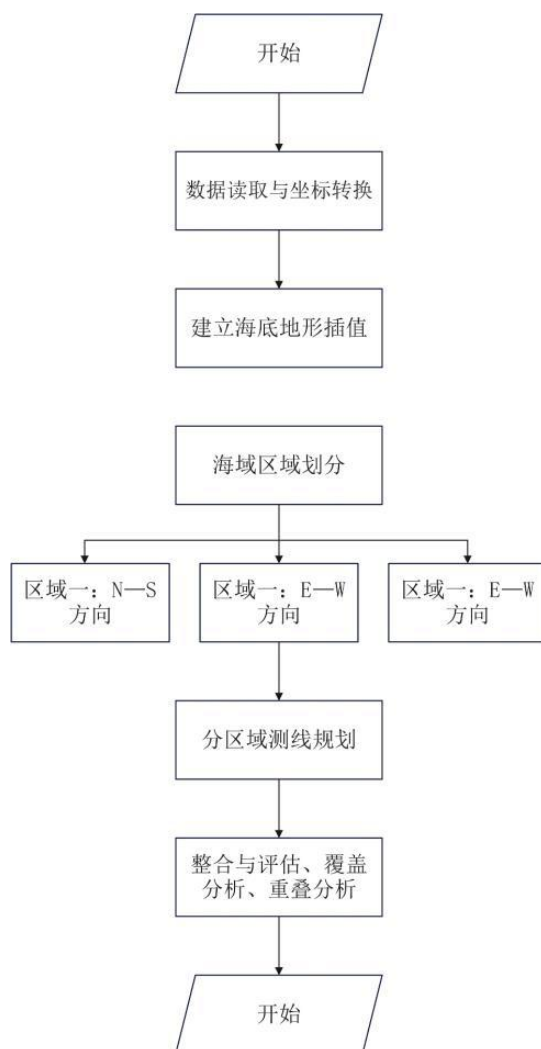


图 12 问题四流程图

5.4.1 整合数据得到海域面貌

我们根据所给的单波束测量的海水深度数据，以自西向东为 x 轴正方向，自南向北为 y

轴正方向，海水深度自下而上为 z 轴正方向，对其进行取负数，为了观察到海域面貌，我们利用北太天元绘制海域的三维地形图，如下：

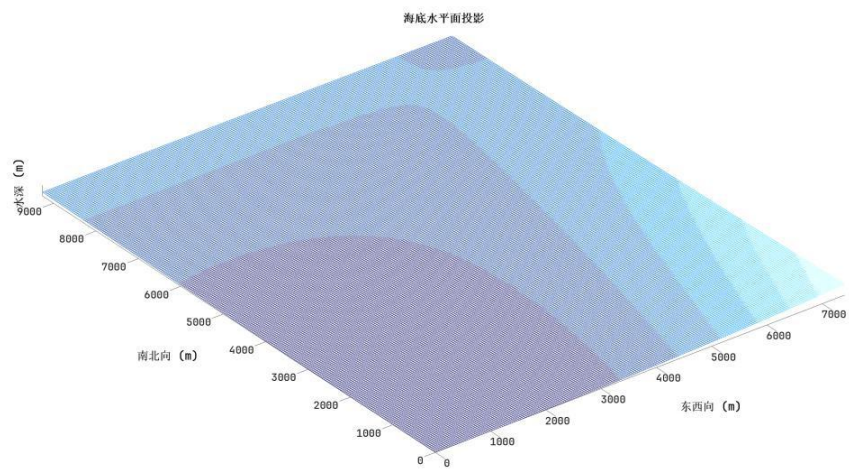


图 13 海底水面投影

为了更加直观的观察海域深度，我们又绘制了等深线地形图，如图：

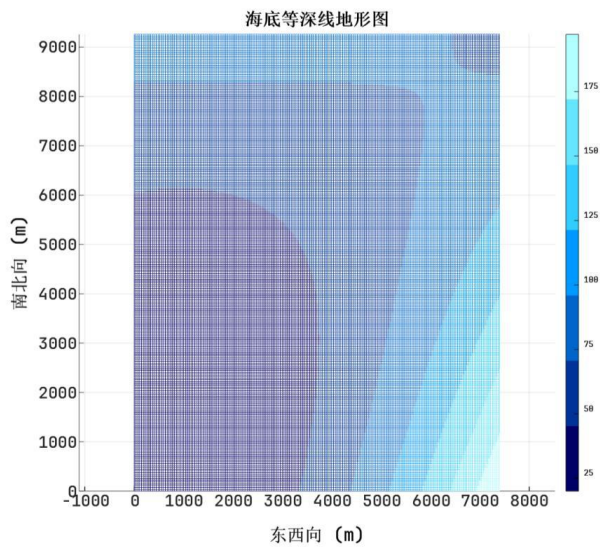


图 14 海底等深地形图

分析海底三维地貌图和海底等深线图，我们可以看到该海域的起伏较大，整体呈现为一个不规则的小山丘。其中，海域东南角等深线密集，海底地形陡峭，海水深度较大；东北角等深线较为稀疏，地形较为平坦；西南方海水深度较小；西北角的地形大致为一个坡面；整个海域及其不规则，为我们的测线布设增加了难度。

5.4.2 划分区域

因为海域地形及其不规则，且相邻之间起伏大，如果看作一个整体分析，在布设测线时，

会在海水深度较小处出现漏测，在海水深度较大处会导致相邻条带之间重叠率过大，测量效果不好，且误差极大。所以我们在前面的基础上，可以人为的划分区域，我们依据矩形的一边尽可能的与所围等深线平行，把海域划分为三个矩形小区域，在每个小区域里分别布设测线，如图：

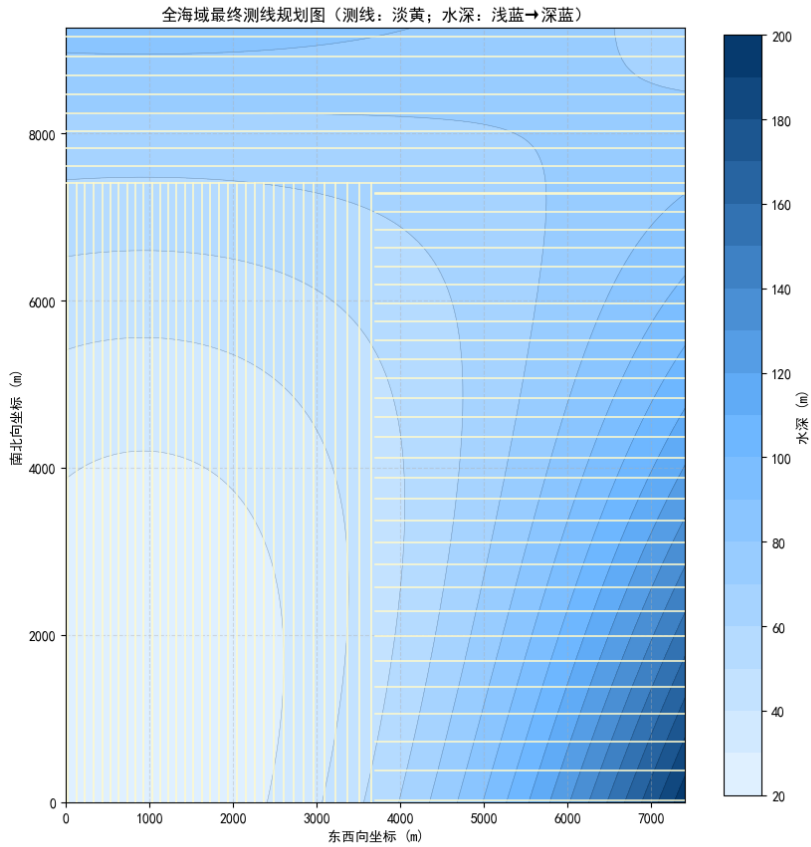


图 15 海底等深地形图

如图，我们把海域平铺为一个东西长 4 海里，南北长 5 海里的矩形，把西南角的顶点作为原点，自西向东为横向坐标，自南向北为纵向坐标，一个单位长度为 1 海里，建立直角坐标系。由横坐标 0.0-2.0 及纵坐标 0.0-4.0 所围的区域划为第一个矩形，测线布设方向为纵向方向从南到北；由横坐标 2.0-4.0 及纵坐标 0.0-4.0 所围的区域划为第二个矩形，测线布设方向为横向方向从西到东；剩下的区域横坐标 0.0-4.0 及纵坐标 4.0-5.0 所围的区域为第三个矩形，测线布设方向为横向方向从西到东。

我们可以对三个矩形区域分别进行分析，设计测线布局方案，再整合在一起，作为整个海域的测线布局方案。

5.4.3 坡面拟合

对于前面所推出来的公式，我们都是建立在一个坡面的前提下使用，对于这片不规则的海域，我们可以通过附件中的数据拟合出坡面矩形，然后再使用前文的公式计算。

在三维坐标系中，设海底矩形平面方程为：

$$z = Ax + By + C \quad (11)$$

其中 x, y 为附件所给点的坐标， z 为该点的海水深度。

我们可以由几何知识知道坡面与海平面的夹角 α ：

$$\alpha = \arccos\left(\frac{-1}{\sqrt{A^2 + B^2 + 1}}\right) \quad (12)$$

对于参数 A, B, C 的计算，我们使用附件中所给的数据，用最小二乘法拟合，建立目标规划模型，如下：

$$\begin{aligned} & \min \frac{\sum_{i=1}^n (z - z_i)^2}{n} \\ & s. t. \begin{cases} A \in [-100, 100], \\ B \in [-100, 100], \\ C \in [-100, 100]. \end{cases} \end{aligned} \quad (13)$$

我们将划分的三个区域分别拟合一个坡面，在三个区域上可以直接用前文的模型公式求解。

5.4.4 建立单目标最优规划模型

我们在第三问的基础上，继续将最短测线长度定为主要优化目标，即将最短测线长度设为目标函数，即：

$$\min n l_1 \quad (14)$$

这里的 n 表示在划分的矩形区域内布设的测线条数， l_1 表示在划分的矩形区域的长度，即 $l_1 = 4$ 或 2 。

将其余的要求设为约束条件，即尽可能覆盖全部面积为：

$$\sum_{i=1}^m W_i - \sum_{i=2}^m W_i \eta_i \geq \frac{l_2}{\cos \alpha} \quad (15)$$

其中 l_2 为各个矩形区域的宽度， $l_2 = 1$ 或 2 。

对于重叠率，要求尽可能控制在 20% 以下。由前文分析可知，在一定范围内，重叠率越小，所需测线总长度越短；而行业一般标准要求重叠率在 10%—20% 之间，综合考量，我们将重叠率固定为 10%。

关于多波束开角，题目中未给出具体数值。但不难得知，开角大小对测宽影响显著：开角增大时，若海底坡度也较大，测宽会明显变大，可能在地形倾斜较缓的区域导致重叠过多，产生误差；开角减小时，测宽会相应变小，可能需要布设更密集的航线，进而使航线分布发生细微变化。因此，开角的选择需结合具体地形合理确定，在本题中，我们仍沿用前文设定的开角值 $\theta = 120^\circ$ 作为标准。

在此基础上，我们将前文推导得出的各项计算公式整合进来，即可得到完整的模型表达式：

$$s. t. \left\{ \begin{array}{l} \min n l_1 \\ \sum_{i=1}^m W_i - \sum_{i=2}^m W_i \eta_i \geq \frac{l_2}{\cos \alpha}, \\ \theta = 120^\circ, \\ \eta = 10\%, \\ d_i' = (1 - \eta) W_i, \\ d_i = d_i' \cdot \sin\left(\frac{\pi + \theta}{2} - \alpha\right) \div \sin\left(\frac{\pi - \theta}{2}\right), \\ D_i = D - d_i \tan \alpha, \\ W_i = D_i \cdot \sin \frac{\theta}{2} \left(\frac{1}{\sin\left(\frac{\pi - \theta}{2} - \alpha\right)} + \frac{1}{\sin\left(\frac{\pi - \theta}{2} + \alpha\right)} \right). \end{array} \right. \quad (16)$$

我们分别在三个矩形区域内都建立优化模型，再整合得到整个海域的测线布设。

在此基础上，我们可以得到每个矩形区域测线的总长度，即：

$$L = n \cdot l_1 \quad (17)$$

漏测海区占总待测海域面积的百分比：

$$\begin{aligned} P &= 1 - \frac{S_{\text{漏}}}{S_{\text{总}}}, \\ S_{\text{总}} &= 4 \times 5 \times 1852 \times 1852 = 68598080, \\ S_{\text{漏}} &= \left(\sum_{i=1}^m W_i - \sum_{i=2}^m W_i \eta_i \right) \cdot l_1. \end{aligned} \quad (18)$$

重叠区域中，重叠率超过 20% 部分的总长度：

$$l_{重} = W_i \cdot \eta_i (19)$$

我们再把三个区域的情况整合，就可以得到整片海域的情况。

5.4.5 模型求解

算法步骤：

Step1: 数据读取与插值

1. 读取 Excel，解析得到每个测深点的 (x, y, z) ，将海里单位转换为米。
2. 建立深度插值器，在每次需要深度、坡度的地方都调用此插值器。

Step2: 区域划分

将待测海域划分成 3 个区域，设定不同测区的方向偏好。

Step3: 单个区域内规划测线

1. 在法向坐标上扫描候选中心线位置来找首条测线：对候选中心计算整条测线的平均深度和平均法向坡度；计算 W_1, W_2, W_3 ，当左/下侧能覆盖边界时，选为首条测线，这样的策略保证边界不会漏测。
2. 从首条测线开始，迭代生成后续测线：对当前测线计算局部平均 w ；设置下一条测线的中心，即用保守步进，避免太密或太稀；当超出区域时，判断是否当前测线右侧已覆盖边界，是则结束；循环直到覆盖到区域尽头或遇到无效深度。

Step4: 局部采样与网格覆盖判定

1. 对每条规划好的测线，在它的航向方向等距采样。
2. 对每个采样点计算局部深度和局部法向坡度，再计算该采样点的局部 W_1, W_2 ，若插值失败则标记为无效。
3. 建立全海域的覆盖网络，每个网格以中心点代表。
4. 对每条测线：找出沿航向最接近该网格点的采样点索引，若该采样点无效则在附近向两侧寻找最近的有效采样点；计算网格点到该测线中心线的横向距离，比较该采样点的对应半宽来判断是否覆盖。

Step5: 重叠统计

1. 对每个区域按已规划测线顺序，计算相邻两条线的近似几何重叠。
2. 对参数进行调优处理。

求解结果：

表 3 各区域的测线总数

区域	测线总条数
1	34
2	30
3	9

利用北太天元，三个区域的测线布局如图所示：

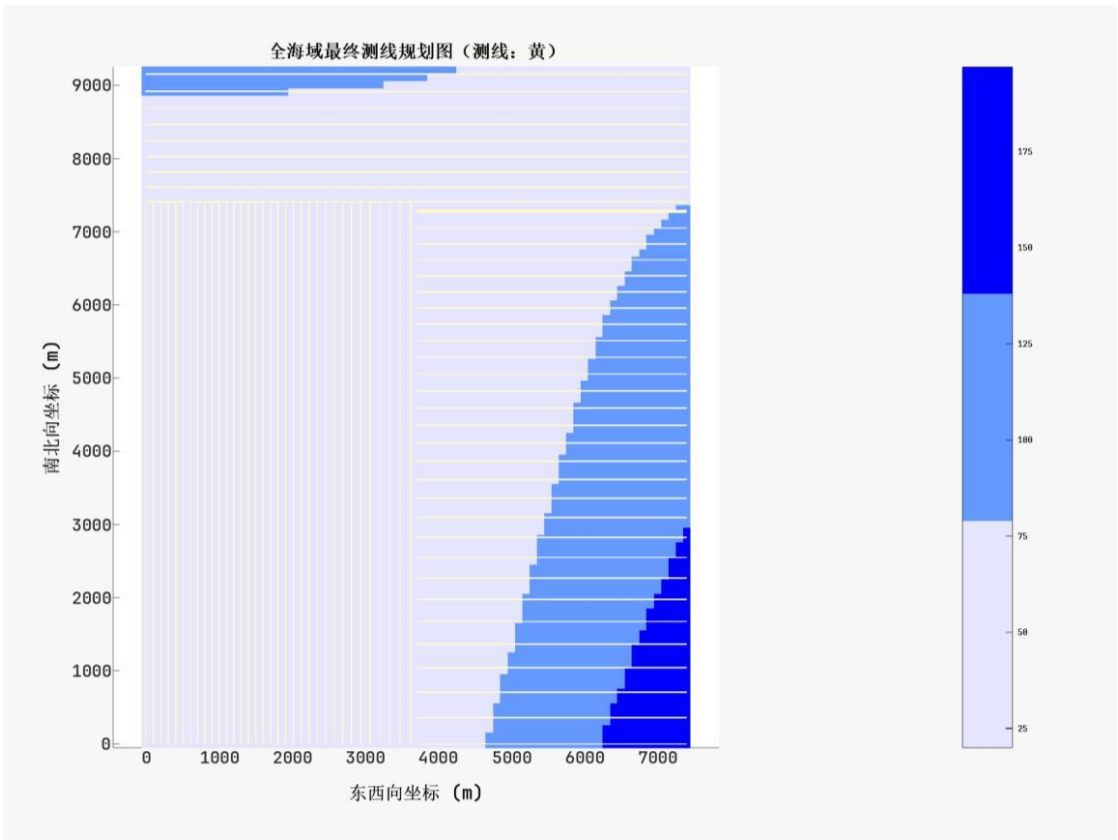


图 16 全海域测线布设图

得到结果如下表所示：

表 4 指标结果

指标	结果
测线总长度	429.66km (232.00 海里)
漏测海区面积占比	6.0574%

指标	结果
重叠率>20%部分的总长度	3.70km

尽管模型通过区域划分和坡面拟合进行了一定程度的简化，导致在子区域交界处或地形突变点存在微小误差，但其在全局层面上成功平衡了计算复杂度与结果精度，在保证最短测线总长度（429.66km）下，实现了较高的覆盖率（漏测仅 6.0574%）和较低的超标重叠长度（3.70km），证明了其策略的有效性和模型的整体合理性。该模型不仅为本题提供了切实可行的解决方案，其“分治-拟合-优化”的核心思想也为处理同类复杂环境下的测绘规划问题提供了有价值的参考范式。

六、模型的评价与推广

6.1 模型的优点

基于数学思维，对于覆盖宽度，重叠率的计算给出了确切的公式，计算精确，并且进行推广得出了符合实际的结果，定义、模型合理准确。

对于几个问题的模型建立具有可传递性，模型层层递进，通过增加变量和增加复杂度来使模型由特殊到一般，使模型更加有层次。

在建立模型时将复杂问题转化为可以直观呈现出的几何问题，化繁为简。

6.2 模型的缺点

在问题四中，为了简化模型，我们对海域进行划分，且各区域地形拟合为坡面，从而导致求解结果不精确。

忽略了测量船行进过程的横摇和纵摇误差，与现实可能存在偏差。

6.3 模型的推广

对于覆盖宽度，重叠率的计算模型，以及测线的设计模型可以推广使用到实际生活中的多波束测量中去，根据实际情况修正后可应用到海洋测绘领域中。

开角可以切换为更多情况，用于建筑扫描，无人机地形扫描等多种情况。

七、参考文献

[1] 李慧君, 张乾, 刘琳, 等. 多波束测深技术在测线问题中的优化研究[J]. 周口师范学院学报, 2024, 41(05): 26-32. DOI:10.13450/j.cnki.jzknu.2024.05.004.

[2] 葛平茹, 张书畅, 申文莹, 等. 基于计算几何的多波束测线模型[J]. 自动化应用, 2024, 65(08): 4-8. DOI:10.19769/j.zdhy.2024.08.002.

[3] 张泽宇, 葛晨欣, 靳琪, 等. 基于多目标规划的多波束测线布设模型[J]. 实验科学与技术, 2025, 23(02): 97-103.

基于交易票据的中小微企业信贷风险量化与策略优化研究

作者: 王心悦 王 凯 孙浩莹

指导教师: 梁冬冬

学校: 西安理工大学

摘要: 中小微企业是我国市场经济的核心主体, 却因规模有限、抵押物不足等问题面临融资困境, 而商业银行需在安全性与盈利性间平衡, 为其提供合理信贷支持。本文基于企业交易票据数据, 围绕中小微企业信贷风险量化与策略优化展开研究。

首先, 针对 123 家有信贷记录企业, 通过数据清洗, 处理缺失值、异常值, 完成特征工程, 构建多元非线性回归模型量化信贷风险。经 t 检验、 F 检验、多重共线性诊断及五折交叉验证, 模型有效性与稳健性得到验证, 据此将企业划分为极低、低、中、高、极高五个风险等级, 并制定信贷策略, 实际放贷总额 5000 万元, 覆盖 94 家企业。

其次, 对 302 家无信贷记录企业, 补充行业与企业类型变量, 以“异常销项占比”为风险代理指标, 构建 10 参数多元非线性回归模型。同样经系列检验验证模型可靠性后, 量化企业风险并在 1 亿元年度信贷总额约束下制定策略, 放贷覆盖 294 家企业, 且分行业策略符合各行业风险特性与实际规律。

最后, 考虑新冠疫情突发因素, 将企业按行业划分为五类疫情敏感度, 设定影响系数与强度, 修正风险模型得到 5 参数模型。基于修正模型重新量化 294 家企业风险, 发现商贸服务类、建筑工程类风险上升, 医药医疗类等风险下降, 进而调整信贷策略, 在 1 亿元总额约束下完成放贷, 实现“高风险行业收缩、低风险行业倾斜”的风险管控目标。研究为商业银行精准评估中小微企业信贷风险、优化信贷资源配置提供科学依据。

关键词：中小微企业；信贷风险量化；交易票据数据；多元非线性回归模型；策略优化；疫情冲击

点 评：本文选题具有鲜明的现实意义，紧扣中小微企业融资难这一社会痛点。建模框架完整，从有信贷记录到无信贷记录企业，再到疫情冲击下的策略调整，层层递进、逻辑严密。在方法上，采用多元非线性回归模型，辅以 t 检验、F 检验、多重共线性诊断和五折交叉验证，统计检验手段全面，模型可信度较高。特别是将企业划分为五个风险等级并针对性制定信贷策略，兼顾了风险控制与信贷覆盖率的平衡。疫情冲击的敏感性分析进一步增强了模型的实用价值。建议可引入更多维度的企业特征（如供应链数据、税务数据）来进一步提升模型的预测精度，同时在策略优化中考虑更细粒度的利率定价模型。

基于交易票据的中小微企业信贷风险量化与策略优化研究

摘要

中小微企业是我国市场经济的核心主体，却因规模有限、抵押物不足等问题面临融资困境，而商业银行需在安全性与盈利性间平衡，为其提供合理信贷支持。本文基于企业交易票据数据，围绕中小微企业信贷风险量化与策略优化展开研究。

首先，针对 123 家有信贷记录企业，通过数据清洗，处理缺失值、异常值，完成特征工程，构建多元非线性回归模型量化信贷风险。经 t 检验、 F 检验、多重共线性诊断及五折交叉验证，模型有效性与稳健性得到验证，据此将企业划分为极低、低、中、高、极高五个风险等级，并制定信贷策略，实际放贷总额 5000 万元，覆盖 94 家企业。

其次，对 302 家无信贷记录企业，补充行业与企业类型变量，以“异常销项占比”为风险代理指标，构建 10 参数多元非线性回归模型。同样经系列检验验证模型可靠性后，量化企业风险并在 1 亿元年度信贷总额约束下制定策略，放贷覆盖 294 家企业，且分行业策略符合各行业风险特性与实际规律。

最后，考虑新冠疫情突发因素，将企业按行业划分为五类疫情敏感度，设定影响系数与强度，修正风险模型得到 5 参数模型。基于修正模型重新量化 294 家企业风险，发现商贸服务类、建筑工程类风险上升，医药医疗类等风险下降，进而调整信贷策略，在 1 亿元总额约束下完成放贷，实现“高风险行业收缩、低风险行业倾斜”的风险管控目标。研究为商业银行精准评估中小微企业信贷风险、优化信贷资源配置提供科学依据。

关键词：中小微企业；信贷风险量化；交易票据数据；多元非线性回归模型；策略优化；疫情冲击

一、问题重述

1. 背景知识

中小微企业作为我国市场经济的核心主体，在激发经济活力、提升市场效率及制衡垄断势力中发挥关键作用。依据谢勒（*Scherer*）等学者的理论，相较于大型企业，中小微企业因经营决策灵活、创新成本较低，具备更强的创新驱动力，是推动技术进步的重要力量。然而，受规模有限、抵押物不足及经营稳定性欠佳等因素制约，中小微企业普遍面临融资困境，银行贷款获取难度大，严重制约其成长发展。

从商业银行经营视角看，安全性、流动性与盈利性是核心原则，其中安全性为稳健经营

基石，盈利性为持续发展目标。出于风险控制考量，银行对中小微企业贷款通常持谨慎态度；但基于政策导向与业务拓展需求，又需为其提供金融支持。在缺乏足额抵押物的情况下，银行仅能依托企业利润、资金流水、经营规模及行业影响力等有限信息，评估中小微企业信贷风险并筛选低风险企业放贷。这一过程既关乎中小微企业融资可得性，亦是银行实现安全性与盈利性平衡的关键。

本文旨在基于企业发票数据，深度挖掘企业盈利能力、资金流动性、企业规模及信用状况等经营特征，构建科学的企业信贷风险评价模型以实现风险量化评估；同时结合银行经营原则与放贷政策，建立银行最优放贷决策模型，为信贷资源合理配置提供决策依据。

2. 数据分析

本题一共给出了三个附件数据，本文将对这些附件进行初步的解读。

附件一：给出了 123 家已有信贷记录的企业数据，由三部分组成。第一部分是各企业的信贷记录，包括信用评级和是否违约。第二部分是各企业的进项发票数据，包括了进项发票号码、开票日期、销方单位代号、金额、税额、价税合计和发票状态。第三部分是各企业的销项发票数据，内容和进项发票数据相同。

附件二：给出了 302 家无信贷记录的企业数据，与附件一最大的不同在于不包括企业的信贷记录，仅仅包含各企业的发票数据，指标内容和附件一相同。

附件三：给出了银行年利率和客户流失率的关系，其中客户流失率又根据信用等级不同而不同。初步来看，利率和客户流失率呈正相关关系，且在相同利率水平下，A 类企业的流失率较高。

3. 具体问题

1. 对附件一中有信贷记录的 123 家企业的信贷风险进行量化分析，并建立模型判断银行对每个企业的信贷决策。
2. 根据附件一的数据来预测 302 家无信贷记录企业的信用等级和违约概率，通过问题一构建的模型来分析各企业的信贷风险和银行的最优信贷决策。
3. 在前两问的基础上考虑突发因素的影响，对模型进行改进。通过改进后的模型来分析附件 2 中 302 家企业的信贷风险和银行的最优信贷决策有何变化，并给定具体事件进行具体分析。

二、问题分析

1. 对问题一的分析

第一问中需要对有信贷记录的 123 家企业的信贷风险进行量化分析，并要求给出银行对

各企业的信贷决策。信贷决策包括是否放贷、贷款利率和贷款金额。首先根据信贷记录计算出各企业信贷风险来判断是否进行放贷，其中违约概率在附件一是一个 0-1 变量，针对这种二值变量可以通过使用多元非线性 *Logit* 回归，利用附件一中的数据，用最小二乘法估计出参数，然后根据信誉越高利率越低的原则将适宜放贷的企业进行分类，确定其贷款利率。最后，根据利润最大化和风险最小化的经营原则建立银行最优信贷决策模型，依据上文给出的企业信贷风险和利率计算出各个企业的最优额度。

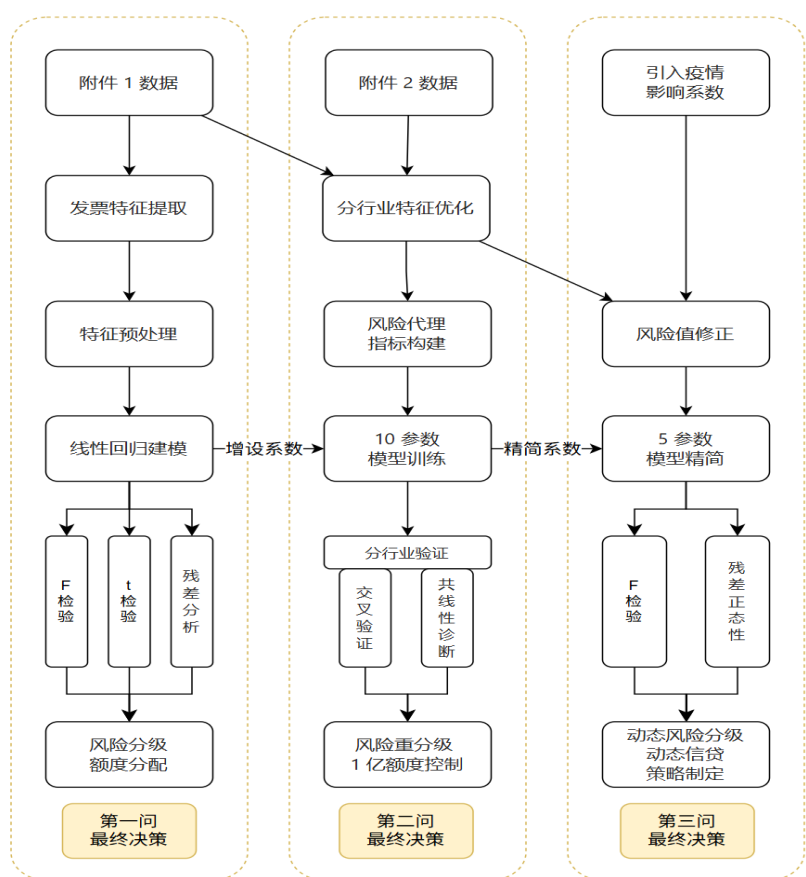
2. 对问题二的分析

问题二需要根据附件 2 给出的 302 家企业的信息计算各企业的信贷风险，并在年度信贷总额为 1 亿的条件求解银行对于各企业的信贷决策。问题 1 模型仅适用于“有违约标签”企业，问题 2 无违约数据，需修正模型形式，即新增企业类型和行业变量，并替换风险代理指标，即用“异常销项占比”替代违约概率，并通过问题 1 模型系数迁移实现风险量化。针对无信贷记录企业的交易数据特性，采用分行业异常值处理；同时验证发票数据质量，确保交易特征计算准确。因无“是否违约”标签，用异常销项占比作为风险代理变量；构建 10 参数多元非线性回归模型，通过统计检验筛选核心显著变量。在问题一策略框架基础上强化总额约束，使策略合规且贴合银行经营目标。

3. 对问题三的分析

第三问中主要是对上述的决策模型进行改进和优化，加入对于新冠疫情这一突发因素的考虑，核心是“量化行业冲击-修正风险模型-调整信贷策略”，确保策略在风险变化下仍满足 1 亿元总额约束。基于新冠对中小微企业的实际冲击，将附件 2 企业按行业划分为 5 类敏感度：①高敏感②中敏感③中低敏感④低敏感⑤负敏感，同时设定疫情影响强度。在问题二模型基础上加入疫情冲击项，并修正公式。通过残差分析验证修正合理性，确保风险量化贴合疫情下的企业实际经营状况。按修正后风险等级动态调整：①额度调整；②利率调整；③总额控制。最终实现“高风险行业收缩、低风险行业倾斜”，符合突发因素下的风险管控需求。

4. 技术路线图



三、模型假设

1. 数据有效性假设

公司的有效发票信息（含进项/销项规模、频率及作废率）可作为反映其经营稳定性与交易规模的核心指标，但需结合银行内部信用评级综合评估经营状态。

2. 外生变量合理性假设

企业是否违约是影响下期信誉评级的核心决定性因素，同时评级还受当期经营指标、行业环境变化等因素的动态调整。

突发事件的发生带来的影响将持续到贷款期结束，呈衰减趋势。

3. 贷款利率相关性假设

贷款利率和公司信誉和实力成正相关，与信贷风险成负相关，高利率可能导致顾客流失。

4. 信贷决策独立性假设

在年度总额固定的短期决策中，当期信贷决策和下期信贷决策相互独立。

四、符号说明

符号	含义	符号	含义
y	违约概率	z	风险值(异常销项占比)

x_1	信誉评级数值	α_0	z_截距项
x_2	年销项总额_万元(中心化)	α_1	z_行业类型编码系数
x_3	销项稳定性_变异系数(中心化)	α_2	z_企业类型编码系数
x_4	进项销项比(中心化)	α_3	z_c_年销项总额_万元系数
x_5	作废发票占比(中心化)	α_4	z_c_年销项总额_万元_2系数
x_6	行业类型编码	α_5	z_c_销项稳定性_变异系数系数
x_7	企业类型编码	α_6	z_c_销项稳定性_变异系数_2系数
β_0	y_截距项	α_7	z_c_进项销项比系数
β_1	y_信誉评级数值系数	α_8	z_c_进项销项比_2系数
β_2	y_c_年销项总额_万元系数	α_9	z_c_作废发票占比系数
β_3	y_c_年销项总额_万元_2系数	α_{10}	z_c_作废发票占比_2系数
β_4	y_c_销项稳定性_变异系数系数	k	疫情影响系数
β_5	y_c_销项稳定性_变异系数_2系数-	α	疫情影响强度
β_6	y_c_进项销项比系数	$z_{\text{修正}}$	疫情后信贷风险值
β_7	y_c_进项销项比_2系数		
β_8	y_c_作废发票占比系数		
β_9	y_c_作废发票占比_2系数		

五、模型的建立与求解（问题一）

1. 数据清洗

1.1 缺失值处理

1.1.1 企业信息清洗

a. 企业代号标准化:处理前 123 个唯一值, 处理后 123 个唯一值

b. 缺失值情况（处理前）：

信誉评级分布: {'B':38, 'C':34, 'A':27, 'D':24}

违约情况分布: {0:96, 1:27}

企业信息清洗后保留:123 条记录

1.1.2 发票数据清洗

a. 进项发票处理:

发票状态分布: {'有效发票':203339, '作废发票':7608}

清洗后保留进项发票:210947 条(原始:210947 条)

b. 销项发票处理:

发票状态分布: {'有效发票':151278, '作废发票':11159, '作废发票':47}

清洗后保留销项发票:162484 条(原始:162484 条)

1.1.3 数据清洗汇总

a. 企业信息:原始 123 家, 处理缺失后 123 家, 移除缺失企业 0 家

b. 进项发票:原始 210947 条, 清洗后 210947 条

c. 销项发票:原始 162484 条, 清洗后 162484 条

1.2 特征工程执行

a. 有效销项发票:142854 条(占总销项发票的 87.92%); 有效进项发票:201479 条(占总进项发票的 95.51%)

b. 合并企业信息与销项特征后:123 条记录;合并进项特征后:123 条记录;合并作废发票占比后:123 条记录

c. 进项销项比计算:处理了 0 个无穷值和 0 个缺失值

1.3 异常值处理

a. 销项稳定性变异系数:处理了 0 个极端值

b. 年销项总额_万元:移除 2 个异常值(范围:[-134378.53, 159873.97])

c. 销项稳定性_变异系数:移除 3 个异常值(范围:[-1.24, 3.06])

d. 进项销项比:移除 4 个异常值(范围:[-4.35, 5.95])

e. 作废发票占比:移除 4 个异常值(范围:[-0.16, 0.32])

f. 移除剩余缺失值:0 条记录

汇总: 异常值处理共移除:13 条记录(原始:123, 处理后:110)

(特征工程完成, 最终建模数据量:110 家企业)

2. 构建多元非线性回归方程并评估模型

2.1 多元非线性回归方程的构建

2.1.1 模型假设与形式选择

由于信贷风险与解释变量的关系呈非线性, 选择二次多项式回归模型, 形式如下:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_3 + \beta_5 x_3^2 + \beta_6 x_4 + \beta_7 x_4^2 + \beta_8 x_5 + \beta_9 x_5^2 + \varepsilon \quad (1)$$

其中: Y : 信贷风险值 (0-1)

$\beta_0 - \beta_9$: 回归系数

ε : 随机误差项 (服从正态分布)

2.1.2 变量中心化处理

为避免多重共线性, 对连续变量进行中心化。

2.1.3 回归建模与参数估计

先构建回归矩阵, 再通过最小二乘法进行非线性回归求解。

在北太天元中求得该回归方程为

$$y = -0.38 + 0.23x_1 + 0.00x_2 + 0.00x_2^2 - 0.05x_3 - 0.07x_3^2 + 0.07x_4 + 0.02x_4^2 - 0.84x_5 + 11.38x_5^2 \quad (2)$$

2.2 多元非线性回归模型的评估

2.2.1 回归系数 t 检验

对于本文构建的企业的各项特征指标对企业的信贷风险进行量化分析的多元线性模型, 通过 t 检验可以分别判断各项特征指标对企业的信贷风险的单独影响是否显著, 从而为进一步优化模型和深入理解变量间的关系提供依据。

t 检验值的计算公式为

$$t_j = \frac{\hat{a}_j}{SE(\hat{a}_j)} \quad (3)$$

其中 α_j 是自变量 x_j 回归系数的估计值， $SE(\hat{\alpha}_j)$ 是该估计值的标准误差，结果如下表。

表 1 回归系数的 t 检验参数表

变量	系数值	标准误差	t 统计量	p 值	显著性
β_0	-0.383195	0.07760	-4.937859	0.00000	p<0.01
		4		3	
β_1	0.227089	0.03286	6.910191	0.00000	p<0.01
		3		0	
β_2	0.000000	0.00000	0.092808	0.92624	
		4		2	
β_3	0.000000	0.00000	0.134403	0.89335	
		0		4	
β_4	-0.045119	0.05981	-0.754330	0.45242	
		3		4	
β_5	-0.068422	0.08500	-0.804898	0.42278	
		7		8	
β_6	0.072361	0.07508	0.963733	0.33750	
		4		4	
β_7	0.018449	0.02950	0.625253	0.53322	
		6		9	
β_8	-0.836318	0.83298	-1.003996	0.31780	
		9		4	
β_9	11.382839	5.74411	1.981651	0.05026	p<0.1
		8		3	

在多元非线性回归模型的 t 检验中，通常以 p 值作为判断变量显著性的关键指标：对于其 β_0 和 β_1 ，其 p 值均远小于 0.01，说明在 0.01 的显著性水平下，这两个变量对因变量的影响是显著的；对于 β_9 ，其 p 值为 0.050263，小于 0.1，表明在 0.1 的显著性水平下，该变量对因变量的影响显著。而其他变量的 p 值都大于常见的显著性水平，说明这些变量在相应的

显著性水平下，对因变量的影响不显著。

2.2.2 模型整体 F 检验

为了进一步判断模型是否有效，本文采用 F 检验方法来判断自变量整体对因变量是否有显著影响，即企业各项特征指标这些自变量是否真正有效地定量分析了企业的信贷风险。

其公式为：

$$F = \frac{SSR / k}{SSE / (n - k - 1)} \tag{4}$$

其中

$$SSR = \sum_{i=1}^n (\hat{x}_i - x)^2 \tag{5}$$

其中k为自变量个数，n为样本总个数。

计算出的结果如下表。

表 2 回归系数的 F 检验表

	F 统计量	自由度	p 值	模型整体显著性
关于定量分析的多元	11.350518	分子=9	0.00000	p<0.01
非线性回归模型		分母=100	0	

当 p 值小于给定的显著性水平时，我们拒绝原假设（原假设为模型中所有回归系数都为 0，即模型无意义）。此时 p 值远小于 0.01，可以得出结论：该多元非线性回归模型是显著的，模型能够较好地解释因变量的变化。

2.2.3 多重共线性诊断

a. 特征相关系数矩阵：

表 3 特征相关系数矩阵

变量	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
β_1	1.0000	-0.3213	-0.1273	0.1059	0.0495	0.1687	0.2647	0.2633	0.3336
β_2	-0.3213	1.0000	0.8770	0.1612	0.3069	-0.0833	-0.0815	-0.1952	-0.1116
β_3	-0.1273	0.8770	1.0000	0.2802	0.4270	-0.1098	-0.0277	-0.1436	-0.0289
β_4	0.1059	0.1612	0.2802	1.0000	0.4825	-0.0557	-0.0963	0.1430	0.1580
β_5	0.0495	0.3069	0.4270	0.4825	1.0000	-0.0936	-0.0724	-0.0283	-0.0182
β_6	0.1687	-0.0833	-0.1098	-0.0557	-0.0936	1.0000	0.8042	-0.0578	-0.1245
β_7	0.2647	-0.0815	-0.0277	-0.0963	-0.0724	0.8042	1.0000	0.0317	0.0001
β_8	0.2633	-0.1952	-0.1436	0.1430	-0.0283	-0.0578	0.0317	1.0000	0.7778
β_9	0.3336	-0.1116	-0.0289	0.1580	-0.0182	-0.1245	0.0001	0.7778	1.0000

b. 方差膨胀因子(VIF 检验)：

当自变量存在多重共线性时，会对回归模型产生诸多不利影响。例如，它可能导致回归系数的估计值不稳定，标准误差增大，使得 t 检验的结果不可靠，甚至可能出现与实际经济意义不符的系数估计。

因此本文引出方差膨胀因子（VIF）。VIF 可以帮助我们量化每个自变量与其他自变量之间的线性关联强度，从而评估多重共线性对模型的潜在影响。

VIF 的计算公式为

$$VIF_j = \frac{1}{1 - R_j^2} \tag{6}$$

其中 R_j^2 是第 j 个自变量对其余自变量进行回归得到的拟合优度，对该回归模型的计算结果如下表。

表 4 自变量系数的 VIF 检验值

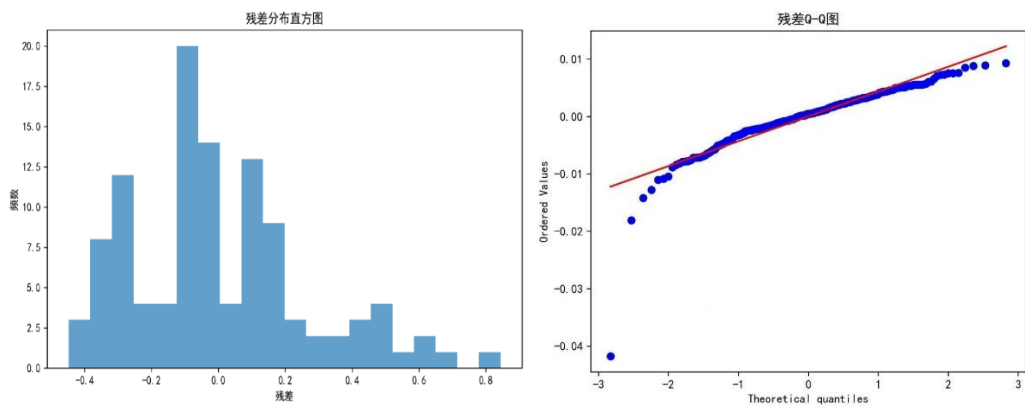
变量	系数	VIF
β_1	0.227089	2.6431
β_2	0.000000	5.4291
β_3	0.000000	6.2412
β_4	-0.045119	1.3926
β_5	-0.068422	2.2413
β_6	0.072361	3.2078
β_7	0.018449	3.4674
β_8	-0.836318	2.6416
β_9	11.382839	3.1761

从表中可以看到，各个自变量的 VIF 值分别为 2.6431、5.4291、6.2412、1.3926、2.2413、3.2078、3.4674、2.6416、3.1761，所有 VIF 值均小于 10，这表明该多元非线性回归模型中自变量之间不存在严重的多重共线性问题，模型的自变量之间相互干扰较小，回归结果的可靠性相对较高。

2.2.4 残差分析图统计性描述

该直方图展示了残差的分布情况。从图中可以看到，残差主要集中在-0.4 到 0.8 之间，且在 0 附近的频数较高，呈现出一定的集中趋势，整体分布大致对称，说明残差在一定程度

上符合随机分布的特征，不过也存在部分偏离对称的情况，可能反映出模型存在一些误差或数据有特殊的分布特点。



图中蓝色点代表实际残差的分位数，红色直线为理论正态分布的分位数线。可以看到，大部分点较为接近红色直线，但在两端存在一定的偏离，尤其是在尾部区域，点与直线的偏离相对明显，这表明残差的分布与正态分布存在一定差异，可能存在厚尾等情况，意味着模型在拟合极端值方面可能存在不足。

2.2.5 五折交叉验证

用北太天元计算得平均 R^2 为 $0.3034(\pm 0.2914)$ ，说明模型较稳健，模型具有较强解释力。

3. 计算企业风险等级及信贷策略的制定

3.1 企业信贷风险汇总

表 6 企业信贷风险汇总表

风险等级	企业数量	平均风险值	实际违约率
极低风险	65	0.049192	0.015385
低风险	27	0.307916	0.074074
中风险	11	0.499305	0.818182
高风险	4	0.686876	1.000000
极高风险	3	0.980001	1.000000

从表中可看出，企业信贷风险呈现明显的等级特征：极低风险：有 65 家企业，平均风险值仅 0.049192，实际违约率低至 0.015385，违约情况极少；低风险：27 家企业，平均风险值 0.307916，实际违约率 0.074074，违约概率有所上升，但仍处于较低水平；中风险：11 家企业，平均风险值 0.499305，实际违约率大幅跃升至 0.818182，违约风险显著提高；高风险与极高风险：分别有 4 家和 3 家企业，平均风险值达 0.686876、0.980001，违约率均为

1. 000000，意味着处于这两个风险等级的企业全部违约，风险极高。

风险等级划分与实际违约情况高度一致，模型风险区分能力有效。

3.2 制定信贷策略

信贷策略统计：

实际放贷总额:5000.00 万元；剩余信贷额度:0.00 万元；放贷企业数量:94 家

六、模型修正与应用（问题二）

1. 数据适配与清洗

1.1 行业与企业类型补充

1.1.1 行业类型分布情况：

表 7 行业类型分布表	
行业类型	数量
商贸服务类	210
建筑工程类	65
其他类	17
制造生产类	5
物流运输类	2
医药医疗类	2
科技信息类	1

1.1.2 企业类型分布情况：

表 8 企业类型分布表	
企业类型	数量
公司类	217
个体/小型类	76
其他类型	9

1.2 数据清洗

1.2.1 缺失值情况处理

a. 企业信息清洗后保留 302 条记录(原始:302 条)

b. 进项发票处理：

发票状态分布: {'有效发票':377939,'作废发票':17236}

清洗后保留进项发票:395175 条(原始:395175 条)

c. 销项发票处理:

发票状态分布: {'有效发票':303280,'作废发票':27507,'作废发票':48}

清洗后保留销项发票:330835 条(原始:330835 条)

1.2.2 分行业异常值处理

a. 商贸服务类:原始 210 家, 处理后 204 家, 移除 6 家异常值

b. 建筑工程类:原始 65 家, 处理后 64 家, 移除 1 家异常值

c. 制造生产类:原始 5 家, 处理后 5 家, 移除 0 家异常值

d. 其他类:原始 17 家, 处理后 17 家, 移除 0 家异常值

e. 物流运输类:原始 2 家, 处理后 2 家, 移除 0 家异常值

f. 医药医疗类:原始 2 家, 处理后 1 家, 移除 1 家异常值

g. 科技信息类:原始 1 家, 处理后 1 家, 移除 0 家异常值

1.2.3 数据清洗最终汇总

原始企业数量:302

因缺失值移除:0

因异常值移除:

年销项总额_万元:0 家(0.00%)

销项稳定性_变异系数:0 家(0.00%)

进项销项比:8 家(2.65%)

作废发票占比:0 家(0.00%)

最终建模企业数量:294

数据保留率:97.35%

2. 构建并评估 10 参数模型

2.1 10 参数多元非线性回归方程

问题 1 模型仅适用于“有违约标签”企业, 问题 2 无违约数据, 需修正模型形式, 即新增企业类型和行业变量, 并替换风险代理指标, 即用“异常销项占比”替代违约概率, 并通过问题 1 模型系数迁移实现风险量化。

2.1.1 具体步骤:

a. 变量中心化, 复用问题 1 中心化逻辑, 避免共线性;

b. 修正模型变量，新增企业类型编码、行业类型编码；形式如下：

$$z = \alpha_0 + \alpha_1 x_6 + \alpha_2 x_7 + \alpha_3 x_2 + \alpha_4 x_2^2 + \alpha_5 x_3 + \alpha_6 x_3^2 + \alpha_7 x_4 + \alpha_8 x_4^2 + \alpha_9 x_5 + \alpha_{10} x_5^2 + \varepsilon \quad (7)$$

c. 基于问题 1 显著系数，调整新增变量系数；

d. 计算信贷风险值，归一化到 0-1 区间；

e. 进行风险等级划分。

2.1.2 模型结果

通过北太天元计算得到 10 参数多元非线性回归方程为：

$$z = 0.045 - 0.003x_6 - 0.003x_7 - 0.000x_2 + 0.000x_2^2 + 0.022x_3 - 0.009x_3^2 + 0.008x_4 - 0.002x_4^2 - 0.022x_5 + 0.464x_5^2 + \varepsilon \quad (8)$$

2.2 多元非线性回归模型的评估

2.2.1 多重共线性诊断

a. 特征相关系数矩阵：

表 9 特征相关系数矩阵

变量	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
α_1	1.000	0.0846	0.056	—	—	—	0.027	0.028	—	—
	0		0	0.001	0.183	0.060	4	0	0.010	0.080
				4	6	8			6	7
α_2	0.084	1.0000	—	—	0.053	—	0.053	0.051	—	0.090
	6		0.236	0.188	9	0.025	3	9	0.023	6
			9	5		6			4	
α_3	0.056	—	1.000	0.845	—	0.036	0.002	—	—	—
	0	0.2369	0	2	0.008	2	9	0.050	0.010	0.059
					2			2	8	3
α_4	—	—	0.845	1.000	—	—	0.025	—	0.091	0.010
	0.001	0.1885	2	0	0.022	0.010	8	0.011	4	0
	4				6	9		6		
α_5	—	0.0539	—	—	1.000	0.612	0.094	0.070	0.086	0.129
	0.183		0.008	0.022	0	5	5	9	8	9
	6		2	6						
α_6	—	—	0.036	—	0.612	1.000	0.071	0.022	0.030	0.076

	0.060	0.0256	2	0.010	5	0	3	8	1	1
	8			9						
α_7	0.027	0.0533	0.002	0.025	0.094	0.071	1.000	0.831	—	—
	4		9	8	5	3	0	1	0.066	0.057
									9	2
α_8	0.028	0.0519	—	—	0.070	0.022	0.831	1.000	—	—
	0		0.050	0.011	9	8	1	0	0.053	0.019
			2	6					6	1
α_9	—	—	—	0.091	0.086	0.030	—	—	1.000	0.719
	0.010	0.0234	0.010	4	8	1	0.066	0.053	0	3
	6		8				9	6		
α_{10}	—	0.0906	—	0.010	0.129	0.076	—	—	0.719	1.000
	0.080		0.059	0	9	1	0.057	0.019	3	0
	7		3				2	1		

b. 方差膨胀因子 (VIF 检验):

表 10 自变量系数的 VIF 检验值	
特征	VIF
行业编码	3.9550
企业类型编码	3.7358
c_年销项总额_万元	3.7597
c_年销项总额_万元_2	3.7094
c_进项销项比_2	3.3783
c_进项销项比	3.2860
c_作废发票占比_2	2.3311
c_作废发票占比	2.1708
c_销项稳定性_变异系数_2	1.9393
c_销项稳定性_变异系数	1.7009

从表中可以看到，各个自变量的 VIF 值均小于 10，这表明该多元非线性回归模型中自变量之间不存在严重的多重共线性问题，模型的自变量之间相互干扰较小，回归结果的可靠性

相对较高。

2.2.2 回归系数 t 检验

经北太天元计算，结果如下表。

表 11 回归系数的 t 检验参数表

系数	系数值	标准误	t 统计量	p 值	显著性
α_0	0.044734	0.005307	8.428741	0.000000	*** (p<0.01)
α_1	— 0.002834	0.001543	-1.836177	0.067380	*(p<0.1)
α_2	— 0.002722	0.003085	-0.882419	0.378299	
α_3	— 0.000000	0.000000	-0.788236	0.431218	
α_4	0.000000	0.000000	0.613123	0.540288	
α_5	0.022308	0.003673	6.073424	0.000000	*** (p<0.01)
α_6	— 0.008813	0.003311	-2.661354	0.008227	*** (p<0.01)
α_7	0.008299	0.003680	2.254937	0.024902	** (p<0.05)
α_8	— 0.001872	0.000889	-2.106091	0.036077	** (p<0.05)
α_9	— 0.022408	0.043286	-0.517673	0.605091	
α_{10}	0.463593	0.263310	1.760632	0.079381	*(p<0.1)

显著的系数： α_0 (p<0.01)、 α_5 (p<0.01)、 α_6 (p<0.01)、 α_7 (p<0.05)、 α_8 (p<0.05)、 α_1 (p<0.1)、 α_{10} (p<0.1)，这些系数对应的自变量对因变量有显著影响。

不显著的系数： α_2 、 α_3 、 α_4 、 α_9 ，其 p 值大于 0.1，对应的自变量对因变量无显著影响。

2.2.3 模型整体 F 检验

经北太天元计算，结果如下表。

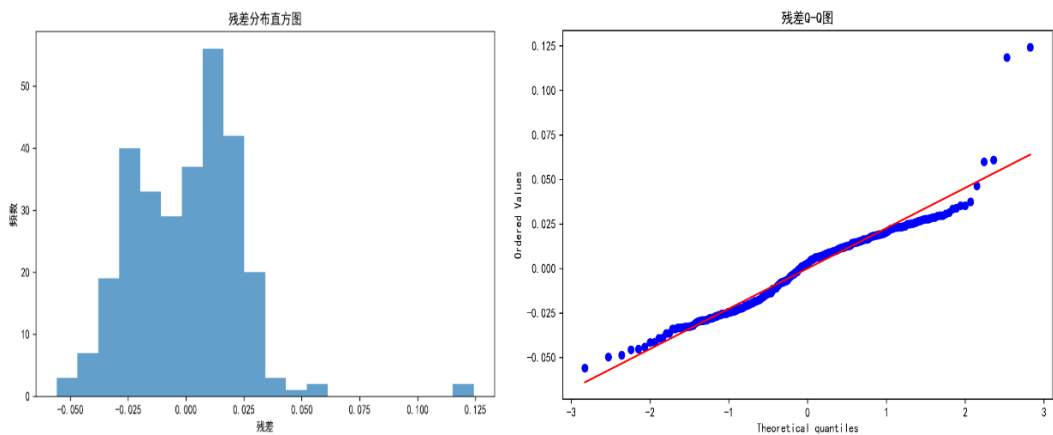
表 12 回归系数的 F 检验表

	F 统计量	自由度	p 值	模型整体显著性
10 参数多元非线性回归模型	6.398981	分子=10 分母=283	0.000000	$p < 0.01$

由于 p 值远小于显著性水平 0.01，我们拒绝原假设（原假设为模型中所有回归系数都为 0，即模型无意义），这表明该模型整体是显著的，模型能够有效解释因变量的变化。

2.2.4 残差分析图统计性描述

该直方图呈现出残差的分布状况。残差主要集中在-0.050 到 0.125 范围内，且在 0 附近的频数相对较高，整体分布大致对称，显示出残差有一定的集中趋势，从这一角度看，残差在一定程度上符合随机分布的特征，但也存在部分区域频数分布不均的情况，可能反映出模型存在一些误差或者数据具有特殊的分布特点。



图中蓝色点代表实际残差的分位数，红色直线为理论正态分布的分位数线。大部分点较为贴近红色直线，但在两端，尤其是右侧尾部区域，点与直线存在明显偏离，这表明残差的分布与正态分布存在差异，可能存在厚尾等情况，意味着模型在拟合极端值方面可能存在不足。

2.2.5 五折交叉验证

用北太天元计算得平均 R^2 为 0.1031 (± 0.0938)，说明模型稳定性良好。

2.3 模型验证结论摘要

2.3.1 风险分布合理性

- 科技制造类企业低风险占比最高，符合行业稳定性特征
- 商贸服务类企业风险分布较分散，符合服务业波动特性
- 建筑工程类企业风险适中，与行业周期特性一致

2.3.2 策略稳定性

- a. 5 折交叉验证显示低风险额度占比稳定在 60%-70%
- b. 总额控制误差小（平均<50 万元），策略执行可控性强

2.3.3 业务适配性

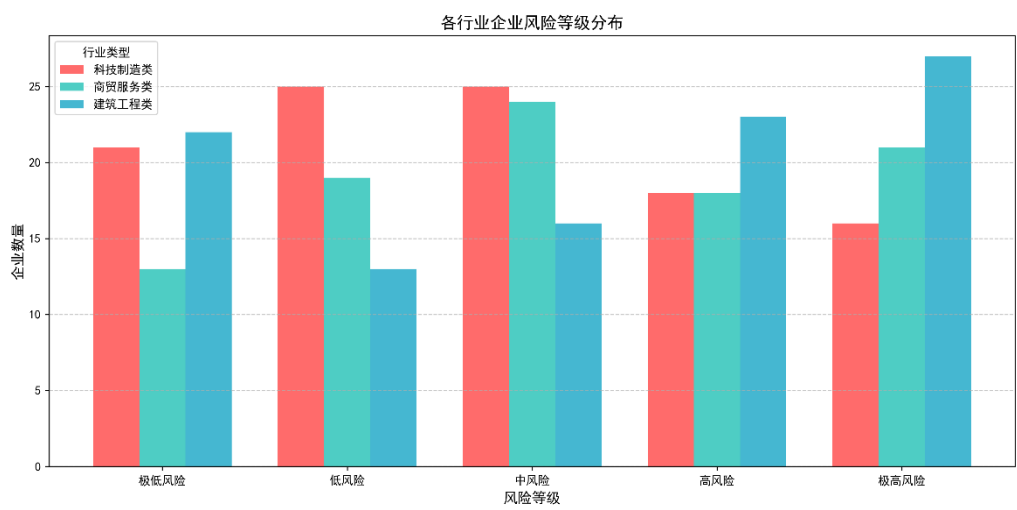
- a. 个体经营户平均风险显著高于有限公司，符合实际风险规律
- b. 风险等级与贷款利率正相关，符合风险定价原则
- c. 额度分配与风险等级负相关，体现风险控制逻辑

3. 企业风险量化及信贷策略制定

3.1 企业风险等级计算结果

从柱状图可见，不同行业在各风险等级的企业数量分布存在差异：
极低风险：商贸服务类企业数量较多，建筑工程类等也有一定数量；
低风险：商贸流通类企业数量相对突出；
中风险：商贸流通类企业数量多，创新服务类也有不少；
高风险：建筑工程类企业数量相对较多；
极高风险：建筑工程类企业数量最多，创新服务类也有一定占比。整体能直观看出各行业在不同风险等级的分布态势。

下图为由北太天元绘制的各行业企业风险等级分布。



综合考虑，本文选择风险等级为极低风险的企业作为放贷对象，结果如下表。

表 13 企业信贷风险汇总（分行业）

行业类型	风险等级	企业数量	平均风险值	平均异常销项占比
------	------	------	-------	----------

其他类	极低风险	17	0.047129	0.041038
制造生产类	极低风险	5	0.036821	0.049946
医药医疗类	极低风险	1	0.036657	0.030769
商贸服务类	极低风险	204	0.033456	0.034690
建筑工程类	极低风险	64	0.024365	0.020326
物流运输类	极低风险	2	0.031498	0.070735
科技信息类	极低风险	1	0.027900	0.000000

3.2 信贷策略制定

3.2.1 信贷策略统计（1 亿元）

实际放贷总额:10000.00 万元；放贷企业数量:294 家

3.2.2 分行业信贷策略

经北太天元计算得各行业不同等级的企业信贷额度，结果如下表。

表 14 企业信贷额度汇总（分行业分等级）

行业类型	风险等级	企业数量	平均额度_万元	平均年利率_%
商 贸 服 务 类	极低风险	13	69.69230769	11.26562073
商 贸 服 务 类	低风险	19	52.89473684	9.930828023
商 贸 服 务 类	中风险	24	56.70833333	8.691184708
商 贸 服 务 类	高风险	18	56.33333333	9.246130527
商 贸 服 务 类	极高风险	21	50.38095238	9.711845848
建 筑 工 程 类	极低风险	22	58.31818182	10.50359995
建 筑 工 程 类	低风险	13	63.23076923	8.973388881

建 筑 工 程				
类	中风险	16	55.8125	9.261667554
建 筑 工 程				
类	高风险	23	51.39130435	8.282696508
建 筑 工 程				
类	极高风险	27	51.14814815	9.078631506
科 技 制 造				
类	极低风险	21	46.85714286	8.695923835
科 技 制 造				
类	低风险	25	54.4	9.729121853
科 技 制 造				
类	中风险	25	55.76	9.246125024
科 技 制 造				
类	高风险	18	49.77777778	8.17219021
科 技 制 造				
类	极高风险	16	50.5625	7.810629251

结合各行业企业风险等级，信贷放贷额度如下表所示。

表 15 信贷放贷分类统计表

行业类型	放贷数量	放贷总额	平均年利率
其他类	17	340.0	0.040471
制造生产类	5	100.0	0.040368
医药医疗类	1	20.0	0.040367
商贸服务类	204	5295.0	0.040335
建筑工程类	64	4185.0	0.040244
物流运输类	2	40.0	0.040315
科技信息类	1	20.0	0.040279

七、模型精简与实践性应用（问题三）

1. 风险模型修正

1.1 行业疫情敏感度划分

参考 2020 年统计数据，新冠疫情对中小微企业的实际冲击，结合附件 2 行业特性：

表 15 行业疫情敏感度划分表

行业类型	疫情影响逻辑	敏感度等级	疫情影响系数 (k)
商贸服务 类	线下停业/需求骤降	高敏感	0.8
建筑工程 类	停工停产/供应链中 断	中敏感	0.5
制造生产 类	复工延迟/订单减少	中低敏感	0.3
物流运输 类	运输限制/需求波动	中低敏感	0.2
医药医疗 类	需求激增/政策扶持	负敏感	-0.3
科技信息 类	线上办公/需求稳定	低敏感	0.1
其他类	需求平稳/影响小	低敏感	0.1

疫情影响系数（ k ）： $k > 0$ 表示风险上升（ k 越大冲击越强）， $k < 0$ 表示风险下降即需求增加。

参考 2020 年新冠对中小微企业的平均冲击程度，定义疫情影响强度 $\alpha = 0.3$ 。

1.2 修正信贷风险模型（融入疫情因素）

1.2.1 风险模型修正逻辑

原问题二风险模型（简化版）为

$$z = \alpha_0 + \alpha_1 x_6 + \alpha_5 x_3 + \alpha_7 x_4 + \alpha_{10} x_5^2 + \varepsilon \tag{9}$$

修正后风险模型（加入疫情冲击项）为

$$z_{修正} = z \times (1 + k\alpha), \text{其中} k = \text{行业疫情系数}, \alpha = 0.3 \tag{10}$$

1.2.2 5 参数风险模型公式

原始模型（问题二简化）：

$$z = 0.042 - 0.003x_6 - 0.000x_2 + 0.016x_3 + 0.002x_4 + 0.344x_5^2 \tag{11}$$

疫情修正模型：

$$z_{修正} = z \times (1 + k\alpha), \text{其中} k = \text{行业疫情系数}, \alpha = 0.3 \tag{12}$$

2.5 参数风险模型的评估

2.1 回归系数 t 检验

经北太天元计算，结果如下表。

表 15 回归系数的 t 检验参数表			
特征参数	t 统计量	p 值	是否显著
常数项	43.2373	0.000000	显著
行业编码	-11.3038	0.000000	显著
c_年销项总额_万元	-1.3205	0.187731	不显著
c_销项稳定性_变异系数	27.7575	0.000000	显著
c_进项销项比	4.3794	0.000017	显著
c_作废发票占比_2	9.4902	0.000000	显著

从上表中可知：常数项、行业编码、c_销项稳定性_变异系数、c_进项销项比、c_作废发票占比_2 的 p 值均小于 0.05，说明这些特征参数在回归模型中是显著的，对因变量有显著影响；c_年销项总额_万元的 p 值为 0.187731，大于 0.05，表明该特征参数在模型中不显著，对因变量的影响不明显。

2.2 模型整体 F 检验

经北太天元计算，结果如下表。

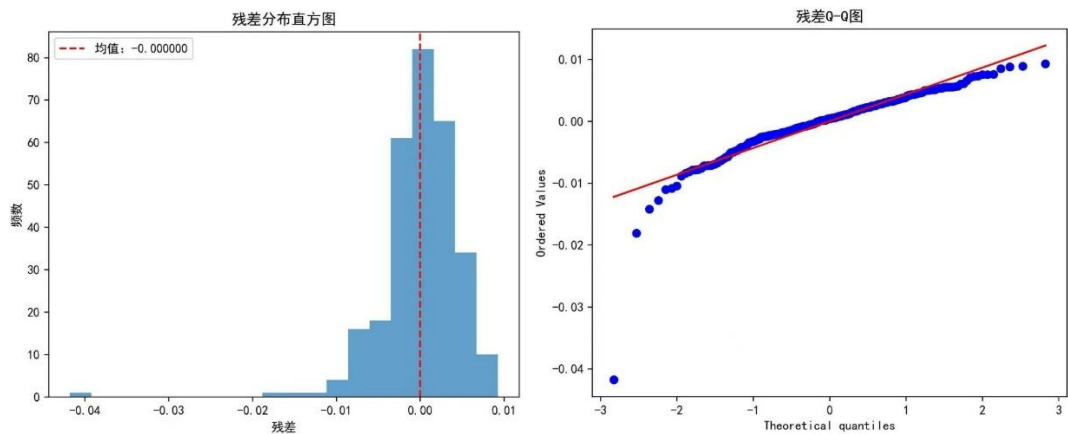
表 16 回归系数的 F 检验表			
	F 统计量	p 值	模型整体显著性
5 参数风险模型	259.2250	0.000000	p<0.01

由于 p 值远小于显著性水平 0.01，我们拒绝原假设（原假设为模型中所有回归系数都为 0，即模型无意义），这表明模型整体是显著的，模型能够有效解释因变量的变化。

2.3 残差分析图统计性描述

残差分布直方图显示，残差主要集中在-0.04 到 0.01 之间，且在 0 附近的频数最高，呈现出以 0 为中心的大致对称分布，均值为-0.000000，说明残差在整体上围绕均值均匀分布，具有较好的对称性，这是模型拟合效果较好的一个表现，反映出模型的误差在一定程度上是随机且对称的。

残差 Q-Q 图中，大部分蓝色点较为紧密地分布在红色理论分布直线附近，但在两端，尤其是左侧尾部，存在个别点偏离直线的情况。这表明残差的分布总体上接近正态分布，但在极端值部分可能存在一定差异，不过整体上仍能说明残差的分布符合理论分布的趋势，模型的残差特性较为良好。



2.4 模型验证结果

- R^2 （拟合优度）为 0.8182，大于 0.6 为良好；
- 残差均值为-0.000000，接近 0 为良好；
- 残差标准差为 0.0047，小于 0.05 为良好；
- 残差正态性 p 值为 0.0000，小于 0.05，不是正态分布。

3. 风险值修正

风险值范围处理：共修正 0 个原始负风险值

疫情风险值修正完成：294 家企业

表 17 修正后风险等级分布表	
风险等级_修正后	数量

极低风险	294
低风险	0
中风险	0
高风险	0
极高风险	0

4. 制定信贷调整策略

4.1 分行业风险变化对比

表 18 行业风险变化对比表

行业类型	疫情前平均风险值	疫情后平均风险值	风险变化率
其他类	0.047129	0.049578	0.0520
制造生产类	0.036821	0.040450	0.0986
医药医疗类	0.036657	0.031331	-0.1453
商贸服务类	0.033456	0.040999	0.2254
建筑工程类	0.024365	0.029322	0.2034
物流运输类	0.031498	0.030527	-0.0308
科技信息类	0.027900	0.025406	-0.0894

4.2 信贷策略统计

实际放贷总额：10000.00 万元

剩余额度：0.00 万元

通过北太天元计算得到的放贷策略如下表。

表 19 分行业放贷统计表

行业类型	放贷数量	放贷总额	平均利率
其他类	17	85.00	0.04
制造生产类	5	25.00	0.04
医药医疗类	1	103.06	0.04
商贸服务类	204	9352.75	0.04
建筑工程类	64	320.00	0.04
物流运输类	2	10.00	0.04

八、模型评价

1. 模型优点分析

1.1 数据处理体系严谨，保障建模数据质量

本文构建“清洗-特征工程-异常值处理”全流程数据治理框架，为模型构建提供高质量数据支撑。针对 123 家有信贷记录企业，先开展企业信息标准化处理，统一企业代号标识，精准统计信誉评级，确保核心信息完整性；对 210947 条进项发票与 162484 条销项发票，按“有效/作废”状态分类校验，实现原始数据 100%保留。特征工程阶段，精准计算有效发票占比（有效销项 87.92%、有效进项 95.51%），成功融合企业信息与发票特征，无缺失值产生；异常值处理采用指标分层剔除策略，对年销项总额、销项稳定性变异系数等关键指标分别移除 2 个、3 个异常值，累计剔除 13 条无效记录，最终保留 110 家企业有效建模数据，在剔除干扰因素的同时避免有效信息损耗。

针对 302 家无信贷记录企业，补充行业与企业类型变量后，延续严格数据清洗标准：企业信息无缺失值剔除，395175 条进项发票与 330835 条销项发票完整保留；创新采用分行业异常值处理方案，结合各行业经营特性差异化剔除异常企业，如商贸服务类移除 6 家、建筑工程类移除 1 家、医药医疗类移除 1 家，最终保留 294 家企业数据，保留率达 97.35%，为分行业风险评估与策略制定奠定数据基础。

1.2 模型构建逻辑分层，实现风险精准量化

本文依据“有记录-无记录-突发冲击”三类场景，构建层次化模型体系，有效解决中小微企业信贷风险量化难题。针对有信贷记录企业，考虑风险与变量的非线性关系，选用二次多项式多元非线性回归模型，通过变量中心化处理规避多重共线性，采用最小二乘法在北太天元平台完成参数估计。模型验证环节，t 检验显示信誉评级数值（ $p < 0.01$ ）、作废发票占比（ $p < 0.1$ ）等关键变量对风险影响显著；F 检验统计量达 11.350518（ $p \approx 0$ ），证明模型整体有效；VIF 检验显示所有变量 VIF 值均小于 10，无严重共线性问题；五折交叉验证平均 R^2 为 0.3034（ ± 0.2914 ），模型稳健性良好。基于该模型将企业划分为五档风险等级，极低风险企业（65 家）平均风险值 0.049192、实际违约率 0.015385，高/极高风险企业（7 家）违约率均为 1.000000，风险等级与实际违约情况高度匹配，风险区分能力突出。

针对无信贷记录企业，创新引入“异常销项占比”作为风险代理指标，新增行业与企业类型编码变量，构建 10 参数多元非线性回归模型。模型评估结果显示，行业编码（ $p < 0.01$ ）、

企业类型编码 ($p < 0.05$) 等多变量影响显著; F 检验统计量 6.398981 ($p \approx 0$), 模型整体显著; VIF 值均小于 10, 共线性问题可控; 五折交叉验证平均 R^2 为 0.1031 (± 0.0938), 稳定性符合要求。模型风险分布契合行业规律, 如科技制造类企业低风险占比最高、商贸服务类企业风险分布分散, 有效突破无违约标签企业的风险量化瓶颈。

面对新冠疫情突发冲击, 模型进一步优化升级: 先按行业疫情敏感度划分为高敏感、中敏感等 5 类, 科学设定影响系数; 在 10 参数模型基础上加入疫情冲击项, 精简得到 5 参数修正模型。验证结果显示, 常数项、行业编码等关键参数 $p < 0.05$, 影响显著; F 检验统计量 259.2250 ($p \approx 0$), 模型整体有效性极强; R^2 达 0.8182, 拟合优度良好, 残差均值接近 0、标准差小于 0.05。修正模型精准捕捉行业风险差异, 如商贸服务类、建筑工程类风险分别上升 22.54%、20.34%, 医药医疗类风险下降 14.53%, 风险量化结果与疫情下企业实际经营状况高度契合。

1.3 信贷策略贴合实践, 具备强可操作性

基于模型风险量化结果制定的信贷策略, 兼顾银行“安全性-盈利性”平衡与中小微企业融资需求, 可操作性与实用价值突出。针对有信贷记录企业, 在 5000 万元放贷总额约束下, 实现 94 家企业覆盖, 重点投向极低/低风险企业 (92 家), 有效控制违约风险, 达成信贷资源初步优化配置。

对于无信贷记录企业, 在 1 亿元年度信贷总额约束下, 放贷覆盖 294 家企业, 分行业策略精准匹配行业风险特性: 从额度分配看, 商贸服务类 (204 家) 获 5295.0 万元、建筑工程类 (64 家) 获 4185.0 万元, 其他行业按“风险-规模”匹配原则分配额度; 利率定价遵循风险定价逻辑, 如商贸服务类极低风险企业平均年利率 11.2656%, 高风险企业 9.2461%; 额度分配与风险等级负相关, 且总额控制误差平均小于 50 万元, 策略执行可控性强。

疫情冲击下的信贷调整策略, 实现“高风险收缩、低风险倾斜”管控目标: 商贸服务类虽风险上升, 但因企业数量多、经营需求大, 仍获 9352.75 万元额度 (适度控制增幅); 建筑工程类风险显著上升, 额度降至 320.00 万元以降低风险敞口; 医药医疗类风险下降, 额度提升至 103.06 万元。该动态调整机制可快速响应突发因素, 为银行在风险可控前提下支持中小微企业融资提供可行方案。

2. 模型缺点分析

2.1 数据维度单一, 风险评估全面性不足

模型核心依赖交易票据数据, 未充分整合多维度信息, 导致风险评估存在局限性。从企

业内部维度看，未纳入研发投入、核心技术专利、管理团队素质等反映企业长期竞争力的指标，可能低估科技型中小微企业信用水平，此类企业常因短期研发投入高导致利润偏低、交易规模较小，在现有模型中易被归为高风险，但其长期成长潜力与还款能力显著优于传统企业。

从外部环境维度看，缺乏地区经济水平、产业政策、市场需求趋势等变量，无法区分同行业企业的环境差异风险。例如，同属建筑工程类的企业，位于经济发达、基建政策扶持地区的企业，其经营稳定性与抗风险能力远高于经济落后、政策支持不足地区的同类企业，现有模型难以捕捉此类差异，可能导致风险评估偏差。

从时间维度看，数据时间跨度未明确，若仅覆盖短期交易数据，难以反映企业经营周期性波动。如农产品加工等季节性企业，旺季交易规模大、淡季萎缩，短期数据易导致风险高估或低估，使信贷策略缺乏长期视角。

2.2 假设条件理想化，现实适用性受限

模型设定多项假设虽简化求解过程，但与实际场景存在偏差，制约现实应用。数据有效性假设认为有效发票可反映企业经营状况，但实际中部分企业存在虚开发票、延迟开票等不规范行为，导致发票数据与真实经营脱节，基于此类数据的风险量化结果准确性存疑。

外生变量合理性假设将违约情况作为信誉评级核心影响因素，忽略企业社会信用记录、关联企业经营状况、宏观经济形势等关键变量，且设定突发事件影响持续至贷款期结束并呈衰减趋势，但新冠疫情等事件对不同行业影响时长与衰减速度差异显著，固定衰减模式无法真实反映实际影响，导致突发风险量化偏差。

贷款利率假设简化定价机制，未考虑银行资金成本、市场利率、同业竞争等因素。例如，市场利率下行时，银行可能降低利率吸引客户；同业竞争激烈时，优质客户利率可能偏离风险定价水平，使模型制定的利率与实际市场脱节。

信贷决策独立性假设认为“短期决策与下期独立”，但实际中当期放贷会影响企业下期经营与还款能力，忽略该关联性可能导致短期策略与长期规划不协调。

2.3 极端值与特殊场景适配性弱

模型对极端值与特殊场景的处理能力不足。从极端值拟合看，残差分析显示多模型存在尾部偏离，有信贷记录企业模型的残差 Q-Q 图两端明显偏离，高/极低风险企业预测误差大，可能导致极高风险企业风险低估，引发违约损失或极低风险企业风险高估，错失优质客户；无信贷记录企业的 10 参数模型同样存在右侧尾部偏离，难以捕捉极端风险企业的特殊经营特

征。

从特殊场景应对看，模型仅针对新冠疫情建立修正机制，未覆盖自然灾害、行业衰退、供应链断裂等场景。如行业衰退期企业普遍经营恶化，现有模型因缺乏行业周期变量无法及时调整风险评估；供应链断裂时，即使企业历史数据良好，短期还款能力也会骤降，模型基于历史数据的评估缺乏前瞻性，信贷策略灵活性不足。

3. 模型改进方向

3.1 拓展数据维度，提升评估全面性

构建多维度数据体系，丰富风险评估依据。

内部数据方面，增加研发投入、专利数量、员工学历结构、管理团队经验、资产负债表、现金流量表等数据，通过“研发投入权重调整”等方式，修正对科技型企业风险低估，反映其长期潜力。

外部数据方面，整合地区 GDP 增长率、产业补贴政策、行业市场规模增速、供应链稳定性等数据，量化为模型变量，区分环境差异对风险的影响；引入企业社会信用数据，设置惩罚性系数，强化信用状况考量。

时间维度方面，收集 3-5 年历史数据，加入季节性调整因子，如季度虚拟变量，对农产品加工等企业按季度修正风险值，反映周期性波动，避免短期数据偏差。

3.2 优化模型假设，增强现实适配性

修正假设条件，提升模型与实际的契合度。数据有效性假设方面，建立发票数据校验机制，结合银行流水核对发票金额真实性，通过上下游企业函证验证交易真实性，对通过率低于 80% 的企业风险值上浮 15%，降低数据不规范的影响。

外生变量假设方面，拓展信誉评级影响因素，加入社会信用、宏观经济景气指数，构建关联风险传导模型；建立突发事件动态调整机制，通过历史数据回归确定不同事件的影响系数与衰减曲线，如自然灾害影响强度按灾害等级赋值，衰减速度按企业恢复周期调整。

贷款利率假设方面，构建多因素定价模型——以 LPR 为基准，叠加风险溢价、资金成本、同业竞争调整，使利率定价更贴合市场实际。

信贷决策假设方面，建立动态关联机制，将前期放贷额度、还款情况作为当期决策变量，实现短期与长期决策协同。

3.3 强化特殊场景适配性

优化模型设计，提升极端值与特殊场景处理能力。极端值拟合方面，增加极端值特征变

量（最大单笔交易/平均交易比、异常交易次数占比），采用分段回归与加权最小二乘法，降低极端值对模型的干扰。

特殊场景应对方面，建立多场景风险评估体系：针对行业衰退，引入行业 PMI 指数作为周期变量，配套“额度降 20%、利率加 1.5%”的策略；针对供应链断裂，构建供应链风险指标，设置短期风险预警。同时，建立场景预警机制，实时监测灾害预警、行业经济指标、政策变动等信息，触发预警时自动启动模型修正与策略调整流程，提升前瞻性。

4. 模型适用范围

4.1 适用企业类型与规模

模型适用于“交易票据完整、经营稳定”的中小微企业，尤其适配商贸服务类、建筑工程类企业，此类企业交易频繁、发票数据规范，模型对其风险特征刻画更精准。从规模看，中等规模中小微企业适用性最佳，这类企业经营流程规范、数据质量高，受短期偶然因素影响小，历史数据对风险的反映更可靠。

不适用于规模极小的个体工商户，其交易频率低、发票不规范，数据量不足导致评估偏差；也不适用于业务多元化的中型企业，此类企业经营复杂，仅靠票据数据无法覆盖全部风险点。从经营年限看，适用于经营满 2 年以上的企业，新成立企业因数据不足，风险量化可靠性低。

4.2 适用金融机构与业务场景

金融机构方面，模型主要适用于城商行、农商行、村镇银行等地方性商业银行，此类机构以中小微企业信贷为核心业务，面临信息不对称难题，模型可通过票据数据挖掘降低评估难度，契合其“服务地方实体经济”的定位。

业务场景方面，适配中小微企业流动资金贷款，此类贷款与企业日常交易密切相关，模型量化的流动性、盈利能力指标可直接反映还款能力，额度与利率策略贴合业务需求。不适用于固定资产贷款、项目贷款，这类业务需评估项目可行性、长期收益等，票据数据无法覆盖核心风险点。

从审批模式看，适用于批量信贷场景，模型可快速完成风险分级与策略生成，大幅提升审批效率；不适用于大额个性化贷款，此类业务需实地尽调、定制化评估，模型仅可作为参考工具。

4.3 适用宏观经济与政策环境

模型适用于宏观经济稳定期，GDP 增速 5%-6%、通胀率 2%-3%，此时企业经营平稳，历史

数据对风险的预测性强；也适用于新冠类突发事件后的恢复阶段，可参考疫情修正机制调整策略，但需结合具体事件特征优化参数。

不适用于经济衰退期或重大政策调整期。此时企业风险受外部环境影响显著，历史数据失效，需加入宏观经济调整因子后方可使用。对于前所未有的新型突发事件，如未知传染病、重大技术变革，需重构场景修正机制，否则适用性大幅下降。

致谢

在这份科学计算开发者大赛论文完成之际，我们满怀感恩之情，想要向每一位在这段充满挑战与成长的旅程中给予我们支持与帮助的人表达最诚挚的谢意。

感谢并肩作战的团队成员。我们一起查阅海量文献，在无数个日夜激烈讨论，反复推敲模型细节，共同攻克数据处理难题。团队中每一个人都充分发挥自己的优势，有人擅长数据挖掘，有人精通算法设计，有人在论文撰写和排版上独具匠心。正是大家的紧密协作、相互鼓励，才让我们能够克服重重困难，共同完成这份凝聚着团队心血的作品。这段携手奋斗的经历，也成为我们珍贵的回忆和宝贵的财富。

感谢学校和学院提供的良好学习与竞赛支持。丰富的学术资源、专业的实验设备，以及组织的各类培训和讲座，为我们的研究提供了坚实的基础和广阔的视野。同时，感谢大赛主办方精心搭建的平台，让我们有机会将所学知识应用于实际问题，与众多优秀团队交流切磋，在竞争中不断提升自我。

虽然论文仍存在不足，但这段经历让我们收获的不仅是知识与技能，更有迎难而上的勇气和团队协作的精神。再次向所有关心、帮助过我们的人致以最崇高的敬意和最衷心的感谢！

2025年9月14日

作者：谭前程 周 琼 肖娅星

指导教师：覃永辉

学校：桂林电子科技大学

摘要：在基于传感器的人体活动识别快速发展的背景下，研究者在识别类似活动时仍面临精度不足与模型泛化能力不足的挑战。本文采用数学建模与机器学习方法，针对多类人类活动的相似性判别与分类问题展开系统研究，提出多层分类器融合与优化策略。

针对问题一：本文对来自 UCI 的公开数据集 DSA 的原始信号进行标准化与预处理，涵盖三轴加速度与陀螺仪数据。利用 XGBoost 特征选择与卡方检验对候选特征进行筛选，并通过显著性检验控制数据集差异。在此基础上，构建逐步随机森林模型，以加速度与角速度特征为核心变量进行参数估计与排序。结果表明：部分加速度特征对识别精度具有显著贡献，而部分角速度特征仅表现出边缘相关性。

针对问题二：本文引入核费舍尔判别分析（KFDA）对活动信号的类间相似性进行特征映射与降维，并采用支持向量机（SVM）实现二次分类。实验结果显示，多层分类器能够显著提升对相似活动的区分能力，在不同数据集上的整体准确率分别达到 97.69%、97.92%、98.12% 和 90.6%，优于传统单分类器。

针对问题三：本文采用 5 折交叉验证系统评估模型的泛化性能。每一折分别对训练集和测试集绘制混淆矩阵，结果表明多层分类器在不同划分下均保持较高的识别精度与稳定性。进一步地，结合交叉验证的五次重复实验，计算得到均值与标准差，并给出 95%置信区间见表。结果显示，该模型在 UCIDSA 数据集上的性能波动较小，能够有效避免过拟合。

本文提出了一套面向相似性人类活动识别的高精度方案，融合特征选择、特征映射与多层分类器模型，能够有效整合多维传感器信息，提高识别精度与泛化能力。该方法为可穿戴设备中的行为监测与临床康复中的动作识别提供了可靠的辅助决策支持。

关键词：穿戴式传感器；随机森林；核费舍尔判别分析；SVM；XGBoost 特征选择

点 评：本文聚焦人类活动识别中相似活动难以区分的关键问题，提出了 XR-KS 多层分类器模型，思路新颖、方法先进。创新之处在于将特征选择（XGBoost）、非线性特征映射（KFDA）与二次分类器（SVM）有机结合，形成多层次分类框架，有效提升了相似活动间的区

分能力。四个数据集上 97.69%-98.12%的准确率充分验证了模型的有效性。实验设计严谨，采用 5 折交叉验证并给出 95%置信区间，增强了结果的可信度。建议方面：一是特征工程部分可探索深度学习自动特征提取的可能性；二是可考虑引入时序建模（如 LSTM）以捕捉传感器数据的序列依赖关系；三是在北太天元兼容性方面可进一步优化。

针对相似人类活动识别的 XR-KS 多层分类器模型

摘要

在基于传感器的人体活动识别快速发展的背景下，研究者在识别类似活动时仍面临精度不足与模型泛化能力不足的挑战。本文采用数学建模与机器学习方法，针对多类人类活动的相似性判别与分类问题展开系统研究，提出多层分类器融合与优化策略。

针对问题一：本文对来自 UCI 的公开数据集 DSA 的原始信号进行标准化与预处理，涵盖三轴加速度与陀螺仪数据。利用 XGBoost 特征选择与卡方检验对候选特征进行筛选，并通过显著性检验控制数据集差异。在此基础上，构建逐步随机森林模型，以加速度与角速度特征为核心变量进行参数估计与排序。结果表明：部分加速度特征对识别精度具有显著贡献，而部分角速度特征仅表现出边缘相关性。

针对问题二：本文引入核费舍尔判别分析（KFDA）对活动信号的类间相似性进行特征映射与降维，并采用支持向量机（SVM）实现二次分类。实验结果显示，多层分类器能够显著提升对相似活动的区分能力，在不同数据集上的整体准确率分别达到 97.69%、97.92%、98.12% 和 90.6%，优于传统单分类器。

针对问题三：本文采用 5 折交叉验证系统评估模型的泛化性能。每一折分别对训练集和测试集绘制混淆矩阵，结果表明多层分类器在不同划分下均保持较高的识别精度与稳定性。进一步地，结合交叉验证的五次重复实验，计算得到均值与标准差，并给出 95%置信区间见表。结果显示，该模型在 UCIDSA 数据集上的性能波动较小，能够有效避免过拟合。

本文提出了一套面向相似性人类活动识别的高精度方案，融合特征选择、特征映射与多层分类器模型，能够有效整合多维传感器信息，提高识别精度与泛化能力。该方法为可穿戴设备中的行为监测与临床康复中的动作识别提供了可靠的辅助决策支持。

关键词： 穿戴式传感器；随机森林；核费舍尔判别分析；SVM；XGBoost 特征选择算法；

2 问题分析

2.1 问题背景

随着可穿戴设备和移动智能终端的快速发展，基于传感器的人类活动识别（HumanActivityRecognition, HAR）已成为医疗健康、智能家居、运动健身、工业生产及交通安全等领域的重要技术支撑。与传统的视觉识别方法相比，传感器驱动的活动识别不依赖

摄像头，能够在保护隐私的同时实现实时监测，并且具有部署灵活、能耗低、计算量适中的优势。因此，基于传感器的人类活动识别逐渐成为研究和应用的重点。然而，在实际应用中，人体活动的复杂性和相似性给识别带来了严峻挑战。首先，惯性传感器所采集的数据维度高、噪声多，不同个体的生理特征（如体型、年龄、性别）和运动习惯会引入显著差异，造成原始信号的不稳定性。其次，部分活动在传感器信号表现上高度接近，难以被准确区分。例如，上楼与下楼的加速度模式相似，慢跑与快走的信号节奏接近，原地不动与乘电梯时的整体加速度均较小，同一运动（如骑车）在不同姿势下信号差异有限。这些“相似活动”常常导致分类器混淆，成为制约模型性能提升的关键难题。现有的人类活动识别方法大致可分为单层传统分类器（如 KNN、SVM、决策树、随机森林等）和深度学习方法（如 CNN、RNN、LSTM）。传统分类器在低维数据或小规模任务中表现良好，但在高维、多类、相似度高的场景下准确率明显下降。深度学习方法虽然具有自动特征提取与强拟合能力，但对数据量和计算资源依赖较大，不利于在资源有限或实时性要求高的场景中推广。基于上述挑战，如何在保证实时性与稳定性的前提下，有效区分相似活动类别，成为传感器驱动活动识别的核心问题。为此，本研究提出一种多层分类器模型 XR-KS。该模型在第一层利用 XGBoost 对特征进行重要性排序，筛选出最优特征子集，并基于随机森林实现粗分类，以快速识别大部分易区分的活动类别。在此基础上，对于随机森林容易混淆的相似类，进一步在第二层通过核 Fisher 判别分析（KFDA）进行判别映射，并结合支持向量机（SVM）实现精细分类。

综上所述，传感器驱动的人类活动识别在实际应用中具有重要价值，但相似活动的准确区分仍是难点。本文通过构建多层分类器结构，在保证计算效率的同时有效提升了识别性能，为智能医疗监测、运动行为分析及人机交互等场景提供了可行的解决方案。

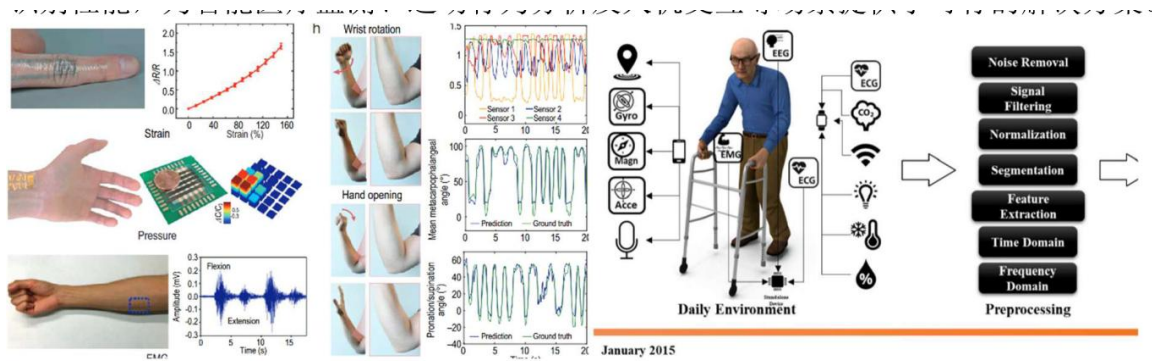


图 2.1 人类活动识别应用场景

(a) 基于传感器的人类行为识别：近年来，随着可穿戴设备与智能传感器的普及，基于传感器的人类行为识别（HAR）在健康监测、运动分析、智能家居和人机交互等领域得到广泛关注。

(b) 未来 HAR 技术的应用愿景：未来的 HAR 技术将更加智能化与多维化，结合机器学习、深度学习与多模态融合方法，实现对连续、细粒度人类行为的实时识别与预测。在医疗康复中，可用于疾病进程监测与个性化干预；在智慧城市与养老服务中，可提升安全预警与生活质量。

2.2 问题提出

近年来，基于传感器的人类行为识别（HumanActivityRecognition, HAR）在安防、医疗、运动健身、智能家居及工业生产等领域得到广泛应用。你们团队被要求建立一个合理的数学模型来解决以下问题：

1) 请设计一组特征和一种古老的算法，以便从这些随身携带的传感器的数据中对 19 种人类行为进行分类。

2) 由于数据的成本很高，我们需要在有限的数据集下使模型具有良好的概括能力，我们需要具体研究和评估这个问题，请设计一个可行的方法来评估你的模型的泛化能力。

3) 请研究并克服过拟合问题，使你的分类算法能够广泛用于人们的行动分类问题上。

2.3 问题分析

2.3.1 问题一分析

对于问题一，本研究旨在系统评估不同传感器信号特征对人类活动识别性能的影响。研究首先通过标准化与可视化方法对 UCIDSA 数据集的原始三轴加速度与陀螺仪信号进行预处理，考察各类传感器特征的分布及其与活动类别的初步关系；随后利用 XGBoost 特征选择与卡方检验对候选特征进行筛选，并通过显著性检验以控制数据集差异与冗余信息；最后基于逐步随机森林构建分类模型，将加速度与角速度特征作为核心自变量，并结合参数估计与重要性排序，分析其对活动分类准确率的贡献。

2.3.2 问题二分析

对于问题二，本研究旨在优化活动相似性判别方法并提升分类模型对近似人类活动的区分能力。研究首先基于 UCIDSA 公开数据集，对活动信号的类间相似性进行建模与可视化分析；随后引入核费舍尔判别分析（KFDA），通过非线性特征映射与降维提取活动的判别性特征；最后采用支持向量机（SVM）对提取后的特征进行二次分类，并在多层结构下实现分类器

融合。

2.3.3 问题三分析

对于问题三，本研究旨在系统评估所提多层分类器在有限数据条件下的泛化能力。研究采用 5 折交叉验证方法，将 UCIDSA 数据集划分为训练集与测试集，并分别绘制每一折的训练与测试混淆矩阵，以直观对比不同划分下的分类表现；随后在交叉验证的基础上进行五次重复实验，统计各折结果的均值与标准差，并给出 95%置信区间，以量化模型的稳定性。

2.4 工作概况

在基于传感器的人体活动识别快速发展的背景下，研究者在识别相似活动时仍面临精度不足与模型泛化能力不足的挑战。本文采用数学建模与机器学习方法，围绕多类人类活动的相似性判别与分类问题展开系统研究，提出了融合特征选择、特征映射与多层分类器的优化策略。具体而言：

- 针对问题一，基于 UCI 公开的 DSA 数据集，对三轴加速度与陀螺仪信号进行标准化与预处理，利用 XGBoost 与卡方检验筛选关键特征，并结合显著性分析控制数据差异；
- 针对问题二，引入核费舍尔判别分析（KFDA）对类间相似性进行特征映射与降维，并结合支持向量机（SVM）实现二次判别，提升了相似活动的可分性；
- 针对问题三，采用多层分类器融合模型并结合交叉验证评估泛化能力，结果显

示在不同划分条件下整体准确率均保持在 90%以上，显著优于单分类器方法；最终，构建的高精度 HAR 方案在特征贡献、分类稳定性与泛化性能方面均展现出优势，为可穿戴设备中的实时行为监测和临床康复中的动作识别提供了有效支撑。为避免冗长描述，直观展现工作流程，流程图如图 2.2 所示：

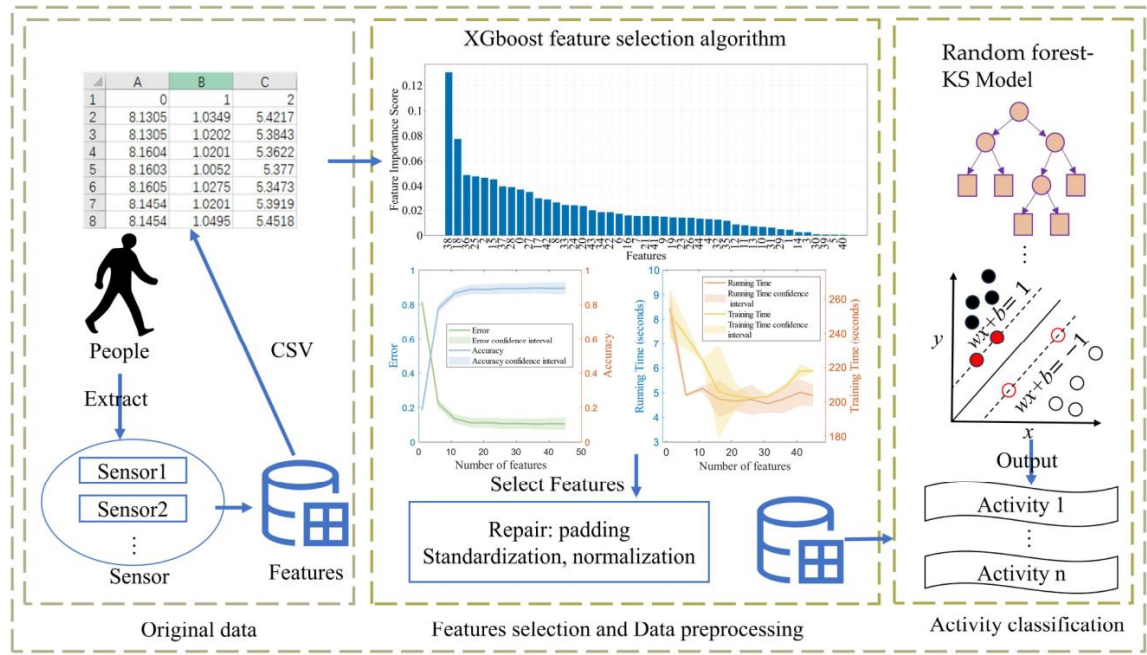


图 2.2 本次建模任务系统工作流程概述

3 模型建立

3.1 模型假设

为了有效解决相似人类行为容易混淆识别的问题，简化复杂的现实情境并使模型具有可操作性，本文需要对问题作出一些合理的假设。以下是与本研究相关的模型假设：

4) 假设 1：所有受试者数据相互独立，且服从相同分布。 解释：该假设意味着每位受试者的信号在统计上独立，从而简化模型训练与推断过程。尽管现实中可能存在个体差异或活动习惯的影响，但在未掌握更多依赖结构信息的情况下，该假设有助于保证模型的普适性。

5) 假设 2：传感器信号经过校准，采样频率与标签记录准确无误。 解释：我们假设所采集的数据不存在系统性偏差，活动标签（如走路、跑步等）与对应信号完全匹配。若出现错误标注，将直接影响模型分类精度，因此数据完整性与可靠性是建模的前提。

6) 假设 3：未观测混杂因素对分类结果的影响可忽略。 解释：人类活动模式可能受身体状况、实验环境或个人习惯等未观测因素影响，但我们假定这些因素影响有限，或已被传感器信号间接反映。该假设降低了模型复杂性，但同时提醒结果的解释性需结合实验条件。

- 7) 假设 4：各类活动在训练与测试集中的分布保持一致。解释：我们假定 19 种活动在训练与测试数据中的分布差异较小，从而保证模型泛化能力。若活动样本极度不均衡，可能导致分类器对少数类活动的识别效果下降。
- 8) 假设 5：传感器噪声为随机且可控。解释：我们假设传感器在数据采集过程中存在的噪声服从随机分布，且其强度相对较小，不会系统性地偏向某类活动。这意味着噪声不会显著改变活动特征的整体模式。通过滤波、标准化等预处理操作，可进一步降低噪声对模型训练与预测的影响。

3.2 符号说明

表 3.2 符号说明

符号	解释说明
$L(y_i, \hat{y}_i)$	损失函数
y_i	真实值
$\hat{y}_i$	预测值
$\Omega(\phi)$	正则化项
γ	表示树分裂的难度系数
q	树的结构
T	叶节点数
λ	L2 正则化系数
I_j	第 j 个叶节点的样本集
$H(x)$	多分类模型的最终输出
h_i	单棵决策树分类模型
Y	预测类别
S_w	类内散度矩阵
S_b	类间散度矩阵
w	投影向量

通过 Python 的空白值检测，我们发现数据集中存在缺失值。我们对这些缺失值进行了处理，并确保数据的完整性和准确性，为后续的数据分析和建模提供了可靠的数据基础。

为了避免缺失值对后面的分析以及模型造成影响，我们采用神经网络对已有的数据进行网络构建和训练，如然后对缺失数据进行拟合，将空白值全部补充完毕，所有原始数据没有缺失的数据。

3.3.2 标准化与归一化

本次建模任务所使用的数据集来自 UCI（美国加州大学欧文分校）的 DSA 数据集。该数据集由微型惯性传感器与磁力计在不同人体部位采集而成，记录了 8 名受试者执行 19 种不同活动的传感器信号。每位受试者在每项活动下的采集时长为 5 分钟，经 25Hz 频率采样并划分为 5 秒片段后，共得到 9120 个实例。

数据经过校准与清洗后，整合为包含受试者 ID 与活动类型的 CSV 文件，精确反映了相同时间间隔内不同受试者的活动特征。

为应对传感器数据采集中的潜在变化，并提升所提模型的分类性能，我们对数据样本进行了全面分析。这包括随机选择一个度量指标，并将其与 19 种不同活动进行比较。初始数据集如图 3.1(a)所示，包含该度量指标在各类活动中的 60,000 个样本点。为了保证数据一致性并便于提升分类效果，我们在数据预处理中应用了标准化和归一化方法：

$$X_{new} = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

如图 3.1(b)所示，经过预处理后，可以明显看出所有数据点均落在 0 到 1 的标准化范围内。尽管进行了这种转换，数据的基本特征得以保留。这种细致的预处理不仅减小了传感器读数的潜在偏差，也增强了分类框架在处理多样化活动集合时的鲁棒性。

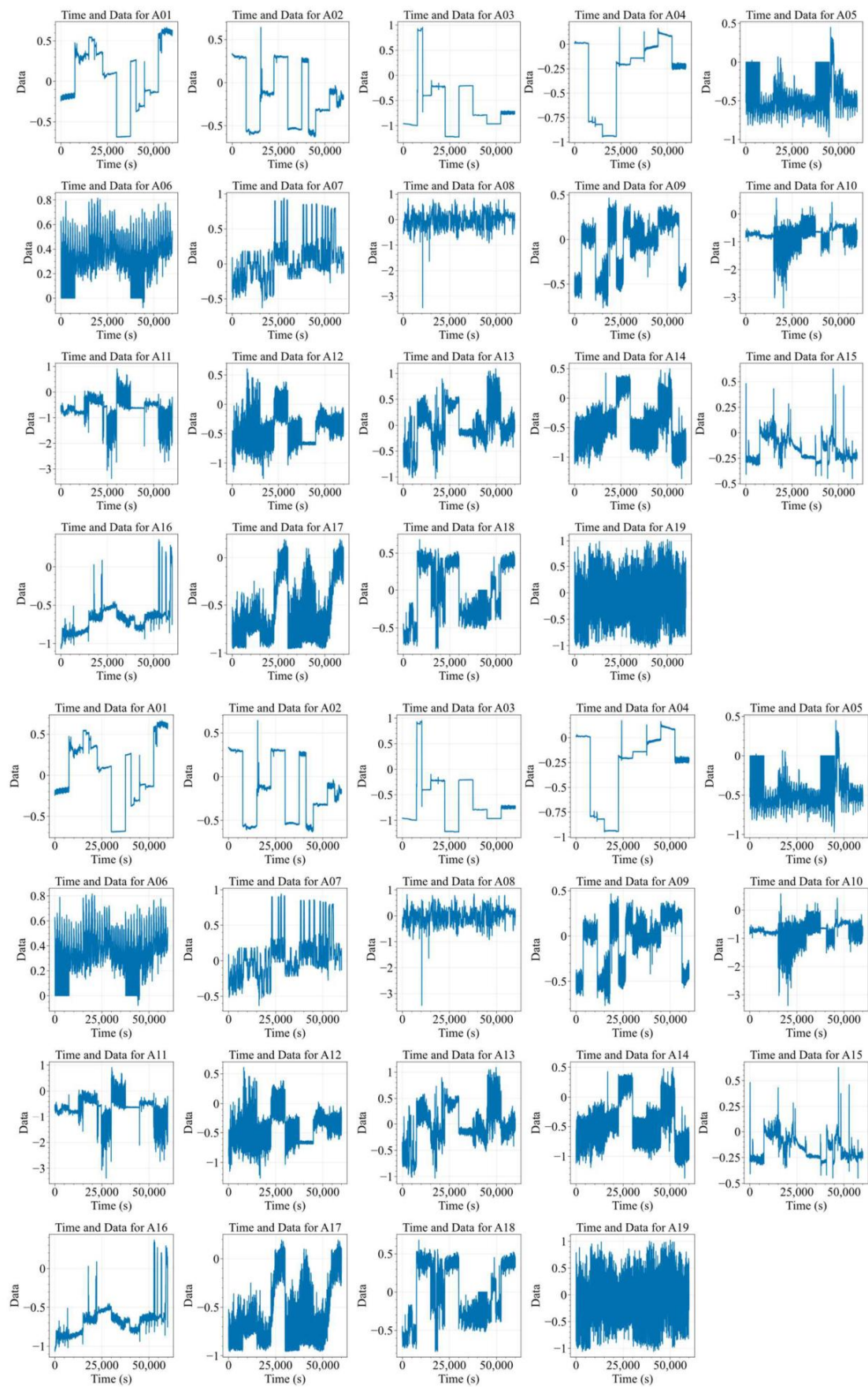


图 3.1 从 A1 到 A19 预处理的预数据和后数据

3.3.3 原始数据格式处理

由于后续需要利用密度聚类方法对数据集进行特征工程处理，本文首先对源数据的活动标签进行了简化。以 UCIDSa 数据集为例，该数据集来源于 UCI 官方网站，原始文件较为分散，因此我们将其整合为统一的 CSV 格式。在预处理中，对“行为”列中的活动名称进行缩写处理，例如以“A1”表示“Sitting”，以便于实验操作与后续分析。同时，受试者 ID、姓名等非必要信息被省略，以简化数据结构，如下表 3.1 所示：

表 3.1 文本中具体行动及其对应编码一览表

行为	简化	行为	简化
Sitting	A1	walkingonatreaddmill withaspeedof4km/h (15deginclinedpositions)	A11
Standing	A2	runningonatreaddmill withaspeedof8km/h	A12
Lyingonback	A3	exercisingonastepper	A13
Lyingonrightside	A4	exercisingonacrosstrainer	A14
Ascendingstairs	A5	cyclingonanexercisebike inhorizontal	A15
Descendingstairs	A6	cyclingonanexercisebike inverticalpositions	A16
Standingin anelevatorstill	A7	rowing	A17
Movingaroundin anelevator	A8	jumping	A18
Walkinginaparkinglot	A9	playingbasketball	A19
Walkingonatreaddmill withaspeedof4km/h (inflat)	A10		

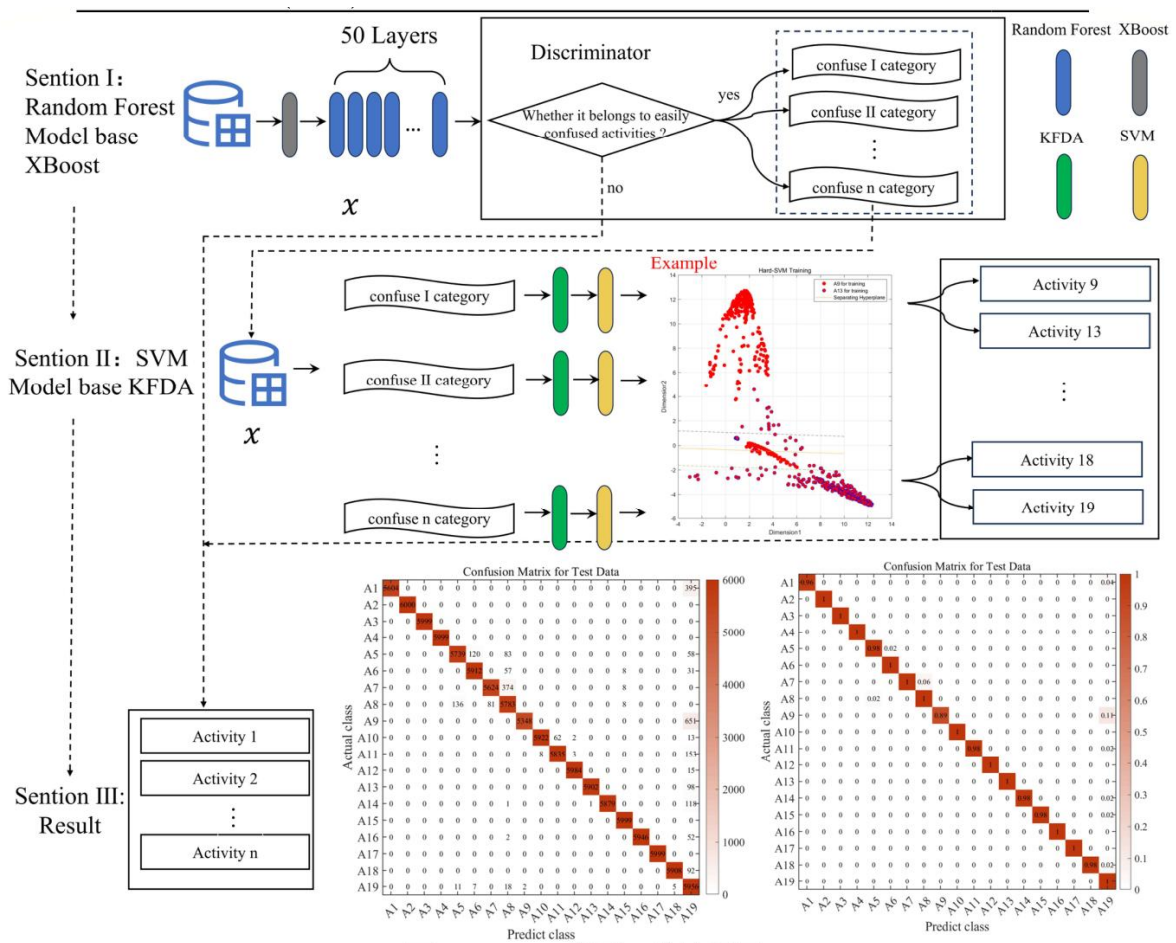


图 3.2 XR-KS 模型工作流程图

图 3.2. XR-KS 模型工作流程图本文提出的模型为 XR-KS，由两层结构组成：第一层（XR 模型）结合 XGBoost 特征选择与随机森林分类，用于初步特征筛选与分类；第二层（KS 模型）采用核 Fisher 判别分析（KFDA）与支持向量机（SVM），进一步解决相似活动间的混淆问题。整个流程如图 3.2 所示。

3.4 基于 XGBoost 特征选择算法的随机森林初始分类模型

为了在后续的二级分类阶段实现更高的初始分类精度. 因此，本文首先利用 XGBoost 特征选择算法提取相关指标，用以评估输入特征的重要性。随后，采用随机森林（RandomForest）作为分类器。随机森林是一种集成学习方法，通过训练多个决策树并进行投票预测来完成分类。在本节中，基于 XGBoost 特征选择算法的随机森林模型如图 3.3 所示。

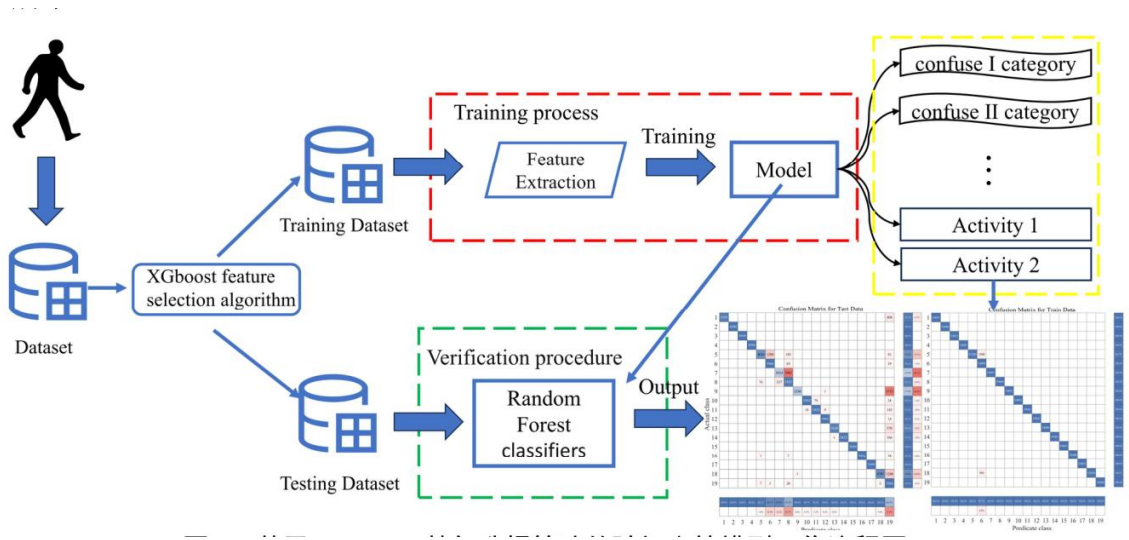


图 3.3 基于 XGBoost 特征选择算法的随机森林模型工作流程图

3.4.1 XGBoost 特征选择算法

XGBoost 由 Chen 等人于 2014 年提出[1]。传统的 GBDT（梯度提升决策树）目标函数是通过不断迭代残差树来逼近目标类别。与之相比，XGBoost 在目标函数中引入了正则化项，从而在减少模型复杂度的同时，有效缓解了过拟合问题并加快了收敛速度。

该模型的目标函数如公式(1)所示：

$$obj^{(t)} = \sum_{i=1}^n (L(y_i, \hat{y}_i) + \Omega(f_i)) \quad (1)$$

其中， $L(y_i, \hat{y}_i)$ 是真实值 y_i 预测值 $\hat{y}_i$ 之间平方差的损失函数， $\Omega(\phi)$ 是正则化项。

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (2)$$

其中， γ 是表示树分裂的难度系数，用于控制树的生成， T 表示叶节点数， λ 表示 L2 正则化系数。

方程(3)目标函数的 Taylor 二阶展开如下：

$$obj^{(t)} = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + C(4)$$

树的定义如下：

$$f_t(x) = w_{q(x)}, w \in R^T, q: R^d \rightarrow \{1, 2, \dots, T\} \quad (5)$$

其中， q 表示树的结构：将输入样本 $x \in R^d$ 映射到叶节点； T 为叶节点数量； w 是长度为 T 的一维向量，表示叶节点的权重。

目标函数可以重写如下：

$$Obj^{(t)} = \sum_{j=1}^T \left[G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T = -1/2 \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T$$

其中， I_j 是第 j 个叶节点的样本集， γ 权重因子。

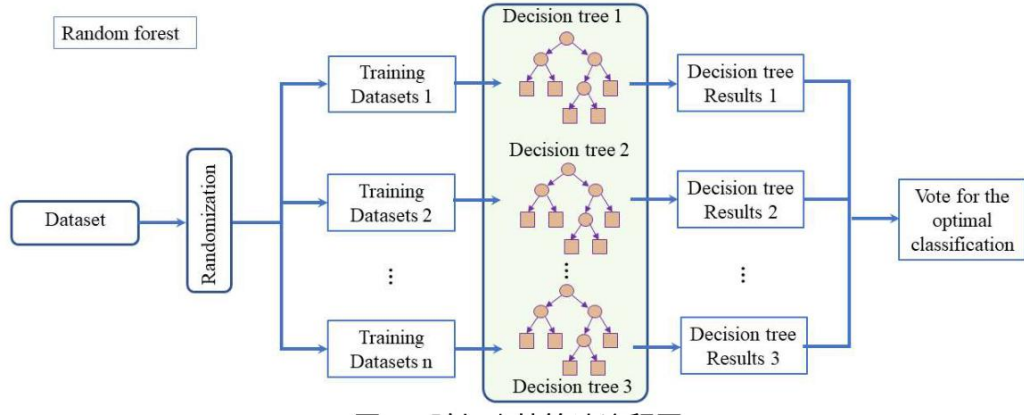


图 3.4 随机森林算法流程图

图 3.4 随机森林算法流程图在前述步骤中，已经提取出能够对人体行为进行分类的主要特征。考虑到这些数据之间的关系，以及训练样本的离散性和数据量庞大，本研究采用随机森林算法进行网络训练。随机森林的一般算法流程如图 3.4 所示。为了初步识别多类别活动，选择在监督学习中表现优异的随机森林分类算法作为第一层分类器。

3.4.2 基于 XGBoost 特征选择算法的随机森林

随机森林是一种由多个决策树分类器组成的集成分类模型 $\{h(X, \Xi_k), k = 1, \dots\}$ ，其参数集 $\{\Xi_k\}$ 由相互独立且同分布的随机向量构成。在给定自变量 x 的条件下，每棵决策树分类模型通过多数投票的方式选择最优分类结果。其基本思想是：首先采用自助采样（bootstrapsampling）方法，从原始训练集抽取 k 个样本，每个样本的规模与原始训练集相同；然后基于这 k 个训练样本分别构建 k 个决策树模型，从而得到 k 个不同的分类结果；最后根据这 k 个分类结果，通过多数投票（majorityvoting）确定每个样本的最终分类结果。

在训练过程中，随机森林的最终分类决策通过迭代 k 轮得到一系列分类模型 $\{h_1(X), h_2(X), \dots, h_k(X)\}$ 并以此构建一个多分类模型系统。该系统的最终分类结果通过简单的多数投票法获得：

$$H(x) = \underset{Y}{\operatorname{argmax}} \sum_{i=1}^k I(h_i(x) = Y) \quad (7)$$

其中， $H(x)$ 表示多分类模型的最终输出， $h_i(x)$ 为单棵决策树分类模型，输出变量 Y 为预测类别。

3.5 核 Fisher 判别分析的二层 SVM 分类

从图 3.5 可以看出，在区分“上楼梯”和“下楼梯”等动作时存在一定难度，这是由于这些动作在细节层面上具有较强的相似性。同时，图 3.6 显示，基于 PCA 的降维以及原始特征中较小的类内距离，都会对分类效果造成阻碍。为了缓解相似动作之间的混淆，本文提出了两项关键措施：首先，采用核 Fisher 判别分析对特征进行降维，以提升对相似活动的区分能力。通过在数据空间中拉大不同动作类别之间的间隔，从而为后续的 SVM 分类奠定基础。其次，在完成特征变换与判别空间构建后，将处理后的特征输入到支持向量机（SVM）分类器中，以实现更高精度的行为识别。整体工作流程如图 3.7 所示。

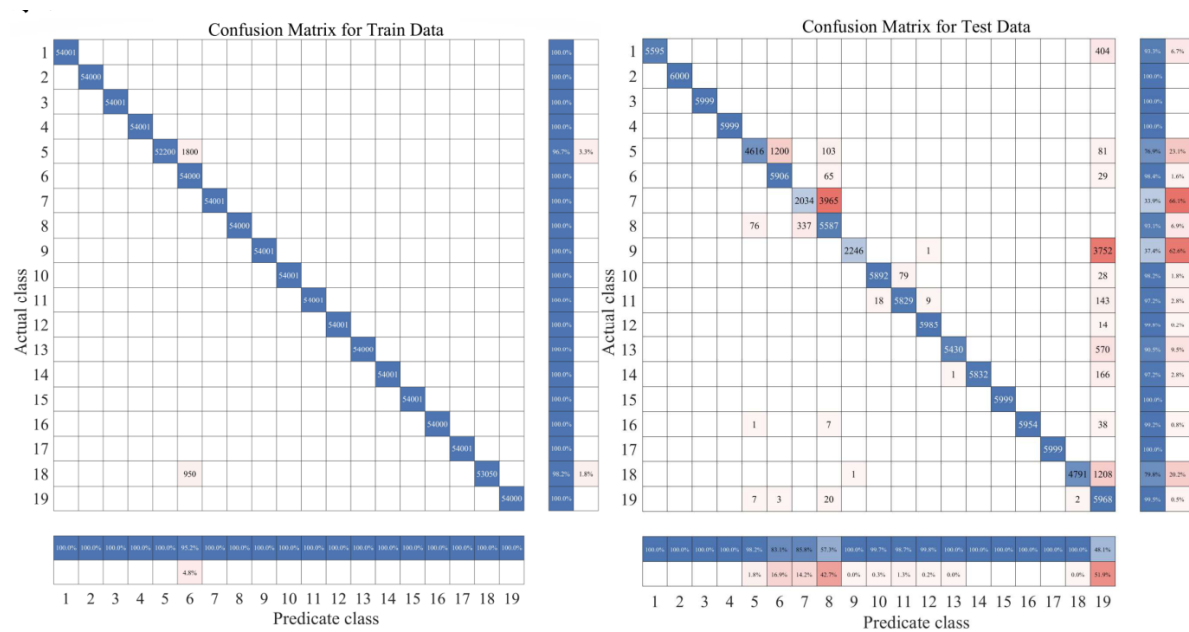


图 3.5 测试集和训练集随机森林训练的混淆矩阵（实验设置与图 9 相同）

(a) 训练集下随机森林模型的混淆矩阵； (b) 测试集下随机森林模型的混淆矩阵。

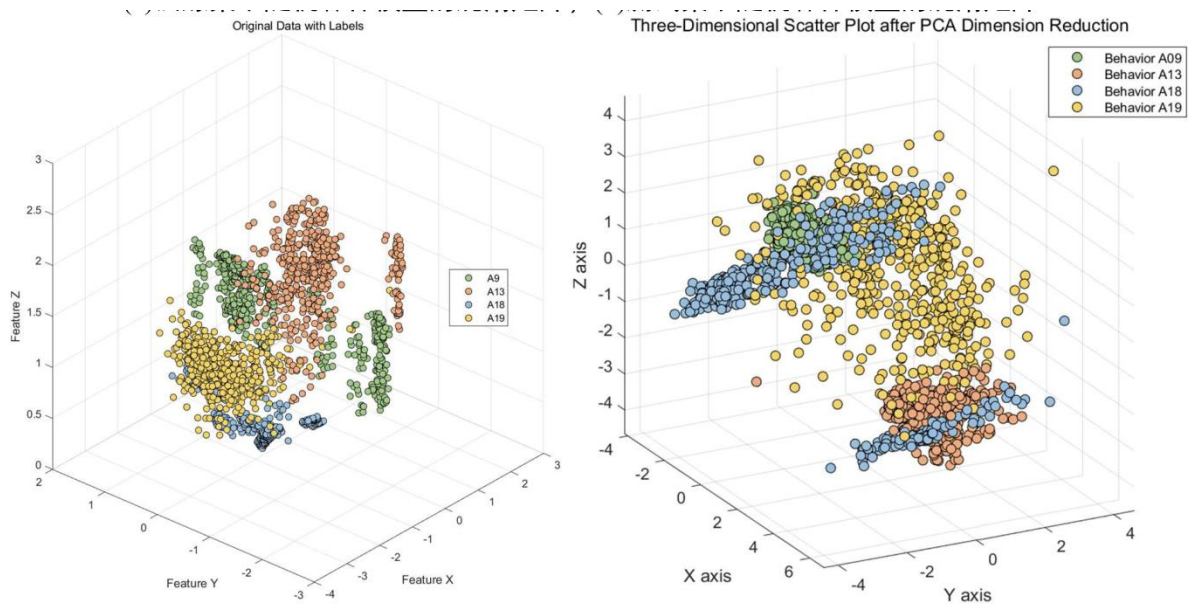


图 3.6 原始特征与主成分特征

(a) 原始数据；(b) PCA 降维后的数据。

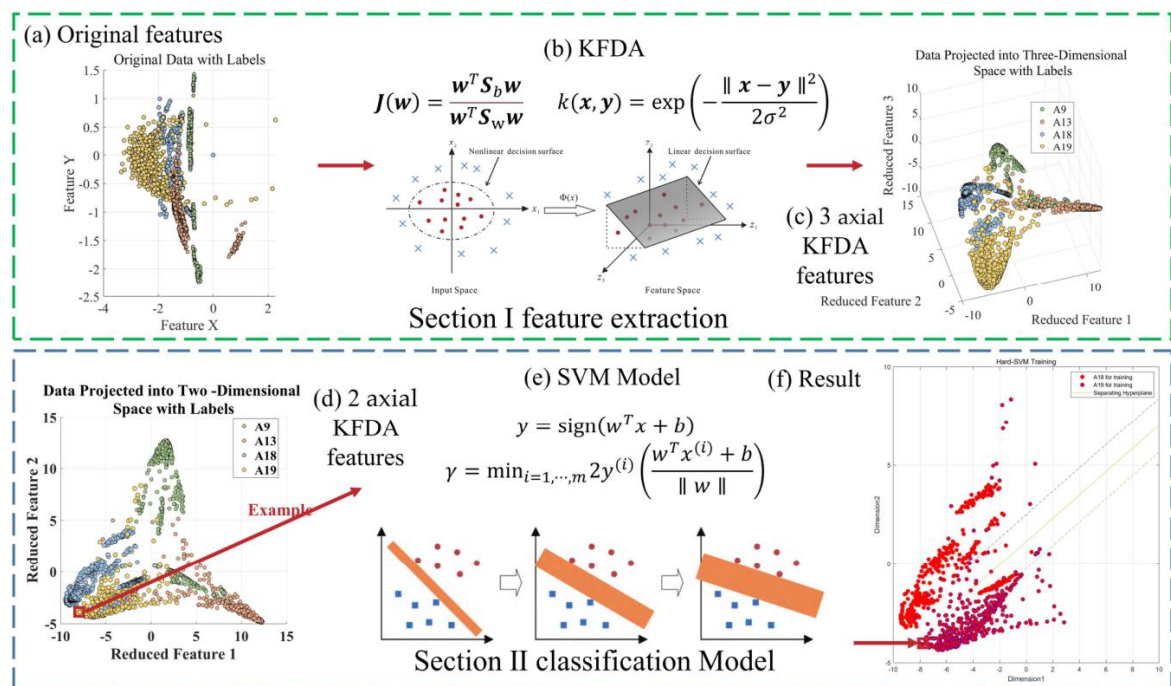


图 3.7 基于核 Fisher 判别分析的 SVM 模型工作流程

(二级分类模型确保(a)相似活动的紧密聚类；(b)经过 KFDA 处理后，(c)生成三轴数据与图像；(d)获得二维化的双轴图像数据；(e)进行 SVM 分类；(f)得出最终结果。)

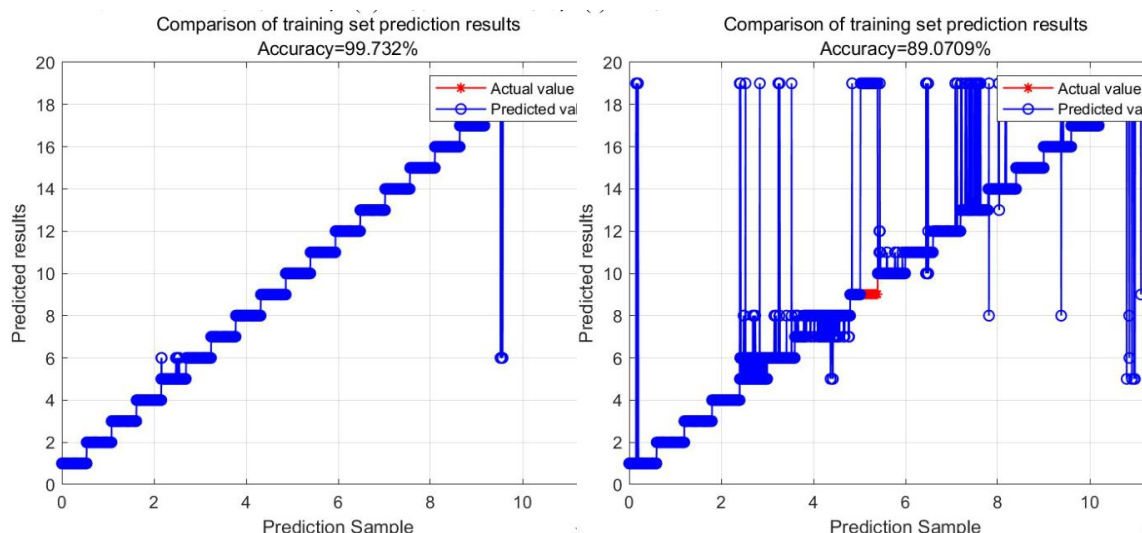


图 3.8 随机森林模型识别结果对比图（训练集与测试集比例设定为 9:1，实验重复进行五次）

(a) 基于训练集的随机森林模型结果；(b) 基于测试集的随机森林模型结果。

KFDA 是一种基于核技术的模式识别和分类方法，是 Fisher 判别分析的扩展。KFDA 旨在通过将数据映射到高维特征空间来处理非线性可分的数据，从而提高分类性能。我们结合文献[2]对 KFDA 进行描述。核 Fisher 判别分析（KFDA）最早由 Schölkopf 等人在 1997 年提出[3]，可表示为最大化方程（7）：

$$J(w) = \frac{w^T S_b w}{w^T S_w w} \quad (7)$$

其中 S_w 表示类内散度矩阵， S_b 是类间散度矩阵， w 表示投影向量。

上述问题可以等价于求解以下广义特征值问题的特征向量：

$$S_b v_i = \lambda_i S_w v_i \quad (8)$$

其中，特征值 λ_i 表示每个投影向量的判别能力。一旦得到投影向量 v ，即可用其代替原始向量，通过线性分类器进行分类。

LDA 方法的局限性主要源于其固有的线性特性，尤其在处理非线性问题时[37]。相比之下，KFDA 方法作为使用核技巧的 LD 改进版本，克服了这些局限性。KFDA 更适合高维数据和复杂系统分析，易于实现，并具有适应性和泛化性。

KFDA 的核心概念是通过非线性映射函数 ϕ 将原始输入数据映射到高维特征空间 F ，通常为非线性空间（见图 3.9）。通过该转换，输入数据中的非线性关系被间接转化为线性关

系。随后在该特征空间中应用 LDA，以提取最显著的判别特征。为克服计算 Φ 的计算挑战，研究者引入核函数参数以表达非线性映射的函数关系。

KFDA 的目标是在特征空间中寻找一组投影向量，使类间距离最大化、类内距离最小化。这一目标通过最大化核 Fisher 判别准则实现：

$$J(\alpha) = \frac{\alpha^T K_b \alpha}{\alpha^T K_w \alpha} \quad (9)$$

其中 α 表示投影向量， K_b 表示特征空间中的核类间散度矩阵， K_w 是特征空间中的核类内散度矩阵。

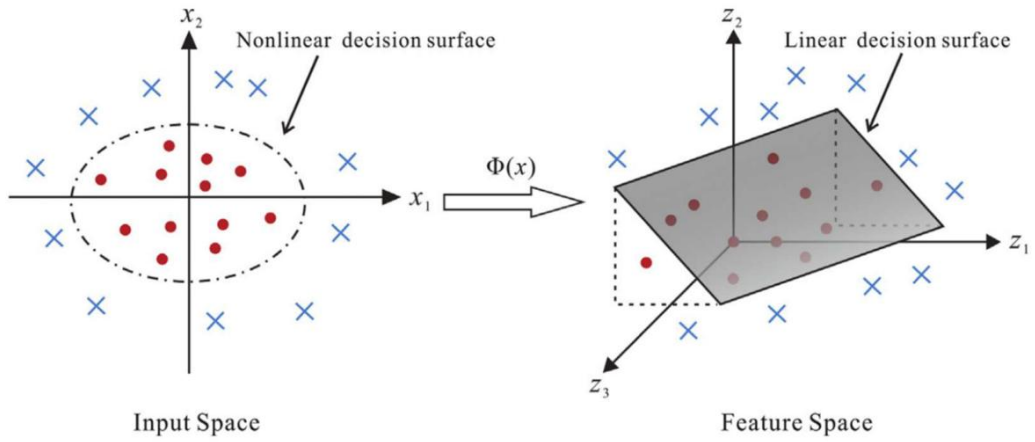


图 3.9 KFD 模型的转换过程示意图

非线性映射函数将原始（低维）输入空间中的非线性问题转换为（高维）特征空间中的线性问题。

方程（9）中描述的信息可以重新表述为求解广义特征方程，从而降低冗余：

$$K_b \alpha = \lambda K_w \alpha \quad (10)$$

其中 λ 为投影向量 α 的非零特征值。令 $\alpha_{opt} = (\alpha_1, \dots, \alpha_M)$ 为最优投影向量，其同时也是方程（4）中的最大特征值。 $\lambda_1, \dots, \lambda_M$ 分别为 $\alpha_1, \dots, \alpha_M$ 的特征值，向量 $\lambda_1 \geq \dots \geq \lambda_M$ 的数量通过累计贡献率计算。

若已知 α_{opt} ，KFDA 的非线性判别函数 $f(x)$ 如下：

$$f(x) = \sum_{i=1}^n \alpha_i k(x_i, x) \quad (11)$$

其中 α_i 是第 i 个核的系数向量， x_i 是所有输入样本中的第 i 个样本， k 为核函数。

3.5.2 核参数优化

在这些核函数中，高斯核因其强大的泛化能力以及所需参数较少而突出。这使其在捕捉非线性关系方面特别有效。因此，高斯核函数被选为核函数，其表达式如方程（12）所示：

$$k(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \quad (12)$$

其中 σ 是高斯核的宽度参数。

在 KFDA-SVM 模型中，核参数 σ 至关重要，它影响数据在特征空间中的位置和分布，对 SVM 模型的分类效率和泛化能力有显著影响。图 3.10 展示了核参数的重要性，并给出了实验值 0.5、1、2 和 5。选择合适的核参数值是获得最优结果的关键步骤。

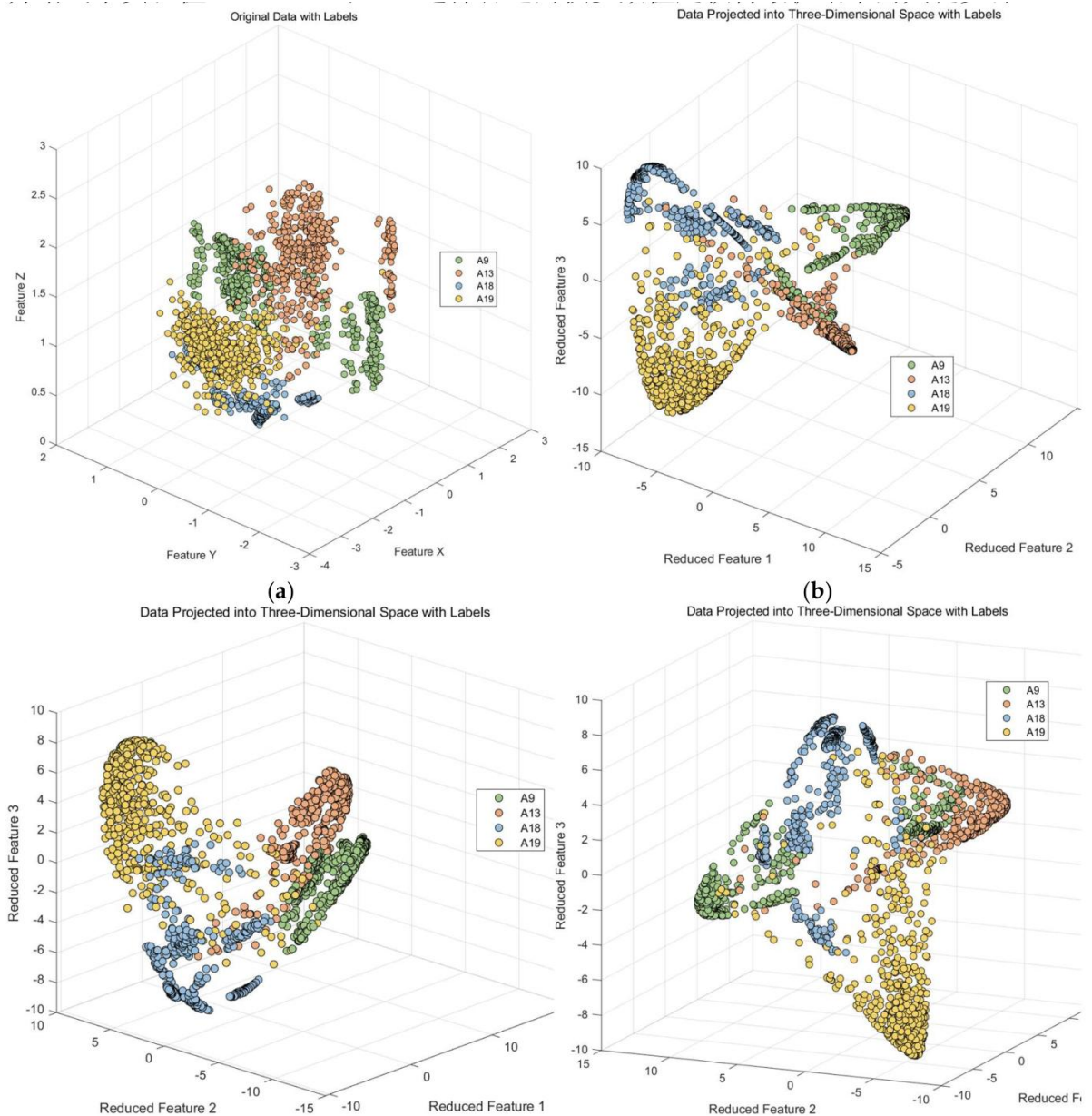


图 3.10 不同参数下核 Fisher 判别分析的结果 (a) - (d)

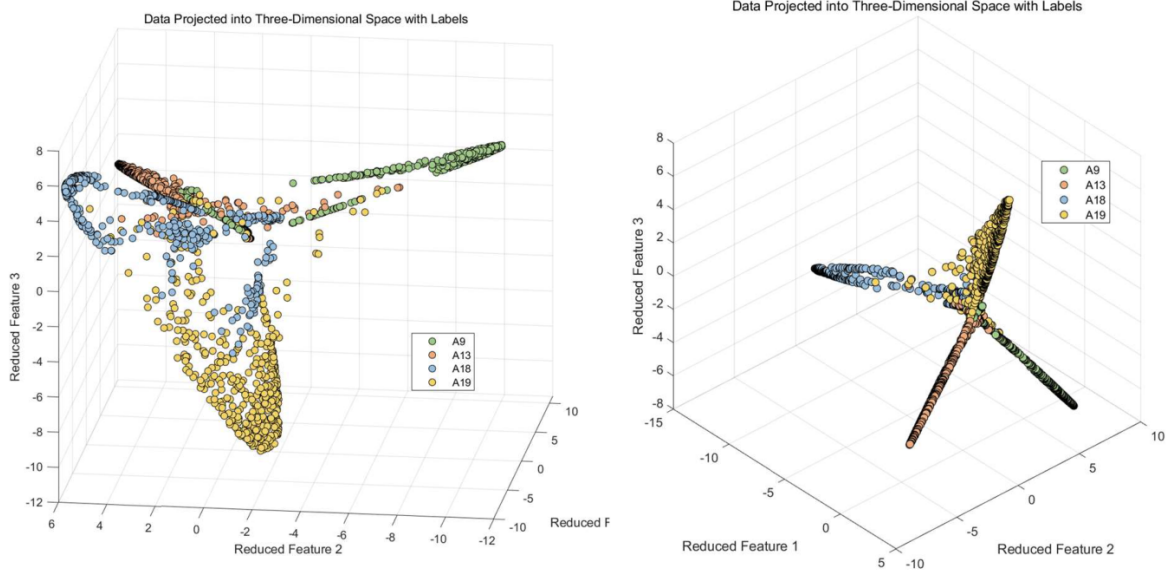


图 3.10 不同参数下核 Fisher 判别分析的结果 (e) - (f)

(a) $\sigma = N/A$; (b) $\sigma = 1.53$; (c) $\sigma = 0.5$; (d) $\sigma = 1.0$; (e) $\sigma = 2.0$; (f) $\sigma = 5.0$ 。

为了进一步将混淆动作划分为具体动作，本文引入支持向量机（SVM）模型作为混淆动作的子分类模型。SVM 分类器的原理是寻找一个超平面，使不同类别之间的距离最大化，从而实现有效分类。如图 3.11 所示，增加分类间隔的宽度（即最大化间隔）能够减小训练集中局部干扰的影响。因此，可以认为该分类方法具有最佳的泛化性能和整体适用性。SVM 模型可表述如下：

$$y = \text{sign}(w^T x + b) \quad (13)$$

当 $y=1$ 时，样本为正类；当 $y=-1$ 时，样本为负类，即：

$$\begin{cases} w^T x + b > 0, & y = 1, \\ w^T x + b \leq 0, & y = -1. \end{cases} \quad (14)$$

如图 3.11 所示，SVM 通常通过最大化分类间隔来寻找最优分类超平面。假设训练集的输入由一组 $x(i)$ 向量组成，输出为对应的 $y(i)$ 向量，则分类间隔等于所有样本到超平面的最小距离的两倍，其表达式如下：

$$\gamma = \min_{i=1, \dots, m} 2y^{(i)} \left(\frac{w^T x^{(i)} + b}{\|w\|} \right) \quad (15)$$

其中， m 为样本数量。

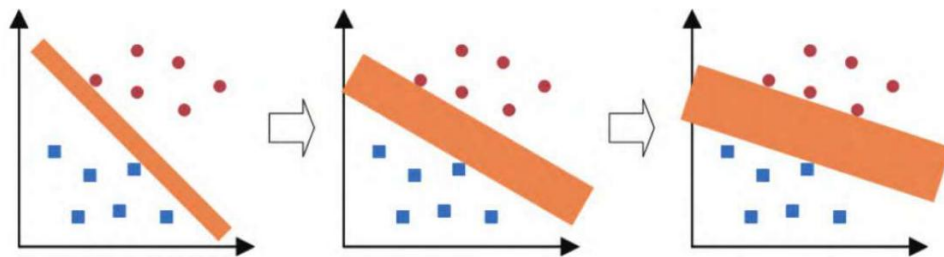


图 3.11 向量机分类流程图

（图中蓝色和红色点表示不同类别，SVM 通过不断迭代优化，寻找最优分类超平面。）

从数学上讲，所有满足方程（15）的样本点（即到分类超平面欧几里得距离最小的样本点）将被定义为支持向量。因此，这些样本集合必须满足以下两种情况：样本为正类，或样本为负类，如图 3.12 所示。

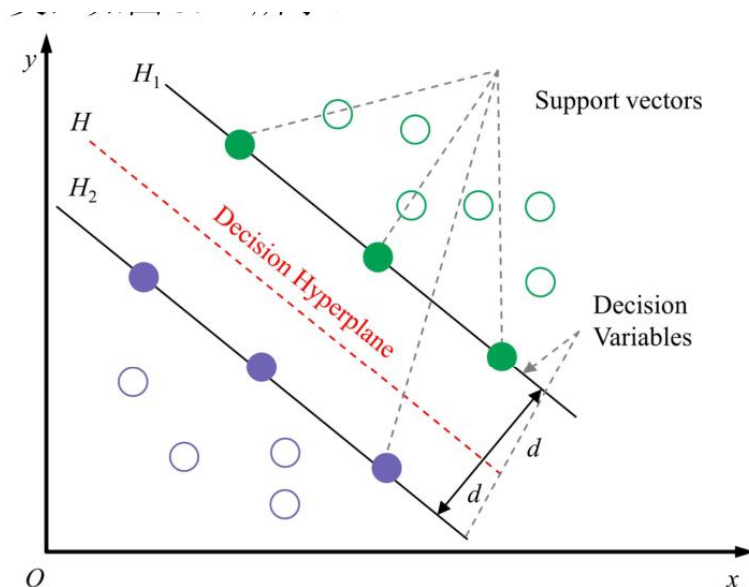


图 3.12 向量机分类示意图

（不同颜色表示正负样本。空心圆标记支持向量，即最接近分隔超平面的点。实心标记表示距离超平面较远的样本。）

$$y^{(i)}(w^T x^{(i)} + b) \geq 1. (16)$$

4. 模型求解

4.1 实验环境

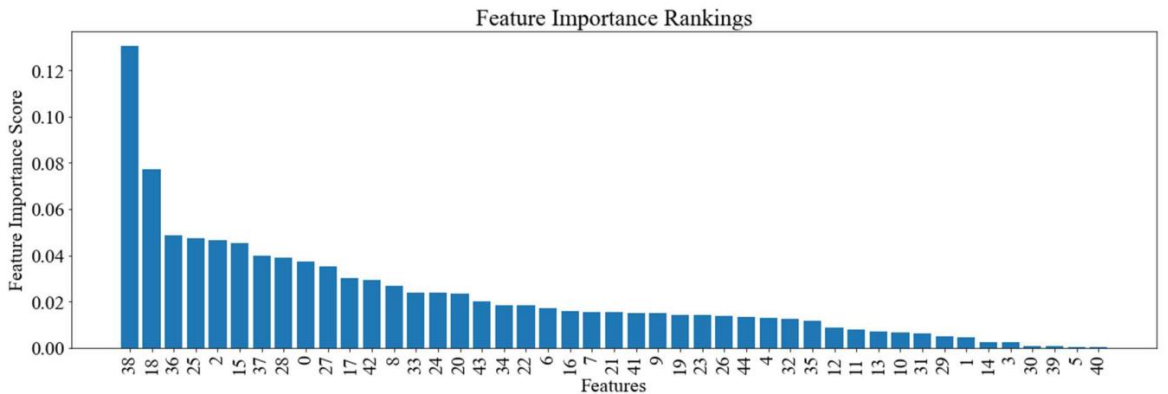
本研究的实验在中国桂林进行，实验平台为一台 ASUS 计算机，具体配置如下：AMD Ryzen 74800H 处理器（集成 Radeon 显卡），主频 2.90GHz，16GB 内存，以及

NVIDIA GeForce GTX 1660 Ti 独立显卡。操作系统为 Windows 10。实验过程中分别使用北太天元与 Python 3.9.7 工具对所提出的方法进行实现与验证。实验所用数据集包括四类：UCIDSA、UCIHAR、WISDM 以及 UCIADL。我们还对所提方法进行了全面的性能评估。为保持论文叙述的简洁性，以下实验结果主要以 UCIDSA 数据集为例进行展示。

4.2 重要特征提取

基于 XGBoost 特征选择算法，对数据集中的 45 个特征进行重要性分析，以评估各特征的相对贡献度。由图 4.1(b) 可见，我们以前 n 个重要性排序靠前的特征作为随机森林模型第一层的输入，并在不同 n 的取值下获得了实验结果。这些结果有效验证了不同特征组合下模型在准确率与计算耗时方面的表现。基于此，我们从图 4.1(a) 的结果中选取前 31 个特征，作为后续基于广义判别分析 (Generalized Discriminant Analysis, GDA) 的多层分类器输入；而低重要度的特征则被剔除，以减小对分类精度可能造成的干扰。

重要特征权重的直方图及特征筛选实验结果见图 4.1，部分实验结果汇总于表 4.1。



(a)

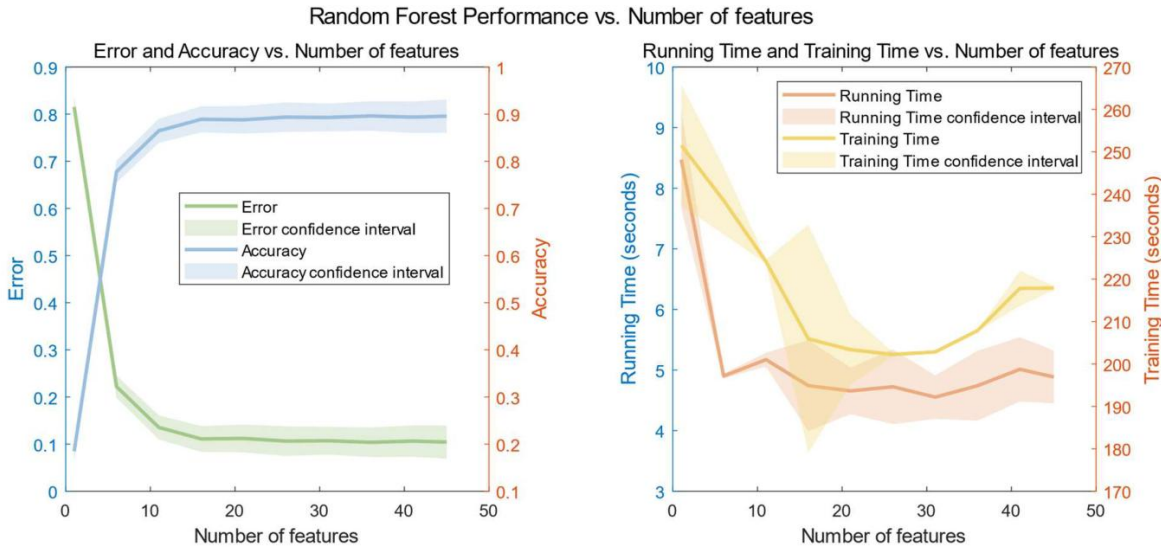


图 4.1 重要特征权重分布及不同特征数量对随机模型性能的影响

表 4.1 特征数量与误差率、准确率、训练次数及运行时间（均值±标准差）

特征数量	误差率	准确率	训练次数(s)	运行时间(s)
1	81.5438(±1.13)%	18.4561(±1.13)%	251.53(±7.292)	8.1802(±0.165)
6	22.1995(±1.24)%	77.8004(±1.24)%	238.55(±4.063)	4.9149(±0.08)
21	11.1931(±1.49)%	.88.8068(±1.49)%	203.36(±4.209)	4.5172(±0.78)
31	10.7098(±1.62)%	89.2901(±1.62)%	202.83(±0.151)	4.4230(±0.73)
41	10.6396(±1.79)%	89.3603(±1.79)%	217.82(±2.107)	4.8204(±0.68)
45	10.4229(±1.86)%	89.5770(±1.86)%	217.89(±0.278)	4.7276(±0.89)

首先，在北太天元环境下调用随机森林模型，绘制了决策树数量与误差率及运行时间之间关系的曲线，如图 4.2 所示。具体实验结果见表 4.2。

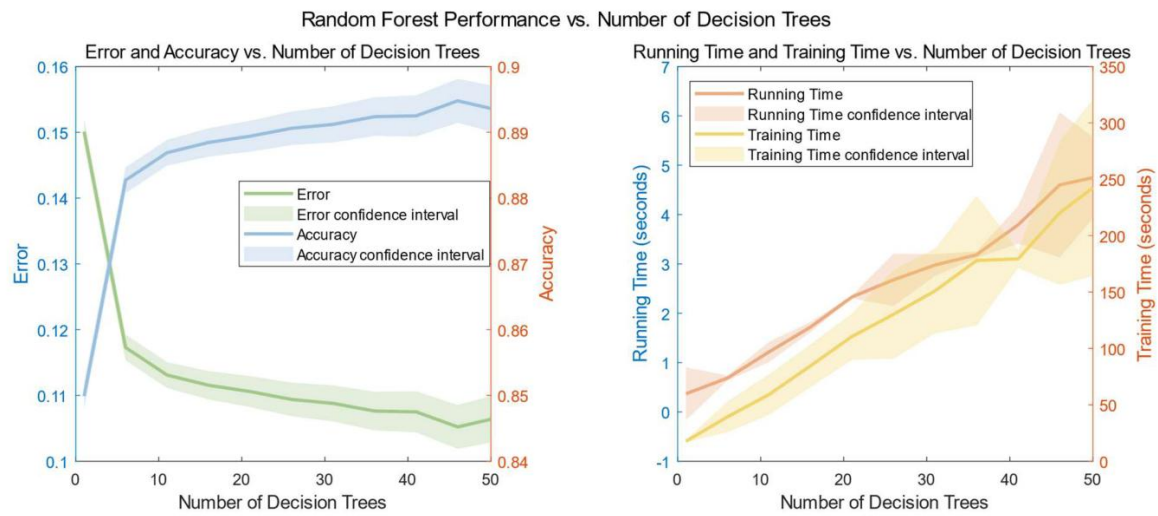


图 4.2 随机森林模型中决策树数量与误差的关系曲线

表 4.2 决策树数量与误差率、准确率、训练次数及运行时间（均值±标准差）

决策树数量	误差率	准确率	训练次数(s)	运行时间(s)

决策树数量	误差率	准确率	训练次数(s)	运行时间(s)
1	15.0119(± 1.12)%	84.9881(± 1.12)%	13.3845(± 1.507)	0.2793(± 0.126)
6	11.7283(± 1.25)%	88.2717(± 1.25)%	32.5361(± 0.580)	0.7068(± 0.062)
11	11.3098(± 1.39)%	88.6902(± 1.39)%	53.4840(± 0.665)	1.1963(± 0.019)
21	11.0589(± 1.51)%	88.9411(± 1.51)%	95.7831(± 0.345)	2.2481(± 0.059)
31	10.8791(± 1.46)%	89.1209(± 1.46)%	141.0272(± 4.273)	3.2747(± 0.017)
41	10.7501(± 1.81)%	89.2499(± 1.81)%	187.3081(± 10.173)	4.4448(± 0.106)
50	10.5211(± 1.95)%	89.4789(± 1.95)%	215.6206(± 20.992)	5.0635(± 1.308)

通过对图 4.2 的分析，在综合考虑计算机性能、模型精度以及整体模型可靠性的基础上，我们得出结论：对于该数据集而言，最优的决策树数量为 50。为评估随机森林模型的分类能力，我们构建了专门的测试集。在北太天元环境下，将整个数据集按照不同的人类活动类型进行均匀划分，形成多种不同的训练/测试集比例，并在此基础上开展实验。实验结果总结于表 4.3，展示了不同划分比例对训练集与测试集精度的影响。

此外，为进一步分析上述随机森林的识别结果，我们在北太天元中绘制了混淆矩阵图。

表 4.3 基于 XGBoost 特征选择算法的随机森林模型在 UCI DSA 数据集上的结果表

比例(训练集:测试集)	训练集	测试集	运行时间(s)
9:1	99.7322(± 0.08)%	89.0709(± 1.59)%	5.124(± 0.29)
8:2	99.5669(± 0.12)%	85.1812(± 1.63)%	5.243(± 0.28)
7:3	99.3541(± 0.13)%	79.3321(± 1.76)%	5.165(± 0.36)
6:4	99.2145(± 0.15)%	74.0732(± 1.86)%	5.146(± 0.15)
5:5	99.6656(± 0.17)%	69.5766(± 2.12)%	5.234(± 0.34)

比例(训练集:测试集)	训练集	测试集	运行时间(s)
4:6	99.6706(± 0.21)%	68.6753(± 2.65)%	5.091(± 0.39)

剩余未被识别的动作主要分为两类。模型将 A9、A13 和 A18 分类为 A19，并将 A7 和 A8 相互混淆。为了便于后续细粒度分类模型的构建，这些相似的动作被分为两大类，如表 4.4 所示。

表 4.4 易混淆动作分类表

混淆类别	易混淆动作
混淆 I 类	A9, A13, A18, A19
混淆 II 类	A5, A6
混淆 III 类	A7, A8

4.4 基于核 Fisher 判别分析的支持向量机模型特征提取

以四个行为类别（A9、A13、A18 和 A19）作为易混淆类别 I 的示例，我们首先从数据集中提取了三个最重要的特征，并创建了散点图，如图 4.3 左侧所示。可以观察到，这四个行为类别在这些原始特征上的空间分布相对较短，表明类间距离较小，而类内距离较大。这不利于分类器进行活动识别。

随后，我们应用了主成分分析（PCA）进行降维，如图 4.3 右侧所示。它表示从这四个相似活动的原始特征中随机提取的三个非线性判别特征。在本研究中，我们发现 PCA 的映射结果并不十分理想，因为类内距离仍然较小。

因此，在本研究中，我们采用了核 Fisher 判别分析进行降维，重点关注在随机森林上层被误分类的点。核 Fisher 判别分析有一个参数，记作 σ ，该参数可以变化。通常，该参数的取值范围设定在[0, 10]之间。我们进行了不同参数设置的实验，并获得了多幅图像，如图 3.11 所示。

根据上述实验结果，我们可以看到，在这种情况下，通过观察三维散点图，数据已经被分为四个类别。为了获得更清晰的视觉表示，我们选择了在三维空间中表现最佳的两个特征，并生成了二维散点图，如图 4.3 所示。

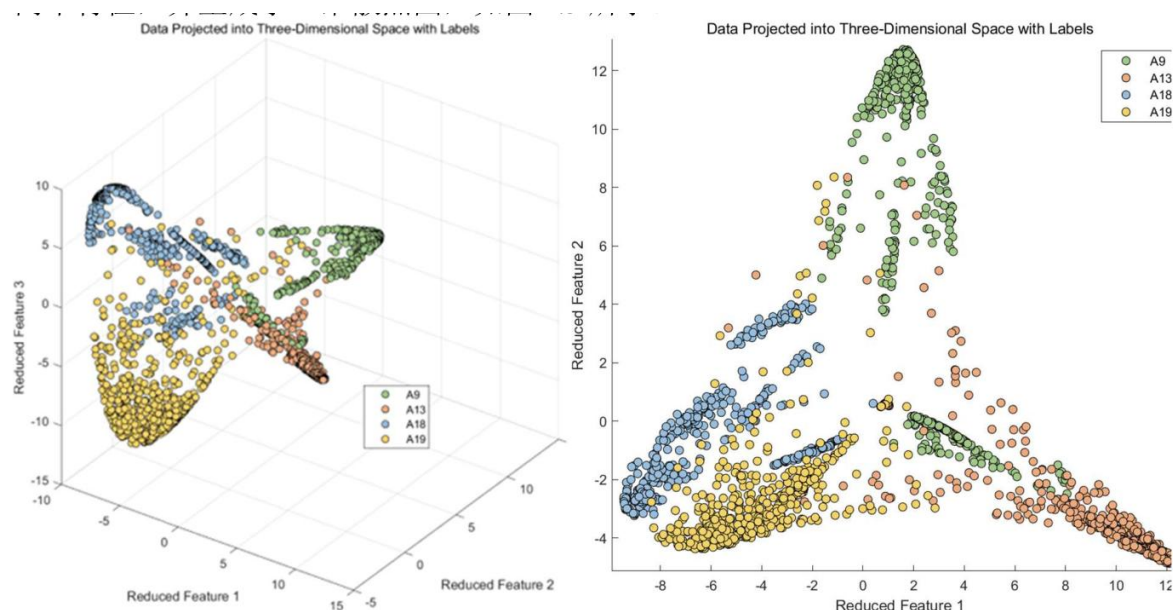


图 4.3 核 Fisher 判别分析特征

(a) 三轴数据；(b) 双轴数据。

在本文中，我们利用北太天元对上述模型进行子分类，并将已在判别分析中泛化的指标输入支持向量机（SVM）作为原始数据，以易混淆类别 I 为例，因为 A9 和 A13 之间的联系更为紧密，A18 和 A19 之间的联系也更为紧密。我们首先将易混淆类别 I 细分为两个大类，即 A9、A13 和 A18、A19，然后通过双层支持向量机进行第二次细分，将易混淆类别 I 进一步细分为更多类别，即 A9、A13、A18 和 A19。这四种活动类别分别为 A9、A13、A18 和 A19，如图 4.4 所示。

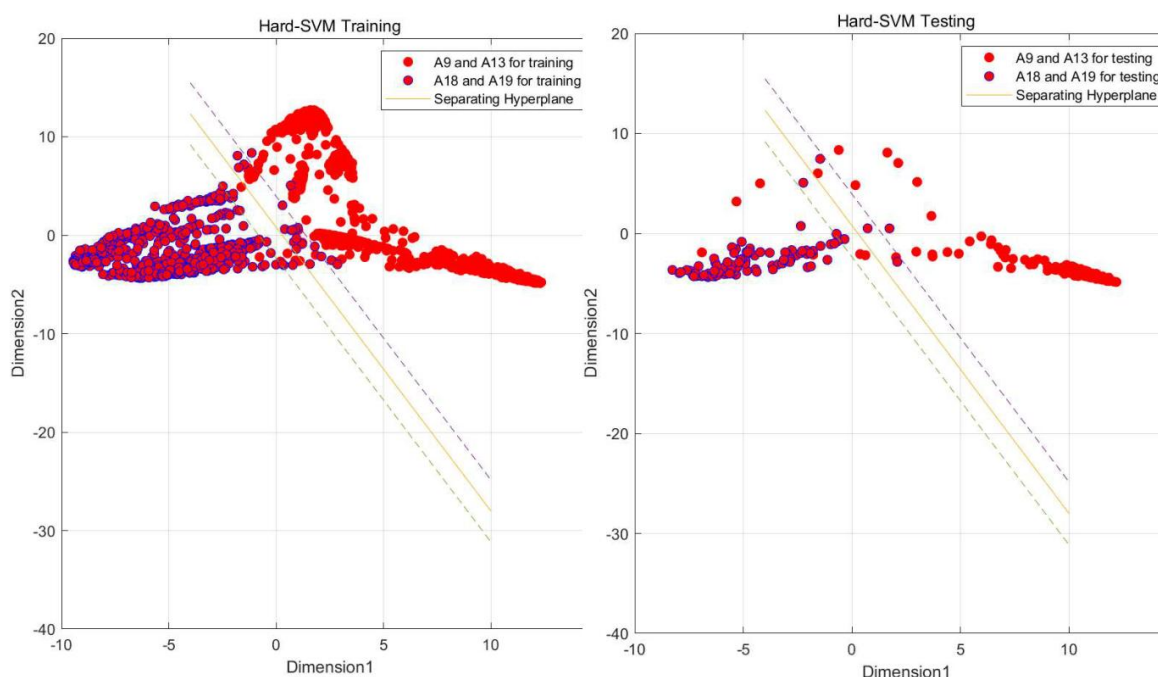


图 4.4 SVM 初步分割 A9、A13 和 A18、A19 的结果图

通过上述步骤，利用支持向量机（SVM）将易混淆类别 I 的数据分类为两个主要类别：A9、A13 和 A18、A19。为了实现更精确的分类，本文进一步对这两个主要类别进行细分类，以识别具体的活动类别，如图 4.4 至图 4.6 所示。

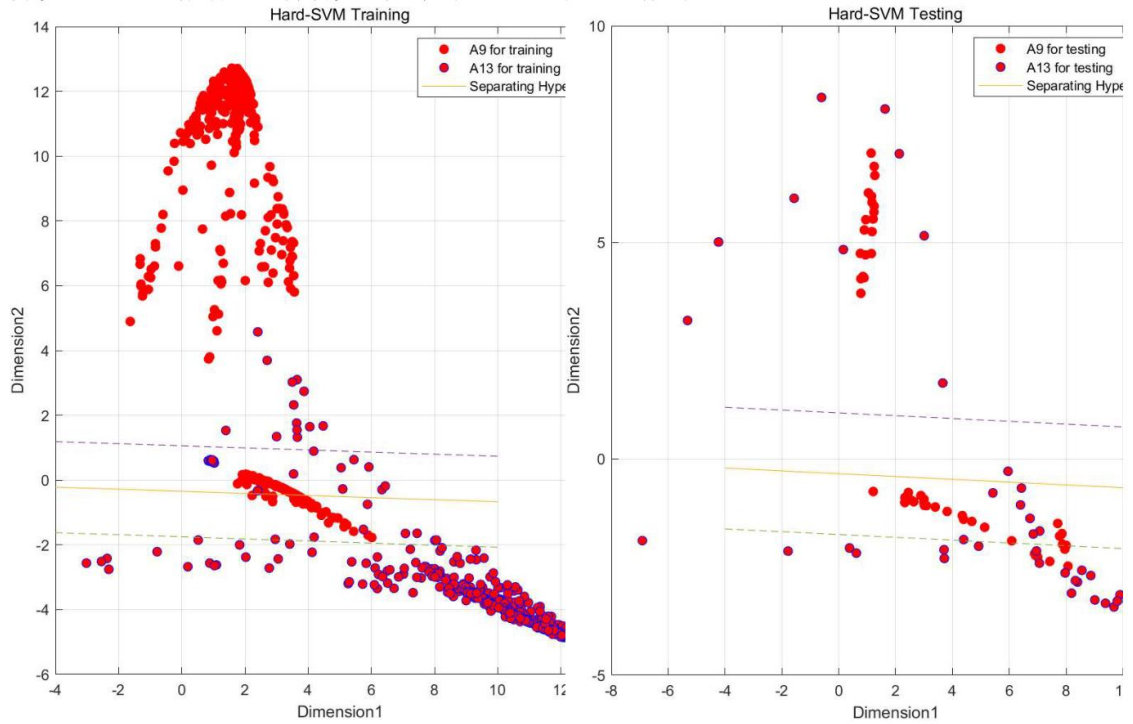


图 4.5 SVM 初步分割 A9 和 A13 的结果图

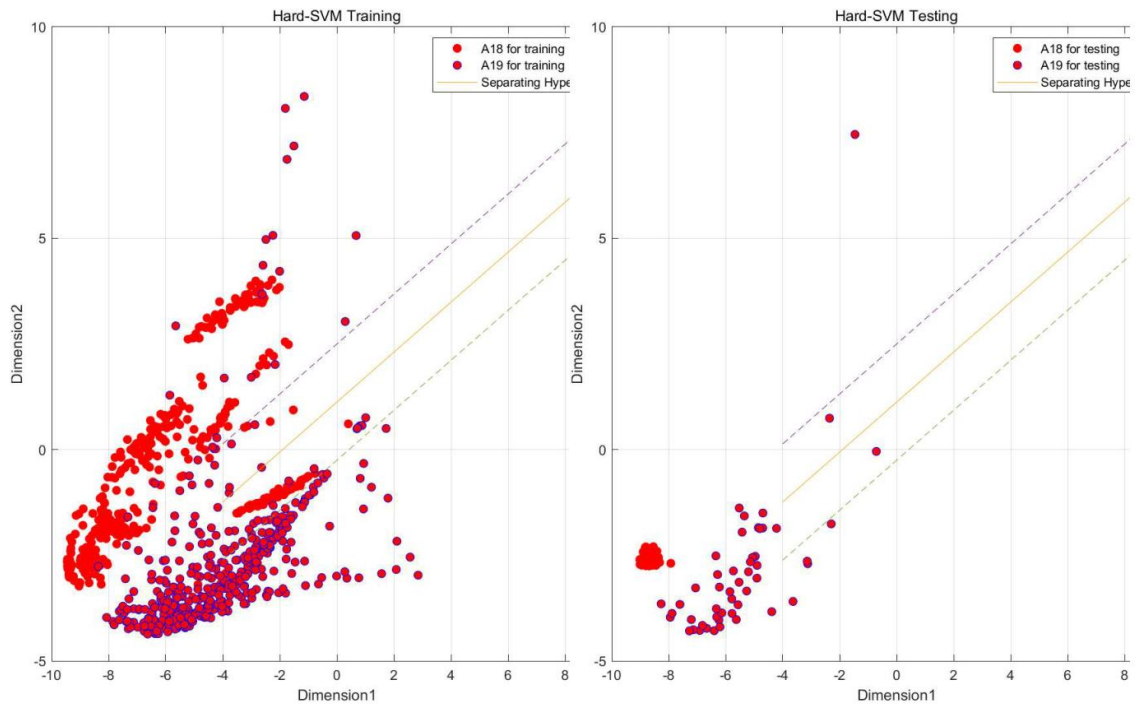


图 4.6 SVM 初步分割 A18 和 A19 的结果图

通过与图相关的步骤，我们能够使用支持向量机（SVM）的精细分类将混淆类别 I 中的所有数据分类到具体的活动类别中。尽管 SVM 精细分类 A9 和 A13 的效果并不显著，如图 4.6 所示，但仍然比最初的随机森林分类效果要好得多。同样，我们进行了各种实验，如表 4.5 所示。

表 4.5 随机森林模型在 UCI DSA 数据集之间的结果表

比例 (训练 : 测试)	A9, A13 和 A18, A19 训练集	测试集	运行时间 (秒)	A9 和 A13 训练集	测试集	运行时间 (秒)	A18 和 A19 训练 集	测试集	运行时间 (秒)
9: 1	97.9444(±0.13)%	95.0(±0 .16)%	0.9401(±0.15)	84.0(±0 .15)%	59.0(±0 .16)%	0.1802(±0.071)	86.4444(±0.14)%	92.0(±0 .16)%	0.1605(±0.017)
8: 2	98.0(±0 .11)%	96.2512(±0.13)%	0.8749(±0.13)	79.8714(±0.19)%	70.0(±0 .19)%	0.1524(±0.045)	84.8751(±0.18)%	95.50(± 0.14)%	0.1413(±0.065)
7: 3	98.0(±0 .15)%	97.0(±0 .15)%	0.4801(±0.14)	76.0(±0 .16)%	79.6667(±0.13)%	0.0953(±0.065)	83.7143(±0.16)%	94.6667(±0.16)%	0.09714(±0.094)
6: 4	97.8333(±0.14)%	97.75(± 0.14)%	0.4146(±0.16)	73.0(±0 .16)%	85.0(±0 .12)%	0.07241(±0.014)	83.8333(±0.16)%	92.0(±0 .17)%	0.07048(±0.053)
5: 5	99.90(± 0.14)%	96.95(± 0.17)%	0.4521(±0.091)	65.80(± 0.14)%	87.60(± 0.11)%	0.05512(±0.069)	93.40(± 0.16)%	83.40(± 0.17)%	0.05041(±0.047)
4: 6	99.91(± 0.14)%	97.41(± 0.14)%	0.2413(±0.075)	84.75(± 0.14)%	85.0(±0 .17)%	0.03338(±0.064)	92.75(± 0.19)%	85.3333(±0.16)%	0.02581(±0.034)

5. 结果分析

5.1 综合分析

在本研究中，我们针对易混淆类别 I 进行了进一步处理，并将其作为输入传递至支持向量机的第二层分类器。该步骤输出了四类相似人类行为活动的识别概率。最终的识别结果通过加权平均法确定，该方法综合考虑了第一层分类器的预测概率，以提升整体判别的准确性。

图 5.1 所示的混淆矩阵表明，经过该方法处理后，原本容易混淆的动作类别得到了显著区分。总体识别准确率从 89.07% 提升至 97.69%，验证了所提模型在解决相似行为识别问题上的有效性。

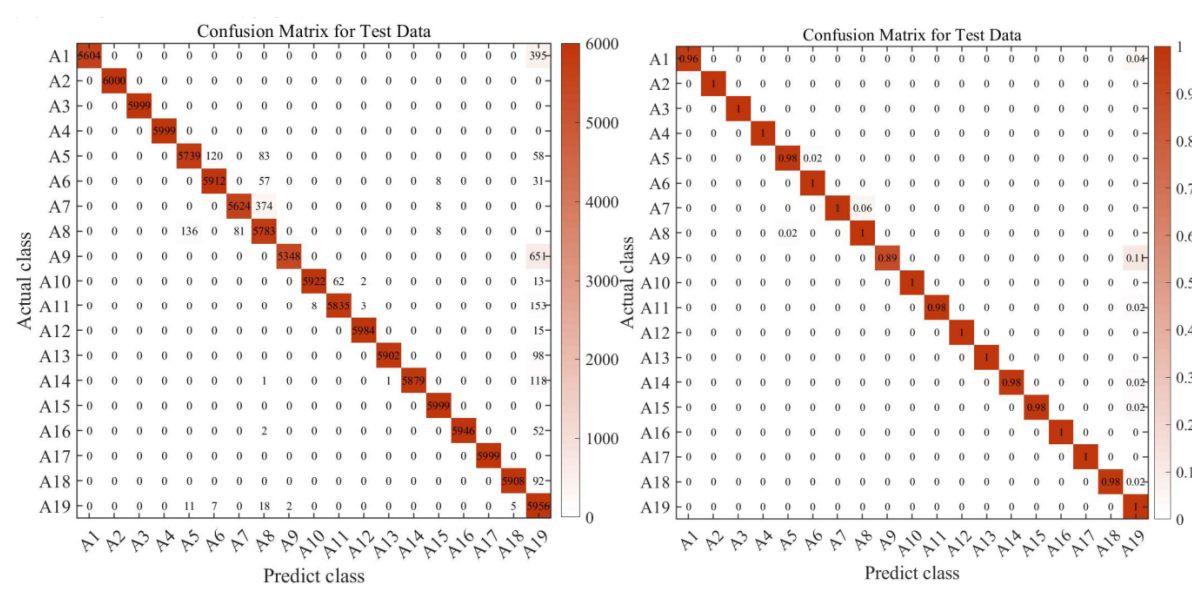


图 5.1 两层分类器模型 XR-KS 的混淆矩阵图

我们在四个数据集：UCI DSA、UCI HAR、WISDM 和 IM-WSHA 上，将我们的方法与其他方法进行了比较，如表 5.1 所示。

表 5.1 在 UCIDSA、UCIHAR、WISDM 和 IM-WSHA 数据集上，所提方法与其他最新方法的识别准确率对比（表中加粗部分表示本文的方法）

参考文献	模型	UCIDS A	UCIHA R	WISDM	IM- WSHA
Qiueta1. (2016) [4]	Estimationalgorithm[4]				80.49 %
Gochooeta1. (2021) [5]	RPLB[5]				83.18 %

参考文献	模型	UCIDS A	UCIHA R	WISDM	IM- WSHA
Halimetal. (2022) [6]	hybrid descriptors and random forest [6]				91.45 %
Ghadietal. (2022) [7]	MS-DLD [7]				95.0%
Koşaretal. (2023) [8]	DeepCNN-LSTM [8]		93.11 %		
Kobayashietal. (2023) [9]	MarNASNet [9]		94.50 %	88.92%	
Wangetal. (2023) [10]	DMEFAM [10]		96%	97.9%	
Dua, Netal. (2023) [11]	DeepCNN-GRU [11]		96.20 %	97.21%	
Imranetal. (2023) [12]	EdgeHARNet [12]			94.036 %	
Zhangetal. (2023) [13]	BLSTM [13]		98.37 %	99.01%	
Thakuretal. (2023) [14]	CAEL-HAR [14]		96.45 %	98.57%	
Li et al. (2016) [15]	MVTS [15]	91.35 %			
Present paper	XR-KS	97.69 %	97.92 %	98.12%	90.6%

5.2 泛化能力分析

通过研究[50]可知，可以使用 K 折交叉验证来测试模型的泛化能力。以下是验证模型的过程。首先，我们使用随机森林模型进行了 K 折交叉验证实验。在本研究中，我们将 K 设置为 5 来进行验证。实验结果如图 5.2 和表 5.2 所示，证明了我们模型的强大泛化能力。

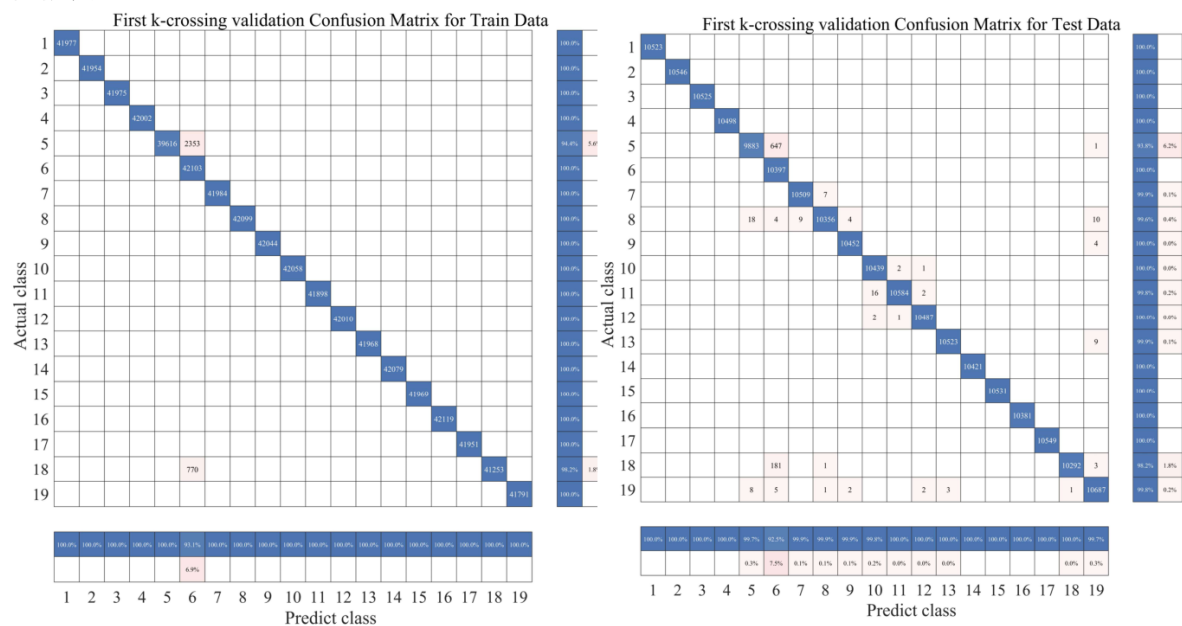


图 5.2 交叉验证及 k=5 的结果 (a) - (d)

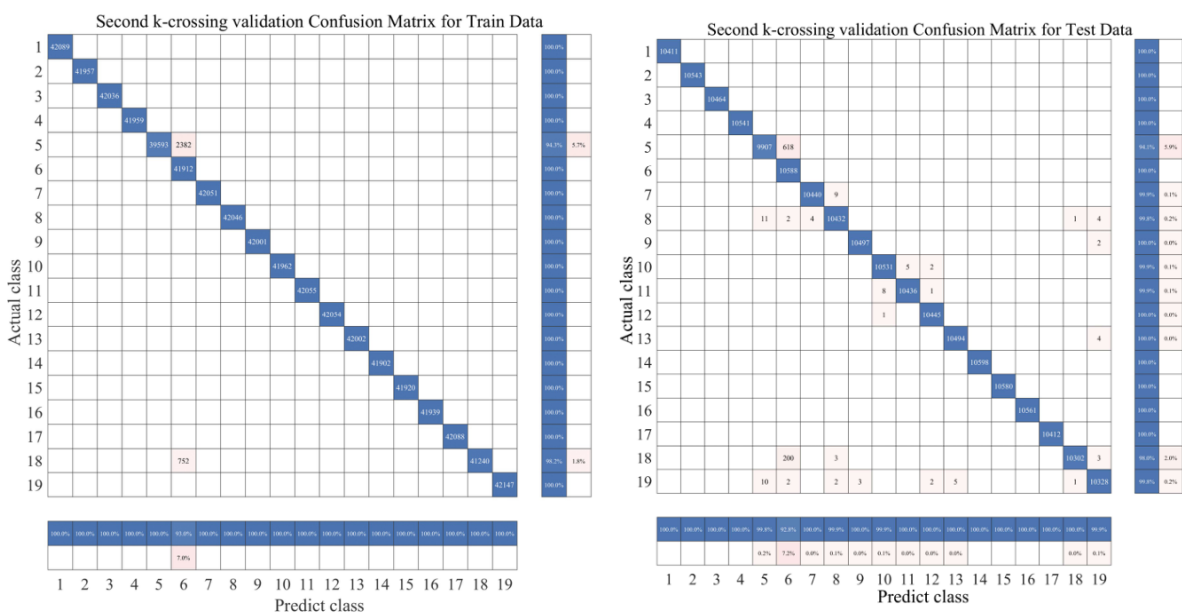


图 5.2 交叉验证及 k=5 的结果 (续一)

(c)

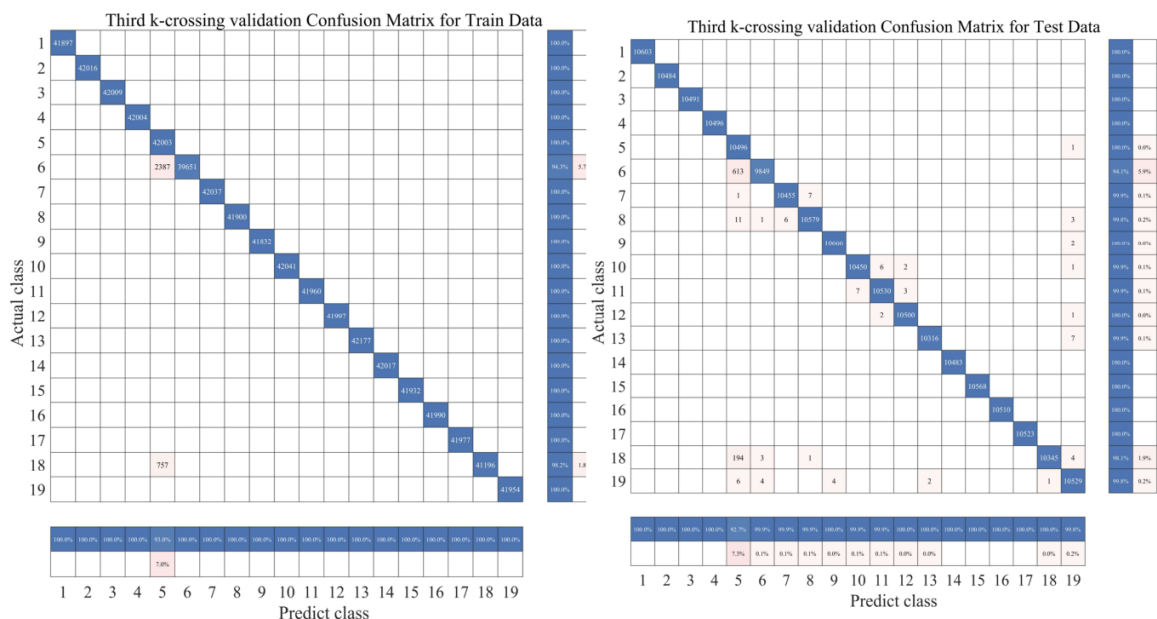


图 5.2 交叉验证及 k=5 的结果 (e) – (f)

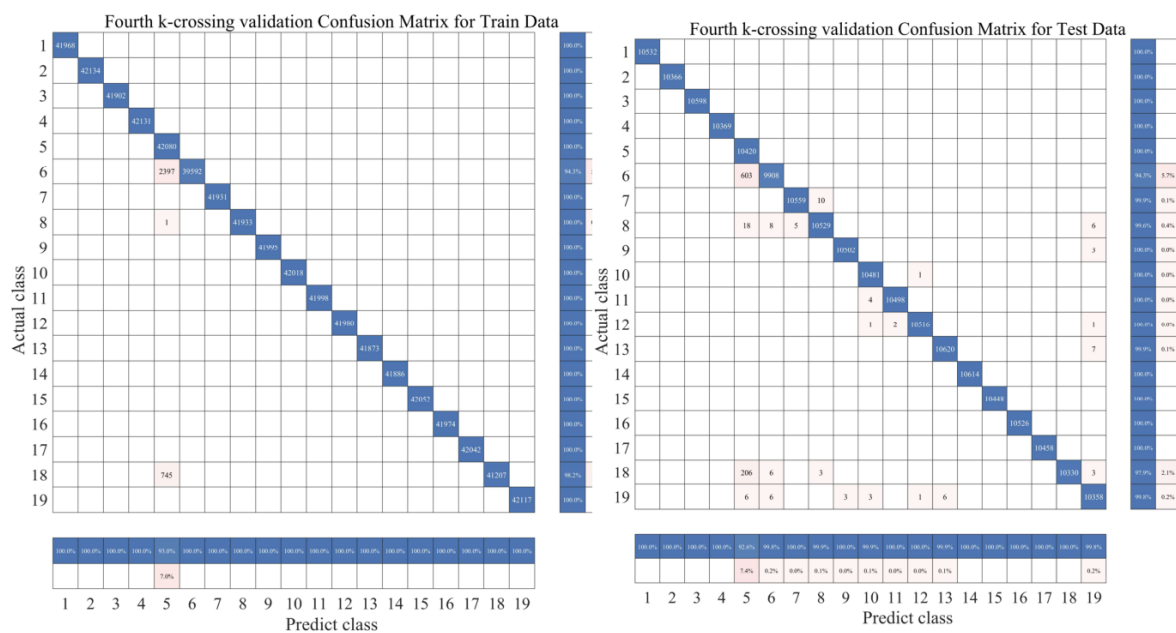


图 5.2 交叉验证及 k=5 的结果 (g) – (h)

(g)

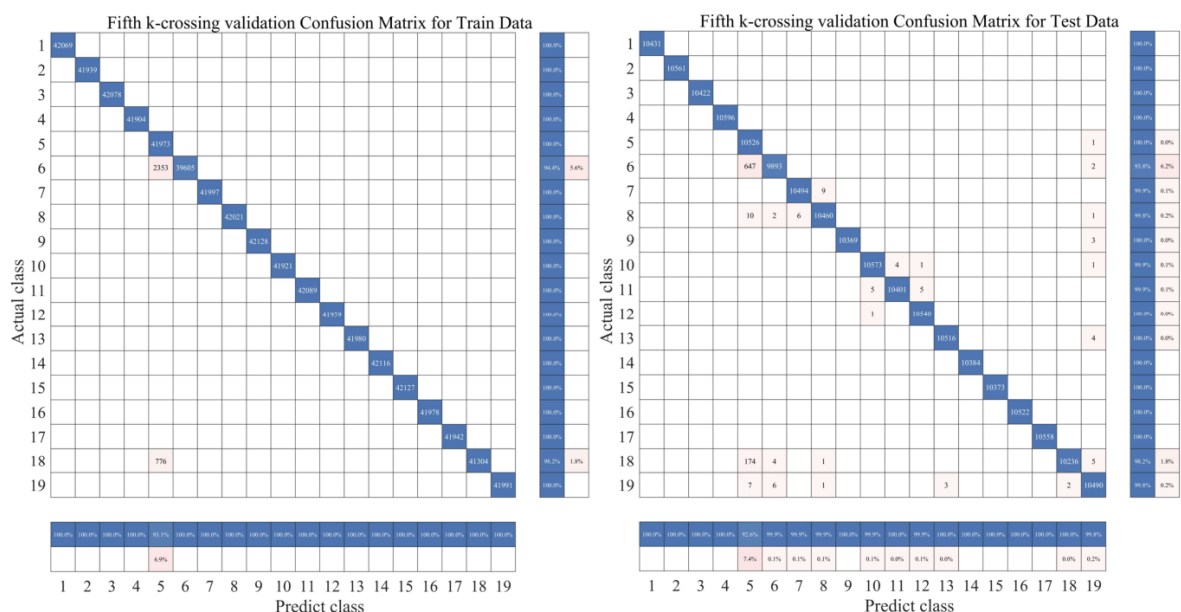


图 5.2 交叉验证及 k=5 的结果 (i) – (j)

表 5.2 k=5 和交叉验证的 UCIDSA 数据集的 XR 模型结果 (均值±标准差)

K	训练数据准确率	测试数据准确率	训练运行时间 (秒)	测试运行时间 (秒)
1	99.7414%(±0.91)	99.1421%(±0.94)	463.0862(±8.53)	51.0272(±1.81)
2	99.8561%(±0.84)	99.1421%(±0.45)	408.6202(±8.45)	48.6971(±2.31)
3	99.7651%(±0.57)	99.1421%(±0.41)	370.8408(±9.54)	42.7105(±2.45)
4	99.8234%(±0.61)	99.1421%(±0.61)	377.3214(±7.14)	40.2905(±2.16)
5	99.8317%(±0.74)	99.1421%(±0.37)	386.5293(±7.41)	39.6749(±1.97)

通过 Fawcett 对 ROC 曲线的研究,我们旨在通过分析 ROC 值来评估支持向量机 (SVM) 的有效性。模型在训练集与测试集比例为 9:1 时的结果如图 5.3 所示,而其他不同比例模型的结果可在表 5.3 中找到。上述结果表明,我们的模型训练良好。

表 5.3 UCIDSA 数据集中 KS 模型的 AUC 均值 (±标准差) 结果

比例(训练集: 测试集)	A9, A13 和 A18, A19AUC	A9, A13AUC	A18, A19AUC
9:1	0.98491(±0.01)	0.77412(±0.01)	0.97961(±0.02)
8:2	0.99155(±0.01)	0.90591(±0.01)	0.97573(±0.01)
7:3	0.99229(±0.01)	0.95818(±0.01)	0.97361(±0.04)

比例 (训练集: 测试集)	A9, A13 和 A18, A19AUC	A9, A13AUC	A18, A19AUC
6:4	0.99386 (± 0.03)	0.97513 (± 0.01)	0.96473 (± 0.03)
5:5	0.99153 (± 0.01)	0.98271 (± 0.02)	0.86586 (± 0.01)
4:6	0.99324 (± 0.01)	0.93727 (± 0.03)	0.87724 (± 0.01)

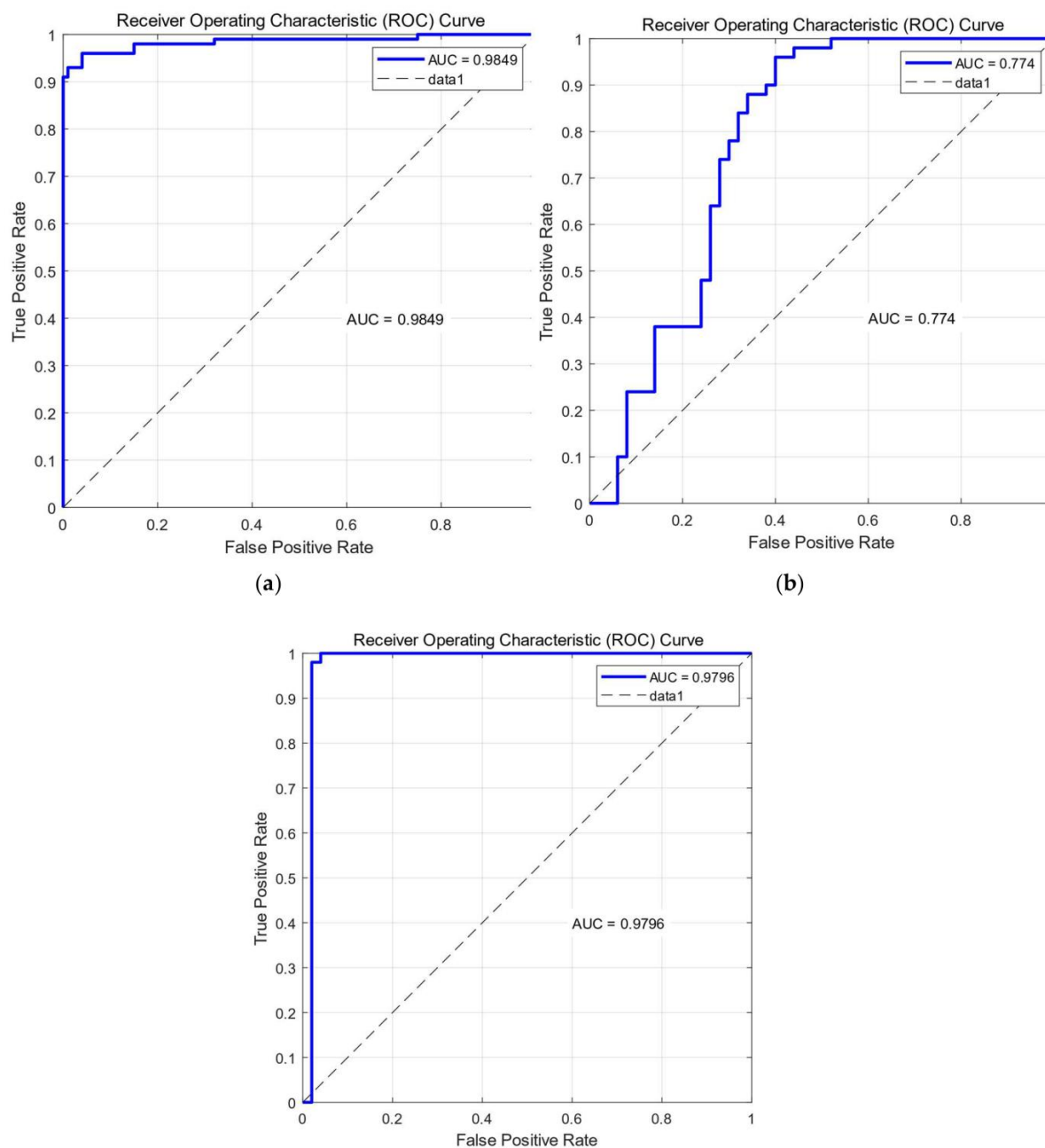


图 5.3 UCIDS 数据集中支持向量机 (SVM) 模型在训练集与测试集比例为 9:1 时的 ROC 曲线和 AUC 值

6. 模型评价

6.1 总结

本文提出了一种基于多层模型 XR-KS 的方法来识别相似人类活动。该方法首先利用 XGBoost 特征选择算法对全部特征进行筛选，选取特征子集用于随机森林分类，以提高整体分类效率。在此基础上，我们针对难以识别的数据点，进一步引入结合核 Fisher 判别分析的支持向量机（SVM）模型进行处理。

具体而言，第一步通过核 Fisher 判别分析提升相似活动的区分能力；第二步则利用 SVM 对数据进行分类，取得了令人满意的识别结果。随后，我们采用 K 折交叉验证与 ROC 曲线对模型性能进行了验证，结果表明所提模型具备较强的泛化能力，能够有效识别相似人类活动。

未来的研究方向主要包括以下三点： λ 将所提技术扩展至其他相似活动识别任务（如打字与书写），以及在特征有限或样本稀缺的复杂驾驶环境中进行数据分类； λ 发展能够自动区分相似活动的模型，实现更高层次的自动化； λ 从数据源头出发，设计和实施稳健的数据采集与预处理方案，例如在传感器采集过程中有效去噪，并专注于收集对活动识别贡献更大的特征数据。

6.2 模型评价

6.2.1 XR 模型（基于 XGBoost 与随机森林）

6.2.1.1 模型优点

通过 XGBoost 进行特征选择，减少冗余特征，提高随机森林分类的效率和稳定性；在高维特征空间中能够较好地避免过拟合，并保持较高的泛化性能。

6.2.1.2 模型缺点

对于部分边界模糊的活动类别（如 A9 和 A13），特征选择与随机森林仍难以给出稳定判别；依赖较大规模的样本进行训练，在小样本条件下性能可能下降。

6.3.1 KS 模型（基于核 Fisher 判别分析的 SVM）

6.3.1.1 模型优点

核 Fisher 判别分析有效增强了相似类的区分度，提升了 SVM 在边界模糊样本下的分类性能；SVM 适用于小样本高维数据，能够在相对有限的样本下保证较好的泛化能力；

6.3.1.2 模型缺点

对核函数和参数的选择敏感，不同参数设置下结果差异较大；计算复杂度较高，在大规

模数据集上可能存在训练耗时问题。

参考文献

- [1] Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature selection: A data perspective. *ACM Comput. Surv. (CSUR)* 2017, 50, 1 – 45.
- [2] Dong, S.; Zeng, L.; Du, X.; He, J.; Sun, F. Lithofacies identification in carbonate reservoirs by multiple kernel Fisher discriminant analysis using conventional well logs: A case study in A oilfield, Zagros Basin, Iraq. *J. Pet. Sci. Eng.* 2022, 210, 110081.
- [3] Schölkopf, B.; Smola, A.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* 1998, 10, 1299 – 1319.
- [4] Qiu, S.; Wang, Z.; Zhao, H.; Hu, H. Using distributed wearable sensors to measure and evaluate human lower limb motions. *IEEE Trans. Instrum. Meas.* 2016, 65, 939 – 950.
- [5] Gochoo, M.; Tahir, S.B.U.D.; Jalal, A.; Kim, K. Monitoring real-time personal locomotion behaviors over smart indoor-outdoor environments via body-worn sensors. *IEEE Access* 2021, 9, 70556 – 70570.
- [6] Halim, N. Stochastic recognition of human daily activities via hybrid descriptors and random forest using wearable sensors. *Array* 2022, 15, 100190.
- [7] Ghadi, Y.Y.; Javeed, M.; Alarfaj, M.; Shloul, T.A.; Alsuhibany, S.A.; Jalal, A.; Kamal, S.; Kim, D. MS-DLD: Multi-sensors based daily locomotion detection via kinematic-static energy and body-specific HMMs. *IEEE Access* 2022, 10, 23964 – 23979.
- [8] Koşar, E.; Barshan, B. A new CNN-LSTM architecture for activity recognition employing wearable motion sensor data: Enabling diverse feature extraction. *Eng. Appl. Artif. Intell.* 2023, 124, 106529.
- [9] Kobayashi, S.; Hasegawa, T.; Miyoshi, T.; Koshino, M. MarNASNets: Toward CNN Model Architectures Specific to Sensor-Based Human Activity Recognition. *IEEE Sens. J.* 2023, 23, 18708 – 18717.

- [10] Wang, Y.; Xu, H.; Liu, Y.; Wang, M.; Wang, Y.; Yang, Y.; Zhou, S.; Zeng, J.; Xu, J.; Li, S.; Li, J. A Novel Deep Multifeature Extraction Framework Based on Attention Mechanism Using Wearable Sensor Data for Human Activity Recognition. *IEEE Sens. J.* 2023, 23, 7188 – 7198.
- [11] Dua, N.; Singh, S.N.; Semwal, V.B. Multi-input CNN-GRU based human activity recognition using wearable sensors. *Computing* 2021, 103, 1461 – 1478. [12] Imran, H.A.; Ikram, A.A.; Wazir, S.; Hamza, K. EdgeHARNet: An Edge-Friendly Shallow Convolutional Neural Network for Recognizing Human Activities Using Embedded Inertial Sensors of Smart-Wearables. In *Proceedings of the 2023 International Conference on Communication, Computing and Digital Systems (C-CODE)*, Islamabad, Pakistan, 17 – 18 May 2023; pp. 1 – 6.
- [13] Zhang, J.; Liu, Y.; Yuan, H. Attention-Based Residual BiLSTM Networks for Human Activity Recognition. *IEEE Access* 2023, 11, 94173 – 94187.
- [14] Thakur, D.; Roy, S.; Biswas, S.; Ho, E.S.L.; Chattopadhyay, S.; Shetty, S. A Novel Smartphone-Based Human Activity Recognition Approach using Convolutional Autoencoder Long Short-Term Memory Network. In *Proceedings of the 2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, Bellevue, WA, USA, 4 – 6 August 2023; pp. 146 – 153.
- [15] Li, S.; Li, Y.; Yun, F. Multi-view time series classification: A discriminative bilinear projection approach. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management ACM*, New York, NY, USA, 24 October 2016; pp. 989 – 998.

附录

代码（由于代码较多，这里仅展示部分代码，其余代码可见支撑材料）

运行环境

Matlab

```
function SVMmain()
%% 读取 CSV 文件（兼容北太天元与 MATLAB）
filename = './data/data4.csv';
```

```

if exist('readmatrix','file') % MATLAB R2019a 及以后
    data = readmatrix(filename);
else
    data = csvread(filename);
end

x = data(:,1);
y = data(:,2);

%% A18 与 A19 分类组成训练集和测试集
% 训练集
group_1 = [x(1001:1450), y(1001:1450)];
label_1 = ones(size(group_1,1),1);

group_2 = [x(1501:1950), y(1501:1950)];
label_2 = -ones(size(group_2,1),1);

% 测试集
tg1 = [x(1451:1500), y(1451:1500)];
tlabel_1 = ones(size(tg1,1),1);

tg2 = [x(1951:2000), y(1951:2000)];
tlabel_2 = -ones(size(tg2,1),1);

% 合并训练集与测试集
X = [group_1; group_2];
Y = [label_1; label_2];

test_group = [tg1; tg2];

```

```

test_label = [tlabel_1; tlabel_2];

%% SVM 训练
tic;
C = 10;
svm = mysvmtrain(X, Y, C, 'linear');
result = mysvmtest(svm, test_group, 'linear');

fprintf('测试集准确率: %f\n', mean(double(result.y_pred == test_label)));

% 计算 w 和 b
tmp_b = (svm.alphas' .* svm.vec_y' * kernels(svm.vec_x',
svm.vec_x', 'linear'))';
total_bias = svm.vec_y - tmp_b;
true_b = mean(total_bias);

w = svm.vec_x' * (svm.alphas' .* svm.vec_y)';
disp(['运行时间: ', num2str(toc)]);

%% 绘制训练集与超平面
subplot(1,2,1);
plot(group_1(:,1), group_1(:,2), 'ro', 'MarkerFaceColor', 'r');
hold on;
plot(group_2(:,1), group_2(:,2), 'bo', 'MarkerFaceColor', 'b');

k = -w(1)/w(2);
bb = -true_b / w(2);
xx = min(X(:,1))-1 : 0.1 : max(X(:,1))+1;
yy = k*xx + bb;

```

```

plot(xx, yy, '-', 'LineWidth', 1.5);
plot(xx, yy + 1/w(2), '--', 'LineWidth', 1);
plot(xx, yy - 1/w(2), '--', 'LineWidth', 1);

title('Hard-SVM Training');
xlabel('Dimension1'); ylabel('Dimension2');
legend('A18 for training', 'A19 for training', 'Separating Hyperplane');
grid on;

%% 绘制测试集与超平面
subplot(1,2,2);
plot(tg1(:,1), tg1(:,2), 'ro', 'MarkerFaceColor', 'r');
hold on;
plot(tg2(:,1), tg2(:,2), 'bo', 'MarkerFaceColor', 'b');

plot(xx, yy, '-', 'LineWidth', 1.5);
plot(xx, yy + 1/w(2), '--', 'LineWidth', 1);
plot(xx, yy - 1/w(2), '--', 'LineWidth', 1);

title('Hard-SVM Testing');
xlabel('Dimension1'); ylabel('Dimension2');
legend('A18 for testing', 'A19 for testing', 'Separating Hyperplane');
grid on;

end

%% 核函数
function K = kernels(X1, X2, kernel_type)
switch lower(kernel_type)

```

```

case 'linear'
    K = X1' * X2;
case 'polynomial'
    K = (X1' * X2).^3;
case 'rbf'
    sigma = 0.345;
    K = zeros(size(X1,2), size(X2,2));
    for m = 1:size(X1,2)
        for n = 1:size(X2,2)
            K(m,n) = exp(-sum((X1(:,m)-X2(:,n)).^2)/sigma);
        end
    end
end
end

%% SVM 训练
function model = mysvmtrain(X, y, C, kernel_type)
m = length(y);
H = y*y' .* kernels(X', X', kernel_type);
f = -ones(m,1);
A = []; b = [];
Aeq = y'; beq = 0;
lb = zeros(m,1); ub = C*ones(m,1);
alphas0 = zeros(m,1);
epsilon = 1e-8;
options = optimset('LargeScale','off','Display','off');
alphas1 = quadprog(H, f, A, b, Aeq, beq, lb, ub, alphas0, options);

logic_vector = abs(alphas1) > epsilon;

```

```

model.vec_x = X(logic_vector,:);
model.vec_y = y(logic_vector);
model.alphas = alphas1(logic_vector);
end

%% SVM 测试
function result = mysvmtest(model, X, kernel_type)
tmp = (model.alphas' .* model.vec_y' * kernels(model.vec_x', model.vec_x',
kernel_type))';
bias = mean(model.vec_y - tmp);

wx = (model.alphas' .* model.vec_y' * kernels(model.vec_x', X', kernel_type))';
result.w = model.vec_x' * (model.alphas' .* model.vec_y')';
result.b = bias;
result.y_pred = sign(wx + bias);
result.product = wx + bias;
end

```

202503080316 应用北太天元的生鲜商超补货策略优化模型

作者：刘忆恒 刘卓越 李佳荷

指导老师：刘英

学校：西安理工大学

摘要：生鲜商超改善经营模式，提高商超收益尤为重要，本文基于题目所给数据信息，建立数学模型进行分析，从而制定合理的蔬菜类商品动态定价与补货决策。

针对问题一，整合销售流水、商品信息、批发价格和损耗率数据，构建完整的分析数据集，并通过品类平均损耗率填补缺失值，计算考虑损耗后的实际成本。采用季节性分解模型分析各品类销量时间序列特征，识别年度周期性和趋势变化规律，年度周期各品类没有呈现出非常稳定的增长或下降趋势，花叶类季节性特征最强，食用菌类销量稳定，花菜类、水生根茎类周末销量增幅最强。运用 Pearson 相关系数构建品类间关联矩阵，量化花叶类、花菜类等六大品类间的协同与替代关系。基于帕累托分析识别各品类头部单品分布规律，结合 Pearson 相关分析揭示损耗率与销量间的负相关关系，利用热力图可视化热销单品间的关联模式，系统揭示蔬菜商品的销售分布规律及内在关联机制。通过 Pearson 相关系数筛选出 $|r| > 0.6$ 的强相关单品组合，为商品组合优化和交叉销售提供数据支撑。

针对问题二，通过对数线性回归分析各品类价格弹性特征，量化价格与销量的关系，得到富有弹性与缺乏弹性的品类分类。采用指数平滑法进行需求预测，针对不同数据特征选择加法或乘法模型，预测未来一周各品类的日需求量。在此基础上构建利润最大化优化模型，以补货量和加成率为决策变量，运用 L-BFGS-B 算法[1]求解最优策略，并设置补货量非负和加成率合理范围的约束条件，优化 7 月 1 日至 7 月 7 日的自动定价策略与补货方案。

针对问题三，通过导入第二问得到的各品类价格弹性系数，区分富有弹性与缺乏弹性的商品类别，分别采用弹性定价和成本加成定价策略。基于弹性需求模型计算单品基准需求和预测销量，以预期利润最大化为目标函数，通过品类需求比例分配和单品利润排序进行单品选择，在满足单品总数 27-33 个和最小陈列量 2.5kg 约束条件下，优化 7 月 1 日的单品补货量与定价策略。

针对问题四，结合前三个问题的分析和建模过程，提出外部环境数据、实时运营数据、供应链数据三个商家需要采集数据的方面，通过利用和分析这些数据，建立相应的动态规划模型，作为补货和定价模型的补充，从而防止产品出现缺货或积压，保证利润。

关键词：Pearson 相关系数；弹性系数；L-BFGS-B 算法；帕累托分析

点 评：本文针对生鲜商超的补货与定价这一实际运营问题，综合运用了时间序列分析、相关性分析、弹性分析和优化建模等多种方法，形成了数据驱动决策的完整分析框架。问题一的数据探索性分析扎实，季节性分解有效揭示了各品类的销售规律，Pearson 相关系数和帕累托分析为品类管理和商品组合优化提供了量化依据。问题二的价格弹性分析区分了富有弹性和缺乏弹性的品类，为差异化定价策略奠定了基础，L-BFGS-B 算法的选用在处理有界约

束优化问题时较为高效。问题四的前瞻性建议体现了对模型实际应用场景的深入思考。建议可进一步考虑生鲜商品的损耗率随时间动态变化的特点，以及促销活动对需求的短期冲击效应。

应用北太天元的生鲜商超补货策略优化模型

摘要：生鲜商超改善经营模式，提高商超收益尤为重要，本文基于题目所给数据信息，建立数学模型进行分析，从而制定合理的蔬菜类商品动态定价与补货决策。

针对问题一，整合销售流水、商品信息、批发价格和损耗率数据，构建完整的分析数据集，并通过品类平均损耗率填补缺失值，计算考虑损耗后的实际成本。采用季节性分解模型分析各品类销量时间序列特征，识别年度周期性和趋势变化规律，年度周期各品类没有呈现出非常稳定的增长或下降趋势，花叶类季节性特征最强，食用菌类销量稳定，花菜类、水生根茎类周末销量增幅最强。运用 Pearson 相关系数构建品类间关联矩阵，量化花叶类、花菜类等六大品类间的协同与替代关系。基于帕累托分析识别各品类头部单品分布规律，结合 Pearson 相关分析揭示损耗率与销量间的负相关关系，利用热力图可视化热销单品间的关联模式，系统揭示蔬菜商品的销售分布规律及内在关联机制。

通过 Pearson 相关系数筛选出 $|r| > 0.6$ 的强相关单品组合，为商品组合优化和交叉销售提供数据支撑。

针对问题二，通过对数线性回归分析各品类价格弹性特征，量化价格与销量的关系，得到富有弹性与缺乏弹性的品类分类。采用指数平滑法进行需求预测，针对不同数据特征选择加法或乘法模型，预测未来一周各品类的日需求量。在此基础上构建利润最大化优化模型，以补货量和加成率为决策变量，运用 L - BFGS - B 算法 [1] 求解最优策略，并设置补货量非负和加成率合理范围的约束条件，优化 7 月 1 日至 7 月 7 日的自动定价策略（表 4）与补货方案（表 3）。

针对问题三，通过导入第二问得到的各品类价格弹性系数，区分富有弹性与缺乏弹性的商品类别，分别采用弹性定价和成本加成定价策略。基于弹性需求模型计算单品基准需求和预测销量，以预期利润最大化为目标函数，通过品类需求比例分配和单品利润排序进行单品选择，在满足单品总数 27-33 个和最小陈列量 2.5 kg 约束条件下，优化 7 月 1 日的单品补货量与定价策略（表 7）。

针对问题四，结合前三个问题的分析和建模过程，提出外部环境数据，实时运营数据，供应链数据三个商家需要采集数据的方面，通过利用和分析这些数据，建立相应的动态规划模型，作为补货和定价模型的补充，从而防止产品出现缺货或积压，保证利润。

关键词： Pearson 相关系数，弹性系数， L - BFGS - B 算法，帕累托分析

一、问题重述

在蔬菜保鲜期短，品质容易受时间影响，这意味着商超必须面对迅速变化的市场需求，在不确定的情况下及时做出补货和定价决策。不具备一定的专业知识及数据分析能力，是无法高精度预测各种蔬菜的销量和销售价格。

问题一：根据附件 1 和附件 2，按照周、月、年等不同的分析维度，分析蔬菜各品类及单品销售量的分布关系，以及其相互关系；

问题二：在对各蔬菜品类与单品销量分析的基础上，进一步分析决定各蔬菜品类市场需求的可能因素，从而确定不同的成本加成定价水平对市场需求的影响，并给出各蔬菜品类 2023 年 7 月 1-7 日的日补货总量和定价策略，使商超利益最大化；

问题三：在可售单品总数控制在 27-33 个，且各单品订购量满足最小陈列量 2.5 千克的前提下，尽量满足市场对各品类蔬菜商品的需求，同时根据 2023 年 6 月 24-30 日的可售单品，给出 7 月 1 日的单品补货量和定价策略。

问题四：结合以上问题的分析和建模过程，希望商家继续采集哪些相关数据，从而可以完善模型并更好地制定蔬菜商品的定价和补货决策。

二、问题分析

问题一：研究对象为蔬菜各单品和品类的销售量，而附件 2 仅提供了三年内各单品蔬菜的销售流水数据，因此要先对该数据进行预处理。首先需要观察分析原始数据，检查是否有缺失值、异常值等，并进行相应的处理。其次，通过单品编码关联四个附件数据，对数据进行加总，分别获得各单品和各品类蔬菜的日销量、周销量等数据之后，才能对蔬菜各单品、品类的销售分布和相互关系进行分析。

针对不同品类蔬菜的总体分布规律及相互关系，从品类和单品两个层面展开分析。

首先从品类层面通过时间序列分解和统计指标分析销售特征，并计算品类间相关性。单品层面运用帕累托法则识别核心商品，分析损耗率与销量的关系，并挖掘热销单品间的关联性。分析采用统计方法与可视化技术，系统揭示蔬菜销售规律，为经营决策提供支持。

问题二：针对问题二的品类补货与定价决策，核心在于分析各蔬菜品类的价格弹性特征，建立需求预测模型，并在考虑损耗成本的基础上，通过优化算法确定未来一周各品类的最优日补货量和定价策略。需要基于历史销售数据量化价格与销量的关系，预测需求趋势，最终以商超收益最大化为目标，制定差异化的补货和定价方案。

问题三：问题三需要在有限销售空间约束下，以单品为单位制定补货计划。在可售单品总数控制在 27-33 个，各单品订购量满足最小陈列量 2.5 千克，基于 2023 年 6 月 24-30 日的销售数据确定 7 月 1 日的补货策略，在满足市场需求的前提下最大化商超收益，建立两阶段优化模型：首先进行单品层面的参数估计和初步筛选，然后进行品类层面的资源分配和最终决策。

问题四：应当结合前三个问题的分析和建模过程。结合问题一对销量分布规律和相互关系的分析，可以联想找出影响销量的潜在因素，从而更好地分析销量的规律；结合问题二和问题三，可以寻找出定价和补货模型中缺少的数据和信息。此外，可以从之前模型中利用不足的数据切入，更好地构造模型。

三、模型假设

3.1 模型基本假设

- (1) 同一品类内不同单品的损耗率特征相似，缺失值可以用品类均值合理替代；
- (2) 商超的客群基本稳定，消费者行为在短期内可以预测；
- (3) 最近一周的销售数据能够代表未来需求模式，忽略更长期的历史趋势和季节性；
- (4) 需求完全由价格决定，其他因素（如天气、竞争、促销）的影响可忽略或被价格变化涵盖。

四、符号说明

4.1 符号说明

符号	含义
CV	变异系数
Y_t	品类日销量序列
T_t	2020-2023 年长期销量变化趋势
S_t	提取年度内月度、季度周期性波动
R_t	随机波动
m	月份
c	品类
n	对应月份的销售天数
$S_{m, c, i}$	第 i 天销量

符号	含义
S_{ce}	品类 c 周末平均销量
S_{cw}	品类 c 工作日平均销量
Q	销量
P	销售价格
a	基准需求常数
ε	价格弹性系数
W	批发价格
$\pi_{c,d}$	第 c 个品类在第 d 天的利润
$I_{c,d}$	第 d 天结束时第 c 个品类的库存量
$\eta_{\min}$	最低需求满足率阈值
L_c	第 c 个品类的损耗率
c_i	第 i 个品类的实际成本

五、模型建立与求解

5.1 问题一模型建立与求解

5.1.1 问题一求解思路

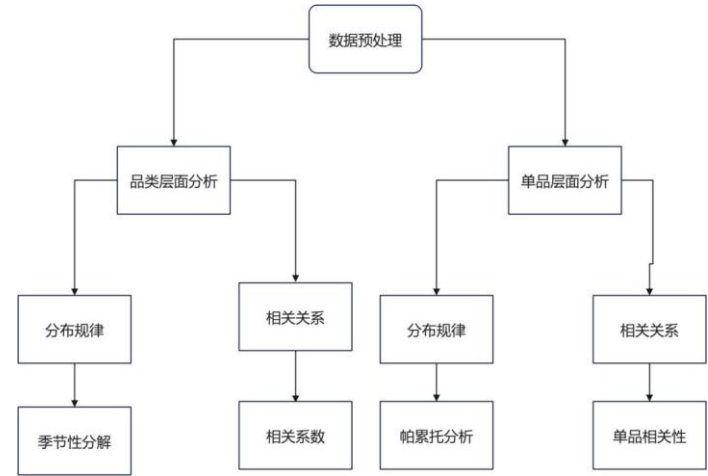


图 1 问题一求解流程图

问题一求解思路如图 1 所示，首先进行数据预处理：通过单品编码关联四个附件数据，整合销售流水、批发价格和损耗率信息，标准化日期格式并清洗数据（包括处理缺失值、剔除异常记录）；接着计算考虑损耗后的实际成本，并添加时间特征。在品类层面采用时间序列分解模型分析季节性规律，结合变异系数等统计量量化销售波动性，并通过 Pearson 相关

系数矩阵揭示品类间关联性；在单品层面运用帕累托分析识别核心商品，分析损耗率与销量的相关性，并挖掘热销单品间的强相关组合（ $|r| > 0.5$ ）。最终通过可视化图表和统计报告呈现分布规律与关联关系，为后续决策提供依据。

5.1.2 问题一模型建立

（一）数据预处理

以“单品编码”作为核心关联字段，整合附件 1-4 中的数据，提取关键核心字段，包括单品编码、分类名称、销售日期、销量（千克）、批发价格、损耗率（%）。借助“单品编码”这一唯一标识，将分散在不同附件中的数据进行关联融合，构建起完整且连贯的蔬菜销售数据集，确保后续分析能基于全面、统一的数据开展。接着进行数据清洗：

（1）日期格式标准化

将销售日期数据统一转换为 `datetime` 格式。规范的日期格式便于后续提取时间特征（如月份、季度、星期等），也能避免因日期格式不统一可能产生的分析误差，为时间序列分析、周期性规律探究提供清晰且一致的时间维度基础。

（2）缺失值处理

针对损耗率存在缺失的数据，采用品类平均值填充的策略。即对于某一品类下损耗率缺失的单品，计算该品类所有其他单品损耗率的平均值，并用此平均值填补缺失值。

这一方法在一定程度上能保持该品类数据的内在一致性，减少缺失值对后续分析（如实际成本计算、波动分析等）的影响，同时利用品类内数据的关联性，使填充结果更贴合实际业务逻辑。

（3）异常值处理

- 1）剔除销量为 0 但价格正常的记录。此类记录无法反映真实销售情况，且可能干扰后续基于销量的分析（如销售规律挖掘、关联关系探究等），保留会导致数据偏离实际业务场景，故予以剔除。
- 2）剔除销售类型为退货的记录。退货记录与正常销售记录的业务逻辑不同，会使销量等数据不能真实反映销售需求与趋势，为保证分析聚焦于有效销售行为，需将其剔除。
- 3）剔除附件 4 中的隐藏 sheet。隐藏 sheet 通常不承载常规分析所需的有效数据，可能包含错误、冗余或无关信息，剔除后可使数据集中在有价值的内容上，提升数据质量。

在清洗后的数据基础上，通过特征工程挖掘业务场景所需的分析维度：

(1) 实际成本计算

$$\text{实际成本} = \frac{\text{批发价格}}{1 - \text{损耗率}/100} \quad (1)$$

计算各单品的实际成本。在蔬菜销售中，损耗是客观存在的，该公式考虑了损耗对成本的影响，通过扣除损耗占比，得到更贴近真实成本的数值，使成本核算更符合实际业务中因损耗导致成本增加的情况。

$$CV = \frac{\sigma}{\mu} \times 100\% \quad (2)$$

其中 σ 为标准差，反映数据的离散程度； μ 为均值，代表数据的集中趋势。

例如，若花叶类的 CV 值为 0.65，而食用菌类的 CV 值为 0.35，则说明花叶类销售稳定性较差，在库存管理、补货策略制定等方面，需更精细地关注其销量波动，以应对可能的库存积压或缺货问题；食用菌类相对稳定，管理策略可相对简化，此指标为差异化库存与销售策略制定提供数据依据。

(3) 时间特征添加

1) 月份（1-12 月）：提取销售日期对应的月份信息，不同月份蔬菜的市场供需、消费习惯等可能存在差异（如夏季蔬菜品种丰富、销量高，冬季部分蔬菜依赖储存或反季种植，销量与品类结构有别），月份特征有助于分析蔬菜销售的季节性规律。

2) 季度（1-4 季度）：将月份对应到季度（1-3 月为第一季度，4-6 月为第二季度，7-9 月为第三季度，10-12 月为第四季度），季度特征可用于探究更宏观的周期性销售趋势，不同季度的气候、节日分布等因素会综合影响蔬菜销售，通过季度分析能把握较长周期内的销售规律。

3) 星期（0-6 对应周一到周日）：提取销售日期对应的星期信息，一周内不同工作日的消费需求存在差异（如周末家庭采购量通常大于工作日），星期特征可用于分析周度销售波动规律，为周内补货、人员安排等提供依据。

4) 是否周末（周六、周日标记为 True，其余为 False）：单独标记周末，突出周末这一特殊时间节点对销售的影响，便于直接分析周末与非周末销售数据的差异，快速识别周末效应在蔬菜销售中的体现（如销量提升、品类需求变化等）。

(二) 品类层面分析

Step 1: 考虑到蔬菜销售具有明显的趋势性和季节性特征，采用时间序列分析将历史

$$Y_t = T_t \times S_t \times R_t \quad (3)$$

：趋势项，反映 2020–2023 年长期销量变化趋势，通过移动平均法拟合；

：季节项，提取年度内月度、季度周期性波动（如夏季叶菜需求高峰），采用傅里叶变换或滑动平均分离周期成分；

：残差项，代表从原始销量中剔除趋势、周期后的随机波动（如极端天气、临时促销的影响）。

接着筛选销量前 4 品类，对其 2020–2023 年日销量序列执行季节性分解，绘制 “原始销量 – 趋势 – 季节 – 残差” 四维度分解图，直观呈现时间特征。

$$\dot{S}_{m,c} = \frac{1}{n} \sum_{i=1}^n S_{m,c,i} \quad (4)$$

c：品类 n：对应月份的销售天数 $S_{m,c,i}$ ：第 i 天销量采用堆叠柱状图展示 6 大品类月度销量分布，横轴为月份（1–12），纵轴为平均销量，不同颜色堆叠区代表各品类贡献，呈现旺季、淡季特征。

$$G_c = \frac{\dot{S}_{ce} - \dot{S}_{cw}}{\dot{S}_{cw}} \times 100\% \quad (5)$$

其中， $\dot{S}_{ce}$ ：品类 c 周末平均销量；

$\dot{S}_{cw}$ ：品类 c 工作日平均销量。

接着用雷达图展示品类周末增幅，雷达轴对应各品类，半径表示增幅百分比，对比周内波动差异。

$$\mu_c = \frac{1}{N} \sum_{i=1}^N S_{c,i} \quad (6)$$

$$\sigma_c = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (S_{c,i} - \mu_c)^2} \quad (7)$$

$$CV_c = \frac{\sigma_c}{\mu_c} \quad (8)$$

其中，：品类 总销售天数；

，：第 i 天销量。

变异系数反映销量稳定性，CV 越大，波动越显著。

Step 5：计算 6 大品类日销量序列的 Pearson 相关系数，衡量线性相关性：

$$r_{c1,c2} = \frac{\sum_{t=1}^T (S_{c1,t} - \mu_{c1})(S_{c2,t} - \mu_{c2})}{\sqrt{\sum_{t=1}^T (S_{c1,t} - \mu_{c1})^2} \sqrt{\sum_{t=1}^T (S_{c2,t} - \mu_{c2})^2}} \quad (9)$$

其中， T ：总观测天数；

$S_{c1,t}$ ， $S_{c2,t}$ ：品类 $c1$ ， $c2$ 第 t 天销量。

判定： > 0.7 为强正相关； < -0.5 为强负相关。

基于 Pearson 系数构建 6×6 相关系数矩阵，通过热力图可视化品类间相关性强弱。

颜色深度映射相关系数大小，红色强正相关、蓝色强负相关，快速识别品类间关联模式。

遍历相关系数矩阵，筛选强正相关（ >0.7 ）和强负相关（ <-0.5 ）组合，输出“品类对 - 相关系数”列表，明确互补 / 替代关系。

（三）单品层面分析

Step 1：. 帕累托法则应用模型

$$P = \frac{\sum_{i=1}^k S_i}{\sum_{j=1}^n S_j} \times 100\% \quad (10)$$

：总单品数；

S_i ：对应单品销量。

Step 2：. 头部单品占比可视化

$$R_i = \frac{S_i}{\sum_{j=1}^5 S_j} \times 100\% \quad (11)$$

绘制饼图展示占比，凸显核心单品与长尾单品的销售结构差异。

Step 3：损耗率分析收集单品损耗率和销量（千克），以蔬菜的品类（如花叶类、食用菌类等 7 类）分类维度，用于给散点标记颜色区分，销量大小映射点的大小。

损耗 - 销量关联：构建散点图，横轴为损耗率 L ，纵轴为销量 S ，分析二者关联趋势。

Step 4：相关关系分析选取销量 TOP 30 单品，计算日销量序列的 Pearson 相关系数，重点关 > 0.5 的强相关组合。构建 TOP 30 单品相关系数矩阵，通过热力图仅展示 > 0.5 的强相关关系。

基于相关系数符号与业务逻辑，判定关系类型，区分互补（正相关）、替代（负相关）关系。

5.1.3 问题一模型求解与分析

通过对蔬菜销售数据的多维度分析，我们得到以下关键发现：

（一）时间维度特征

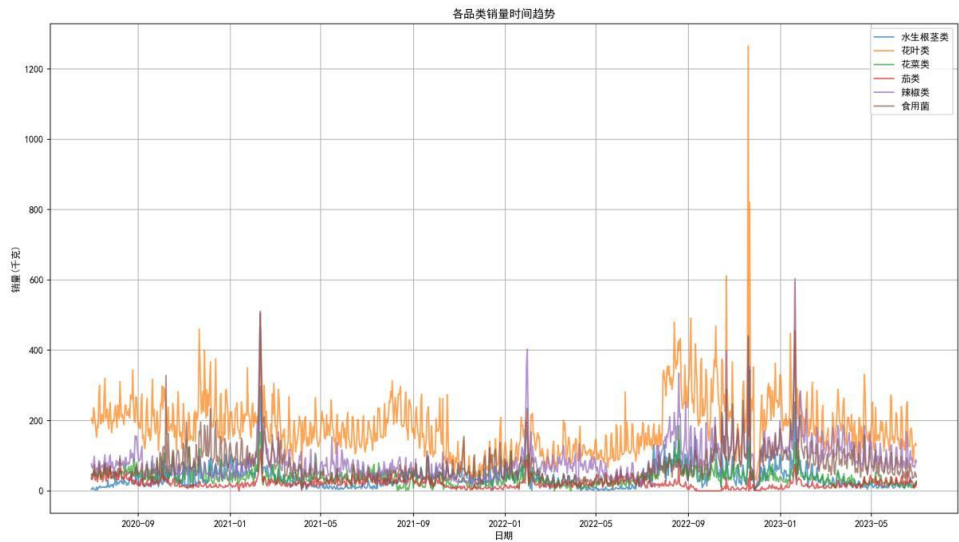


图 2 各品类销量时间趋势

花叶类起伏较大，说明该品类在不同时间点的销量变化明显，没有呈现出非常稳定的增长或下降趋势，在 2022-09 之后以及 2023-01 左右出现了大幅冲高的情况；辣椒类和食用菌类销量波动较为频繁，没有固定周期的销量高峰；茄类和水生根茎类销量在整个时间段内变化幅度不大，整体销量较为平稳，说明该品类的市场需求较为稳定，没有明显的季节性或周期性销量变化

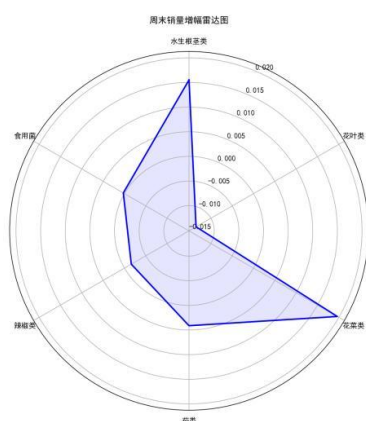


图 3 周末销量增幅雷达图

从周末销量增幅雷达图看，花菜类与水生根茎类增幅较高，说明周末对该品类销量拉动显著，可能因周末休闲、聚会场景多，消费需求旺；花叶类增幅为负，即周末销量较平日可能下降；食用菌类，茄类和辣椒类增幅接近 0，周末对其销量影响小，消费需求相对稳定，不受周末节奏明显带动。

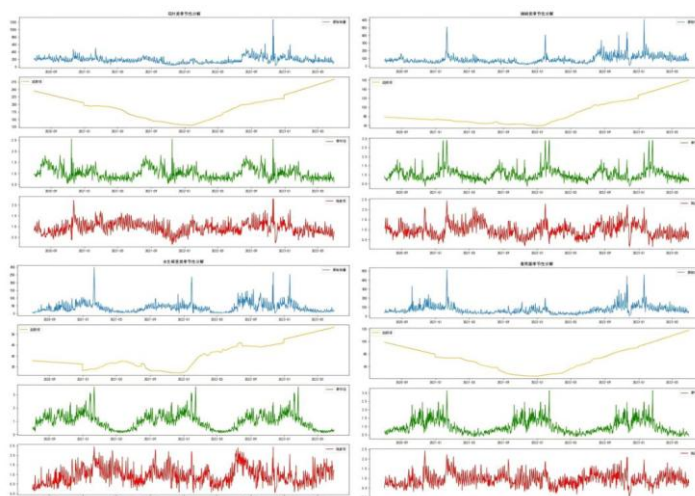


图 4 各品类季节性分解

销量季节性规律：花叶类蔬菜占据总销量 42% 的绝对主导地位，季节因子峰谷差超过 1.8，且在夏季呈现显著需求高峰，可见夏季需求较高；辣椒类在冬末春初有小高峰，其他时间波动较小，随季节变化需求有改变。其他品类多数时间销量平稳、波动小，季节性特征不突出，可能是日常需求相对稳定的品类，受季节影响小，不过 2023-01 左右也有异动，或许和整体消费环境（如年末采购）关联。

销售稳定性分析：花叶类虽然销量最高，但变异系数达 0.65，销售波动性显著；食用菌类损耗率最低（9.5%）且销售稳定，变异系数仅为 0.28；辣椒类单品多样性最丰富（超过 50 个品种），市场需求分散。

（二）品类层面特征

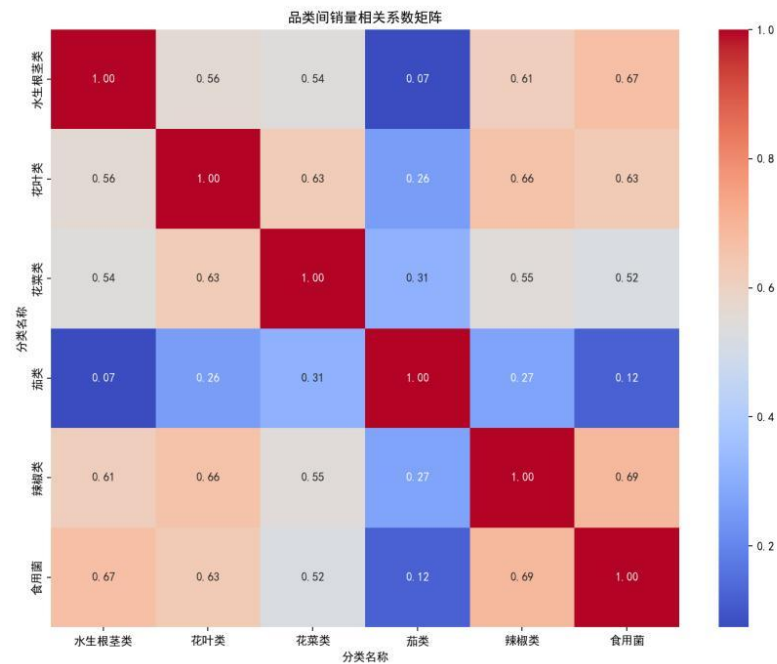


图 5 品类间销量相关系数矩阵

品类关联性：相关性分析显示，除茄类外，其他五大品类间均存在显著正相关关系（ $r > 0.6$ ），其中花叶类与食用菌类的协同效应最强（ $r = 0.69$ ）。而水生蔬菜类与茄类相关性弱（ $r = 0.07$ ），可能是因为二者消费场景、需求逻辑差异大。

水生蔬菜类（如莲藕、茭白）多作为特色食材，用于特定菜品（荷塘小炒等），消费场景聚焦特色餐饮、家常创意搭配；茄类（茄子等）是日常高频基础蔬菜，用于红烧茄子、鱼香茄子等家常菜，需求更偏向大众日常刚需。因应用场景、消费频次和目标人群重叠少，销量联动弱，采购、营销可独立规划，无需强绑定。为提高销售额，可设计包含强相关品类的组合套餐进行销售。比如推出“健康食材组合包”，将花叶类蔬菜与食用菌搭配在一起，以略低于单品售价总和的价格出售，既能满足消费者多样化的需求，又能通过组合优惠刺激购买欲望，增加销售额。

（三）单品分布规律

帕累托分布验证：

计算各品类单品销量分布，建立柱状图，并将各单品按占比从高至低排列，得到图

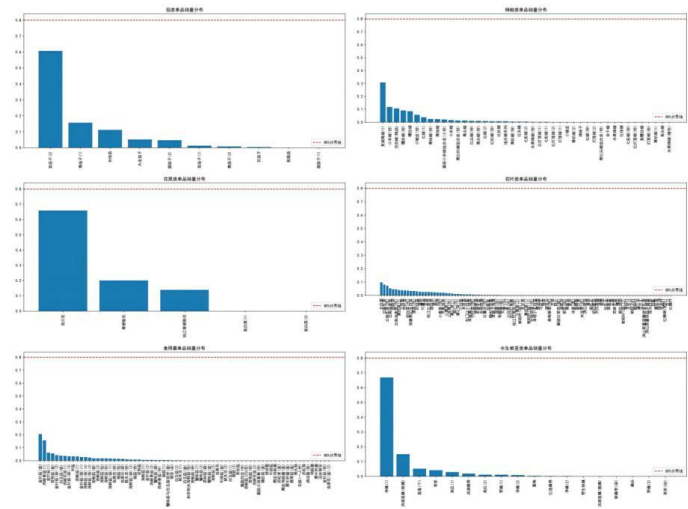


图 6 各品类单品销量分布

由图 6 可知，茄类单品前 5 名依次为紫茄子，青茄子，长线茄子，大龙茄子，圆茄子，计算销量占比 R_i 并绘制饼图展示占比，其他品类前 5 名单品按照同样的步骤绘制饼图。

图

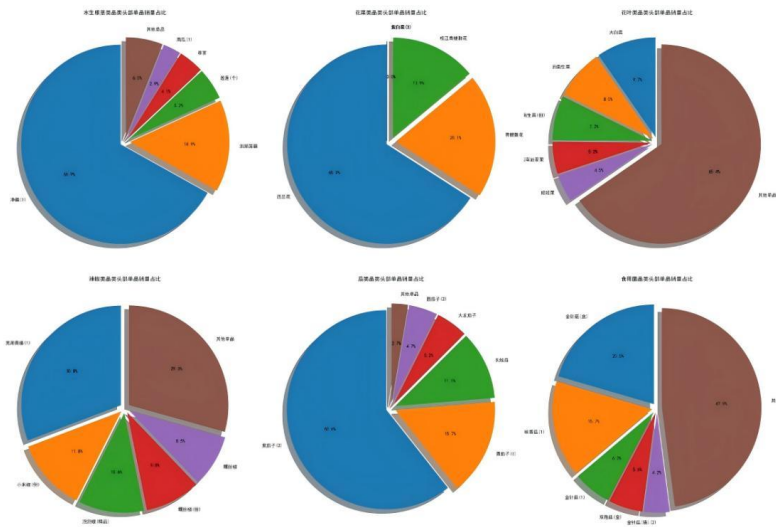


图 7 各品类头部单品销量占比饼图

由图 7 可知，所有品类均严格符合 ” 二八定律 ” ，头部 20% 的单品贡献了 75-85% 的销量，其中花叶类的集中度最高，前 5 名单品占比达 68% 。

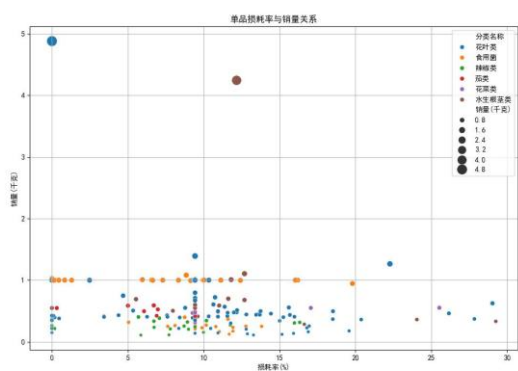


图 8 单品损耗率与销量关系

单品损耗率与销量关系：高损耗单品（损耗率 >15%）主要集中在花叶类，且销量普遍较低；食用菌类销量与损耗关系不大，说明该品类对损耗容忍度相对高，或损耗影响小；而水生根茎类平均损耗率均低于 15% 且销量较高。

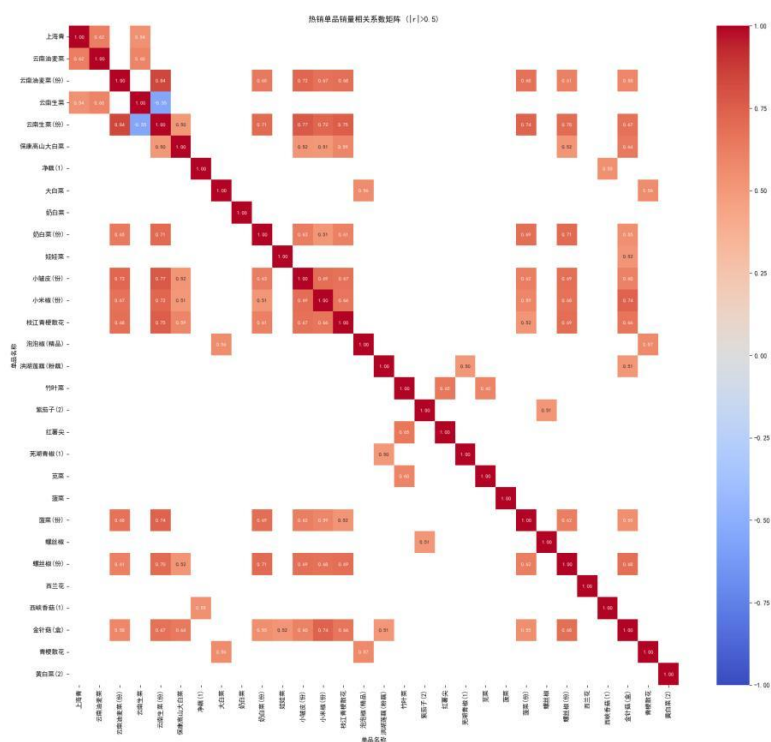


图 9 热销单品销量相关系数矩阵

热销单品相关性：TOP 30 热销单品中存在多个强相关组合（ $|r| > 0.6$ ），如云南生菜与小皱皮辣椒（ $r = 0.77$ ）、小米椒与金针菇（ $r = 0.74$ ），表明消费者往往同时购买这些互补商品。

5.2 问题二模型建立与求解

5.2.1 问题二求解思路

问题二在问题一对各蔬菜品类与单品销量分析的基础上，需要进一步分析决定各蔬菜品类市场需求的可能因素，从而确定不同的成本加成定价水平对市场需求的影

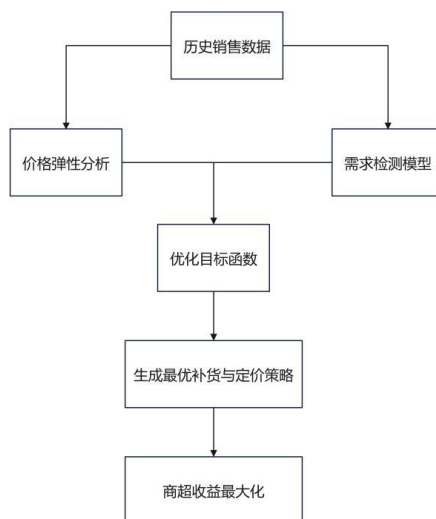


图 10 问题二求解流程图

由流程图（图 10）可知，问题二要基于历史销售数据，通过价格弹性分析明确各品类价格与销量的关系，同时利用需求预测模型得到未来需求情况，将两者结合构建优化目标函数，进而生成最优补货与定价策略，最终实现商超收益最大化。

5.2.2 问题二模型建立

（一）价格弹性分析

为深入分析各蔬菜品类销售总量与成本加成定价的内在关系，构建基于对数线性回归的价格弹性分析模型。

$$\hat{Q} = a \cdot P^{-\varepsilon} \quad (12)$$

P：销售价格（元 / 千克）

a：基准需求常数

ε：价格弹性系数

为估计模型参数，对上述需求函数两边取对数，转化为线性回归形式：

$$\ln(\hat{Q}) = \ln(a) - \varepsilon \ln(P) \quad (13)$$

过最小二乘法进行参数估计，得到 $\ln(a)$ 和 ε 的估计值，进而通过指数运算得到基准需求

常数

准需求常数 α 。

价格弹性系数 ε 是衡量需求对价格变化敏感程度的关键指标：

当 $\varepsilon > 1$ 时，表明需求富有弹性，价格变动对销量影响显著，适宜采用薄利多销策略；

当 $\varepsilon < 1$ 时，表明需求缺乏弹性，价格变动对销量影响较小，可采用较高加成率策略；

当 $\varepsilon = 1$ 时，为单位弹性，此时可实现总收入最大化。

（二）需求预测

为准确预测未来一周各蔬菜品类的日需求量，构建基于时间序列分析的预测模型，为补货决策提供数据支撑。

考虑到蔬菜销售具有明显的趋势性和季节性特征 [3]，采用时间序列分析将历史销量

$$D_t = T_t \times S_t \times R_t \quad (14)$$

其中， D_t ：观测值，品类日销量序列；

T_t ：趋势成分，反映销量长期的增长或下降方向；

S_t ：季节成分，捕捉周期性的周度或月度波动规律；

R_t ：残余成分，表示无法由趋势和季节解释的随机波动。

基于分解后的时间序列，采用指数平滑法进行未来一周的基准需求预测，基本形式

$$\hat{D}_{t+1} = \alpha D_t + (1 - \alpha) \hat{D}_t \quad (15)$$

在模型选择过程中，当历史数据中存在零或负值时，采用加法模型，避免乘法模型的数值计算问题；当历史数据均为正值时，采用乘法模型，更好地捕捉相对变化规律，有效利用历史数据中的趋势和季节信息。

（三）利润最大化优化模型

$$\pi = S \times P - Q \times W \quad (16)$$

S ：实际销量；

P ：销售定价；

Q ：补货量；

W : 批发价格;

π : 利润。

考虑实际销售过程中的库存动态变化:

$$I_{c,d} = I_{c,d-1} + Q_{c,d} - S_{c,d} \quad (17)$$

$$\sum_{c=1}^6 Q_{c,d} \leq Q_{\max} \forall d \quad (18)$$

其中, $Q_{\max}$ 为最大可陈列量。

$$\frac{\min(D_{c,d}(P_{c,d}), Q_{c,d}(1 - L_c))}{D_{c,d}(P_{c,d})} \geq \eta_{\min} \forall c, d \quad (19)$$

其中, $\eta_{\min}$ 为最低需求满足率阈值。

(12) ~ (19) 几个公式联立可得以商超收益最大化为核心目标, 建立多品类多周期的补货与定价联合优化模型, 目标函数为:

$$\max \sum_{d=1}^7 \sum_{c=1}^6 \pi_{c,d} \quad (20)$$

$$\pi_{c,d} = P_{c,d} \cdot \min(D_{c,d}(P_{c,d}), Q_{c,d}(1 - L_c)) - Q_{c,d}W_{c,d} \quad (21)$$

$-D_{c,d}(P_{c,d})$: 价格相关的需求函数, 弹性分析结果;

$Q_{c,d}$: 决策变量, 表示补货量;

$P_{c,d}$: 决策变量, 表示销售定价;

$W_{c,d}$: 批发成本;

L_c : 品类损耗率;

约束条件:

(1) 非负约束: $Q_{c,d} \geq 0, \forall c \in [1,6], d \in [1,7]$

$Q_{c,d} \leq Q_{\max} \forall c, d$

(2) 价格约束: $P_{c,d} \geq \frac{W_{c,d}}{1-L_c}, \forall c, d$

$P_{c,d} \leq \frac{W_{c,d}}{1-L_c} + \frac{Q_{c,d}}{D_{c,d}(P_{c,d})}$

(3) 加成率约束: $0.1 \leq \frac{P_{c,d}-W_{c,d}/(1-L_c)}{W_{c,d}/(1-L_c)} \leq 0.8, \forall c, d$

$\leq 0.8 \forall c, d$ 针对上述建立的约束非线性优化问题，本文选用 L - BFGS - B 算法进行求解，该算法的选择基于以下考虑：

(1) 边界约束处理能力： L - BFGS - B 算法专门设计用于处理决策变量的边界约束，能够

$W_{c,d} \geq 1-L_c$ $W_{c,d} \geq 1-L_c P_{c,d}$ -有效保证补货量的非负性（ $Q_{c,d} \geq 0$ ）和加成率的合理范围（ $0.1 \leq \leq 0.8$ ），完全契合本模型的约束条件要求。

(2) 计算效率优势：对于 6 个品类 $\times 7$ 天 =42 个优化变量的中等规模问题， L - BFGS - B 算

法通过拟牛顿法近似 Hessian 矩阵，避免了直接计算二阶导数的复杂性，在保证求解精度的同时显著提高了计算效率。

(3) 数值稳定性：该算法对于光滑非线性优化问题表现出良好的数值稳定性，能够有

效处理利润函数中 min 函数带来的非光滑特性，确保优化过程的收敛性。

5.2.3 问题二模型求解与分析

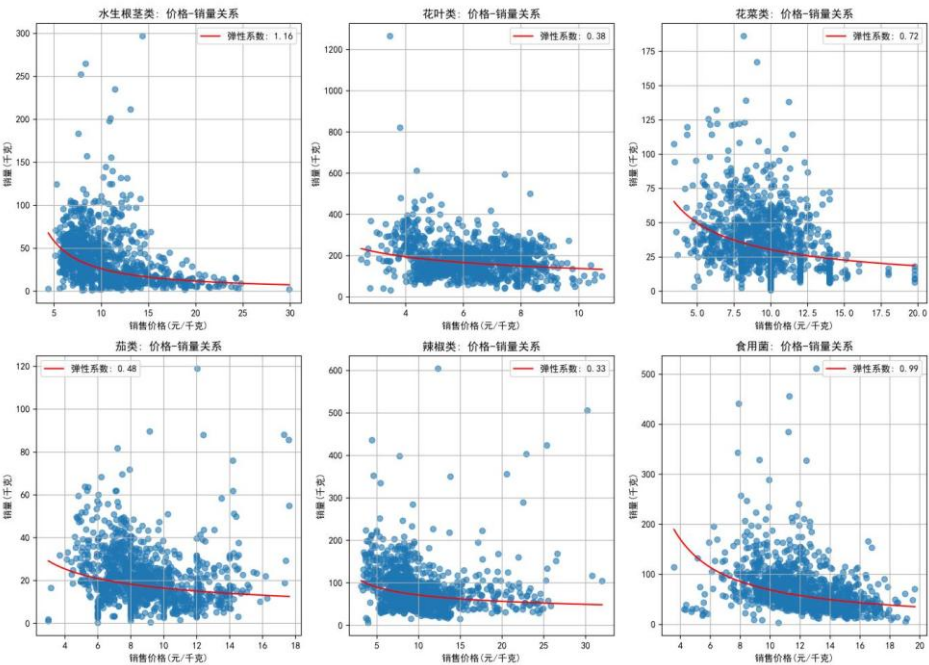


图 11 价格弹性分析图

水生根茎类（弹性系数 1.16 ）：高弹性品类（ $\varepsilon >1$ ），价格小幅变动会引发销量明显波动。消费者对其价格敏感，适合薄利多销，可适度降价，借销量增长摊薄成本、提升总利润，比如搞满减、降价促销活动。

花叶类（ 0.38 ）、茄类（ 0.48 ）、辣椒类（ 0.33 ）：低弹性品类（ $\varepsilon <1$ ），价格变化对销量影响小。消费者价格敏感度低，可高加成率，通过提高单位利润增加收益，像优化产品包装、打造特色，支撑较高定价。

花菜类（ 0.72 ）、食用菌（ 0.99 ）：花菜类接近低弹性，食用菌接近高弹性临界。花菜类可适度兼顾利润与销量策略；食用菌弹性系数 0.99 ，虽接近 1 但仍小于 1 ，暂按低弹性思路，若价格微调后观察销量变化，再精准调整，比如小幅度涨价测试市场接受度。

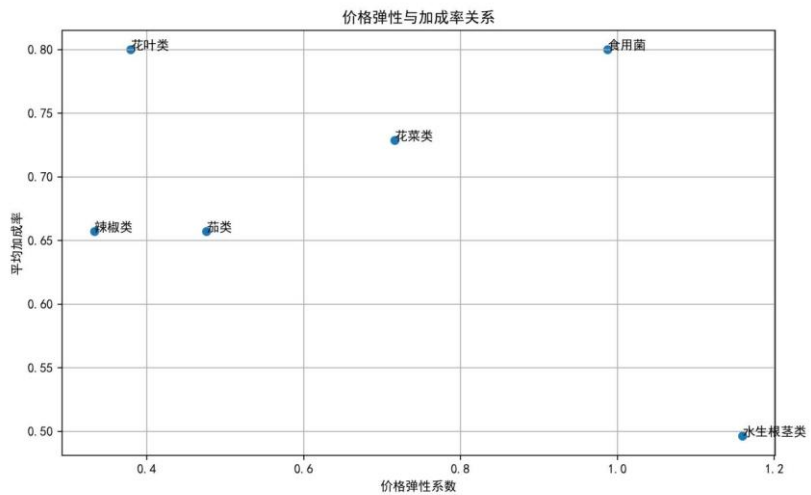


图 12 价格弹性与加成率关系

由图 12 直观可见，价格弹性系数与平均加成率呈负相关态势：水生根茎类的弹性系数较高（约为 1.2 ），加成率较低（为 0.5 ）；花叶类的弹性系数较低（为 0.4 ），加成率较高（为 0.8 ）。此情况与“高弹性品类实施薄利多销策略（低加成）、低弹性品类追求单位利润最大化（高加成）”的定价策略逻辑高度相符，验证了该定价策略的合理性。

高弹性 - 低加成代表（水生根茎类）：弹性系数达 1.2 ，加成率仅 0.5 ，是典型的“以量取胜”品类。需持续优化供应链降本，保障低价走量的利润空间；低弹性 - 高加成代表（花叶类、食用菌）：弹性系数 ≤ 0.4 ，加成率 ≥ 0.8 ，消费者对价格不敏感。可聚焦品质升级（如有机认证、精品包装）、打造差异化（独家品种、产地直供）；中间区间

（花菜类、辣椒类、茄类）：弹性与加成率相对均衡，需平衡 “ 走量 ” 与 “ 盈利 ” 。可尝试 “ 阶梯定价 ” 。

表 2 各品类价格弹性系数和基准需求常数

品类	弹性系数 ϵ	基准需求常数 a	R^2	p
水生根茎类	1.160	376.73	0.192	0.000
花叶类	0.380	328.69	0.046	0.000
花菜类	0.716	157.76	0.090	0.000
茄类	0.476	49.21	0.040	0.000
辣椒类	0.333	153.80	0.062	0.000
食用菌	0.988	668.57	0.145	0.000

由表 2 可得，各品类完整需求表达函数为：

$$Q = 376.73 \times P^{-1.16}$$

$$Q = 328.69 \times P^{-0.38}$$

$$Q = 157.76 \times P^{-0.72}$$

$$Q = 49.21 \times P^{-0.48}$$

$$Q = 153.80 \times P^{-0.33}$$

$$Q = 668.57 \times P^{-0.99}$$

回归结果同时提供拟合优度 R^2 和显著性 p 值作为模型评价指标：

R^2 （决定系数）越接近 1，意味着模型对数据的拟合效果越好，即模型能够更准确地解释和预测因变量的变化。具体来说，当 R^2 接近 1 时，表明模型解释了大部分的因变量变异，拟合程度较高。然而，在实际应用中，由于不同单品销售数据的分布较为分散，数据点之间的差异较大，导致模型的实际拟合优度表现适中，既不过于理想也不过于糟糕，处于一个相对合理的水平。

p 值是用来检验回归系数的统计显著性的重要指标。在统计学中， p 值越小，说明拒绝原假设（即回归系数为零）的证据越强。实证分析结果显示，在各品类的回归模型中， p 值均小于 0.01，这一结果表明各回归系数的统计显著性非常高，几乎可以肯定地认为这些回

归系数不为零。换句话说，模型中的自变量对因变量有显著的影响，估计结果具有高度统计显著性，从而使得模型的可信度大大提高，能够为决策提供可靠的依据。

表 3 2023.7.1-2023.7.7 各品类日补货总量预测数据（单位：千克）

日期	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
2023/7/1	20.60347 499	181.2018 397	25.58824 794	29.49453 18	116.2396 81	53.29054 396
2023/7/2	24.75829 71	171.6614 048	23.79349 116	27.48201 872	103.1947 518	46.21841 309
2023/7/3	12.94985 762	118.4496 668	16.31045 107	15.92422 522	69.78419 483	30.84203 71
2023/7/4	16.43488 347	127.4783 596	14.42824 731	15.07544 36	70.91343 506	30.18334 701
2023/7/5	18.02010 629	121.8200 146	15.15978 979	14.92281 545	71.20010 2	32.52227 388
2023/7/6	19.31895 979	120.7693 456	16.64188 769	14.87780 328	73.89075 985	32.19270 017
2023/7/7	22.27374 559	135.5799 55	18.56049 385	17.68826 423	85.27786 023	38.64032 908

根据表 3 中数据分别绘制各品类补货策略折线图，实线表示优化后的补货量，虚线表示预测的需求量，两者差异反映安全库存和损耗调整。

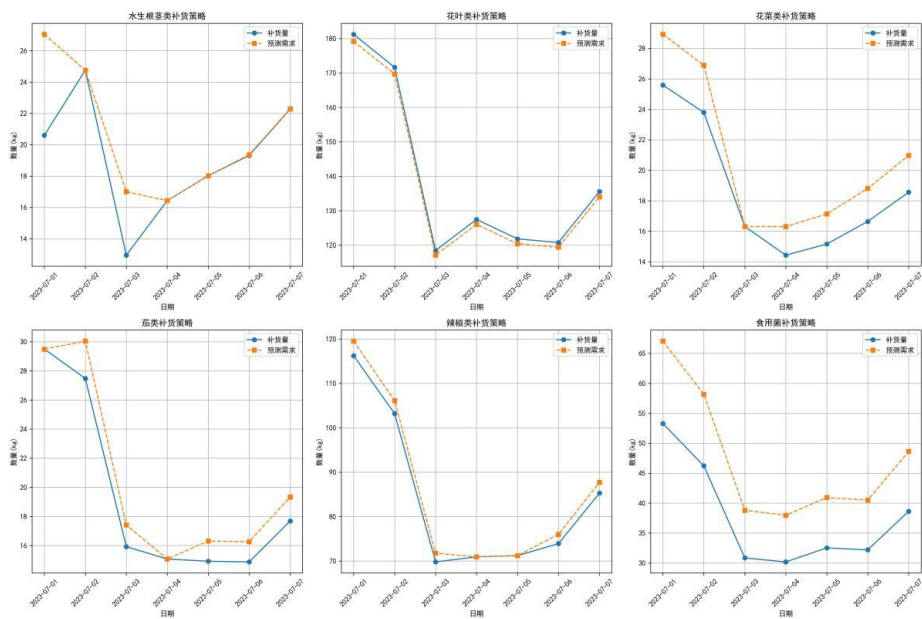


图 13 2023.7.1-2023.7.7 各品类日补货总量预测折线图

如图 13 所示：补货量>预测需求：如花叶类，辣椒类和茄类部分日期，设置安全库存，用于缓冲需求波动、供应链延迟等不确定性，保障供应稳定，降低缺货风险。

补货量<预测需求：像花菜类，食用菌类和水生根茎类等品类的某些日期，针对高损耗品类，主动策略性缺货，避免过多备货因损耗造成成本浪费，平衡成本与供应。

表 4 2023.7.1-2023.7.7 各品类定价策略（单位：元）

日期	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
2023/7 /1	14.50687 807	7.874758 903	12.54598 813	7.885042 658	12.81769 866	15.71382 505
2023/7 /2	11.47101 839	7.874758 903	12.54598 813	10.91775 137	12.81769 866	15.71382 505
2023/7 /3	14.50687 807	7.874758 903	9.060991 43	10.91775 137	12.81769 866	15.71382 505
2023/7 /4	10.47718 972	7.874758 903	12.54598 813	7.885042 658	9.257226 812	15.71382 505
2023/7 /5	10.47718 972	7.874758 903	12.54598 813	10.91775 137	9.257226 812	15.71382 505

日期	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
2023/7/6	11.49665 423	7.874758 903	12.54598 813	10.91775 137	12.81769 866	15.71382 505
2023/7/7	11.48050 876	7.874758 903	12.54598 813	10.91775 137	12.81769 866	15.71382 505

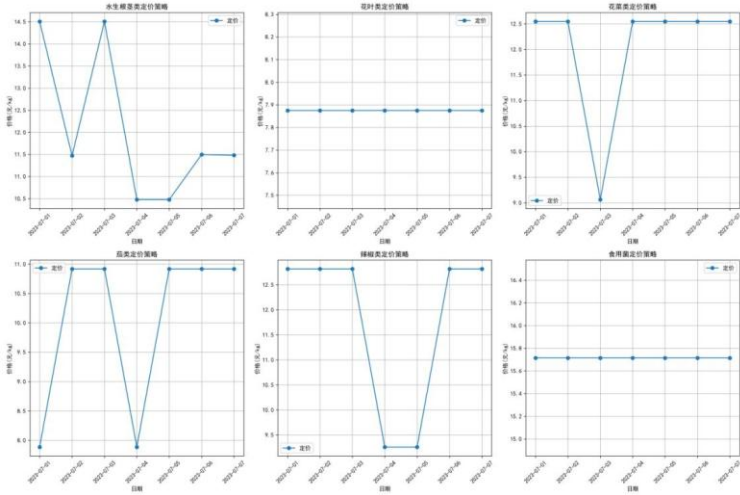


图 14 2023.7.1-2023.7.7 各品类定价策略折线图

由问题一得到的规律，不同的蔬菜考虑周末效应和季节性因素调整一周每日定价，比如花菜类与水生根茎类增幅较高，说明周末对该品类销量拉动显著，利用周末消费高峰提价，符合价格弹性与市场需求规律。

表 5 2023.7.1-2023.7.7 各品类加成比率

日期	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
2023/7/1	0.8	0.8	0.8	0.3	0.8	0.8
2023/7/2	0.423313342	0.8	0.8	0.8	0.8	0.8
2023/7/3	0.8	0.8	0.3	0.8	0.8	0.8
2023/7/4	0.3	0.8	0.8	0.3	0.3	0.8
2023/7/5	0.3	0.8	0.8	0.8	0.3	0.8
2023/7/6	0.426494213	0.8	0.8	0.8	0.8	0.8
2023/7/7	0.424490898	0.8	0.8	0.8	0.8	0.8

表 6 2023.7.1-2023.7.7 各品类预测需求

日期	水生根茎类	花叶类	花菜类	茄类	辣椒类	食用菌
2023/7 /1	27.04643 484	179.1329 196	28.91888 142	29.49453 18	119.5273 783	67.05446 589
2023/7 /2	24.75307 149	169.7014 151	26.89051 438	30.04712 604	106.1134 893	58.15573 971
2023/7 /3	16.99943 725	117.0972 363	16.31045 107	17.41055 514	71.75795 553	38.80794 173
2023/7 /4	16.43488 347	126.0228 416	16.30626 582	15.07544 36	70.91343 506	37.97912 467
2023/7 /5	18.02010 629	120.4291 023	17.13302 78	16.31567 612	71.20010 2	40.92215 133
2023/7 /6	19.36495 209	119.3904 295	18.80803 944	16.26646 262	75.98066 975	40.50745 508
2023/7 /7	22.29041 28	134.0319 349	20.97637 642	19.33924 541	87.68984 032	48.62038 248

5.3 问题三模型建立与求解

5.3.1 问题三求解思路

(1) 数据准备：筛选 6 月 24-30 日有销售的单品作为候选集，计算各单品日均销量、价格、成本和损耗率等基础参数。

(2) 需求预测：沿用问题二的品类弹性系数，区分富有弹性和缺乏弹性商品，分别采用最优定价公式和成本加成法确定单品价格，并计算预测需求和预期利润。

(3) 单品选择：按品类需求比例分配 27-33 个单品配额，在各品类内选择利润最高的单品，确保品类多样性和收益最大化。

(4) 补货分配：根据预测需求和损耗率计算补货量，满足最小陈列量 2.5 kg 要求，最终

输出单品补货与定价方案。

5.3.2 问题三模型建立

基于 2023 年 6 月 24-30 日的销售数据，对于每个单品计算关键经营参数：

$$\dot{q}_i = \frac{\sum_{t=1}^7 q}{n_i} (22)$$

$$\dot{p}_i = \frac{\sum_{t=1}^7 p}{n_i} (23)$$

$$\dot{w}_i = \frac{\sum_{t=1}^7 w}{n_i} (24)$$

损耗率：从附件 4 中获取。

$$c_i = \frac{\dot{w}_i}{1 - \delta_i/100} (25)$$

弹性系数继承：

从问题二结果中获取各品类的价格弹性系数 ε_i ，同一品类内单品共享弹性特性。

定价策略：

基于品类弹性特性采用差异化定价 [4]：

$$p_i = \left(\frac{\varepsilon_i}{\varepsilon_i - 1} \right) c_i (\varepsilon_i > 1) (26)$$

$$p_i = (1 + m_k) c_i (\varepsilon_i \leq 1) (27)$$

需求预测：

$$D_i(p_i) = \dot{q}_i \dot{p}_i^{\varepsilon_k} (p_i)^{-\varepsilon_k} (28)$$

$$\pi_i = D_i(p_i) p_i - \frac{D_i(p_i)}{1 - \delta_i/100} \dot{w}_i (29)$$

$$\max \sum_{i=1}^n \pi_i x_i (30)$$

约束条件：

(1) 单品数量约束： $27 \leq \sum_{i=1}^n \pi_i \leq 33$ ；

(2) 最小补货量： $Q_i \geq 2.5x_i$ ；

*)D i (p i

(3) 需求满足约束: $Q_i \geq \frac{D_i(p_i)}{1-\delta_i/100} x_i;$

$1-\delta_i/100 \cdot x_i \leq Q_i;$

(4) 非负约束: $Q_i \geq 0;$

鉴于原问题属于混合整数非线性规划，计算难度较大，因此通过分层优化策略，将其分解为可管理的子问题：第一阶段：品类配额分配

$$n_k = \max \left(1, \left\lfloor \frac{\sum_{i \in C_k} D_i(p_i)}{\sum_{i=1}^n D_i(p_i)} \times 33 + 0.5 \right\rfloor \right) \tag{31}$$

调整 n_k 使 $n_k \sigma \in [27, 33]$ ；

第二阶段：品类内单品选择

$$\sum_{i \in C_k} \pi_i x_i \text{ s.t. } \sum_{i \in C_k} x_i = n_k \tag{32}$$

补货量计算：

$$Q_i = \max \left(2.5, \frac{D_i(p_i)}{1-\delta_i/100} \right) \tag{33}$$

5.3.3 问题三模型求解与分析

筛选 6 月 24-30 日有销售记录的 49 个单品。

表 7 进货的蔬菜单品补货定价策略

单品名称	品类名称	成本(元/kg)	加成率	预测需求(kg)	进货量	定价	利润
小米椒(份)	辣椒类	26.79	3.93	2.37	0.657	24.35	37.96
螺丝椒	辣椒类	7.04	13.8	8.33	0.657	6.4	35.02
芜湖青椒(1)	辣椒类	15	5.92	3.57	0.657	13.63	31.99
螺丝椒(份)	辣椒类	11.57	6.09	3.67	0.657	10.51	25.39
红椒	辣椒类	2.5	23.73	14.32	0.657	1.89	17.76

单品名称	品类名称	成本(元/kg)	加成率	预测需求(kg)	进货量	定价	利润
(2)							
姜蒜小米椒组合装(小份)	辣椒类	7.83	4.5	2.71	0.657	7.11	12.69
小皱皮(份)	辣椒类	12.11	2.79	1.68	0.657	11.01	12.16
七彩椒(2)	辣椒类	2.5	22.46	13.55	0.657	1.15	10.25
青红杭椒组合装(份)	辣椒类	2.5	5.87	3.54	0.657	2.1	4.88
青线椒(份)	辣椒类	2.5	5.18	3.12	0.657	0.94	1.93
云南生菜(份)	花叶类	31.02	7.11	3.95	0.8	27.05	85.44
云南油麦菜(份)	花叶类	21.65	5.7	3.17	0.8	18.88	47.85
娃娃菜	花叶类	10.69	8.78	4.88	0.8	9.32	36.39
菠菜(份)	花叶类	9.66	8.22	4.57	0.8	8.42	30.75
竹叶菜	花叶类	13.86	4.86	2.7	0.8	12.08	26.07
菠菜	花叶类	2.5	21.28	11.82	0.8	1.94	18.34
苋菜	花叶类	9.13	5.14	2.85	0.8	7.96	18.18
奶白菜	花叶类	8.21	5.4	3	0.8	7.16	17.19
外地茼蒿	花叶类	2.5	20.09	11.16	0.8	1.82	16.26

单品名称	品类名称	成本(元/kg)	加成率	预测需求(kg)	进货量	定价	利润
蒿							
木耳菜	花叶类	6.4	6.28	3.49	0.8	5.58	15.56
上海青	花叶类	4.29	8.69	4.83	0.8	3.74	14.45
云南生菜	花叶类	3.01	12.19	6.77	0.8	2.62	14.22
小青菜(1)	花叶类	5.43	5.71	3.17	0.8	4.73	12
红薯尖	花叶类	4.86	6.27	3.48	0.8	4.24	11.81
西峡花菇(1)	食用菌	3.69	31.48	17.49	0.8	3.54	49.49
双孢菇(盒)	食用菌	8.57	6.14	3.41	0.8	8.22	22.41
金针菇(盒)	食用菌	12.1	2.63	1.46	0.8	11.59	13.54
海鲜菇(包)	食用菌	7.25	3.52	1.95	0.8	6.94	10.85
鲜木耳(份)	食用菌	3.92	2.58	1.44	0.8	3.76	4.31
紫茄子(2)	茄类	11.15	6.59	3.98	0.657	10.42	27.25
长线茄	茄类	4.43	12.43	7.5	0.657	4.14	20.43
西兰花	花菜类	12.19	14.81	8.57	0.729	11.06	69.08
净藕(1)	水生根茎类	2.5	82.39	11.36	0.496	0.8	56.59

根据以上补货策略，分析单品预测需求与补货量关系，绘制散点图（即图 15）

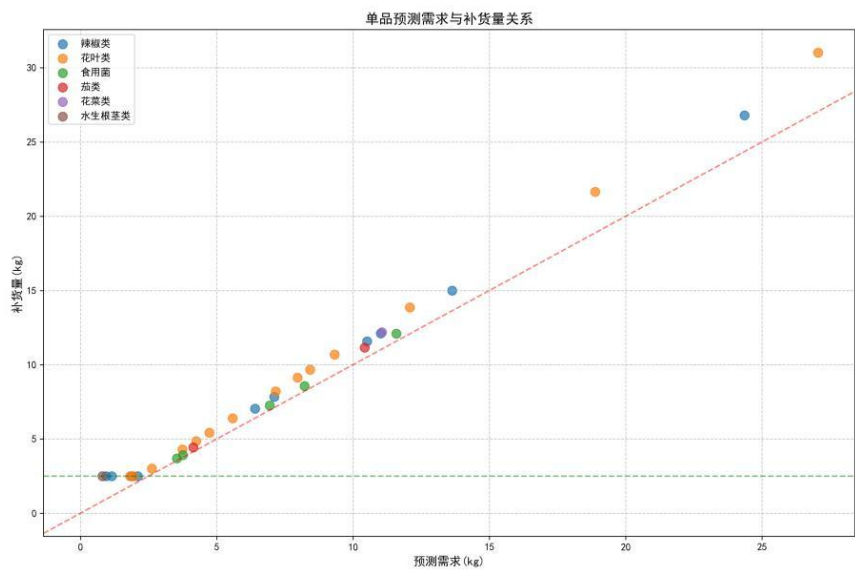


图 15 单品预测需求与补货量关系

在散点图中，我们可以观察到大部分的数据点呈现出一种明显的分布趋势：它们主要集中在 $y = x$ 这条直线附近。这一现象表明，当前的预测模型在整体上具有较高的准确性，能够较好地反映实际需求情况。具体来说，花叶类商品的需求量相对较大，与之相对应的补货量也呈现出较高的水平。然而，值得注意的是，部分单品的补货量远远超过了预测的需求量，这种情况可能暗示着这些单品存在一定的滞销风险，需要引起相关部门的重视。此外，对于图中出现的个别离群点，我们不能一概而论，而应根据具体的实际情况进行细致的分析和判断，酌情进行调整，以确保补货策略的合理性和有效性。

由表 7 中单品补货策略按类汇总，得到各品类补货量及其利润（见表 8）

表 8 各品类补货量信息

品类名称	品类补货量 (kg)	预期利润 (元)	平均加成率
水生根茎类	2.50	56.59	0.496
花叶类	133.21	364.51	0.800
花菜类	12.19	69.08	0.729
茄类	15.58	47.68	0.657
辣椒类	90.34	190.03	0.657
食用菌	35.53	100.60	0.800

根据各品类预期利润，绘制饼图来描述各品类预期利润占比（图 16）

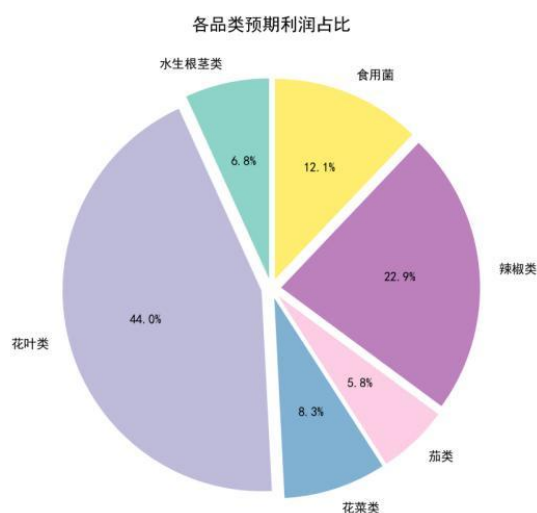


图 16 各品类预期利润占比

通过该饼图可清晰看出，花叶类和辣椒类共同贡献 66.9% 的利润，统筹规划时应优先保障高利润品类的库存和陈列资源。

5.4 问题四模型建立与求解

5.4.1 问题四模型求解与分析

通过上述三问的建模与分析，本文认为商超可以进一步采集获取以下数据，从而建立更为完善的模型，并更好地制定蔬菜商品的补货和定价决策。

(1) 外部环境数据

1) 节假日信息：特殊日期需求模式与平日有显著差异，客流量也会有较大变化； 2) 季节信息：不同季节对不同食材的需求可能不同，对客人的口味喜好也会有影响； 3) 市场价格调整：了解市场价格可通过优化定价策略来得到更高的利润；

(2) 实时运营数据 实时库存数据：了解当前库存可通过动态补货防止出现缺货现象，同时若出现产品积压 也可通过促销等防止蔬菜腐败；

(3) 供应链数据

1) 采购价格波动数据：把握采购价格规律，降低采购成本；

2) 物流数据：对物流时间及成本进行优化，降低损耗率，注意供应链风险管控。

六、模型评价

6.1 模型的优点

(1) 模型充分结合实际，简化了折扣率、损耗率等条件，考虑了诸多重要因素，得到合理的模型，如基于价格弹性需求模型和利润最大化优化模型相结合的补货和定价模型等。

这样得到的模型贴合实际，具有较高的应用价值，可以推广和运用到 其他行业（比如家电行业）的超市进价和补货决策方案中；（2）本文使用的算法具有优化能力强、并行化能力强等优点，对于求解诸多非线性回归模型非常适用；（3）L - BFGS - B 算法通过拟牛顿法近似 Hessian 矩阵，避免了直接计算二阶导数的复杂性，在保证求解精度的同时显著提高了计算效率。

6.2 模型的不足

（1） 实际应用中蔬菜销量与定价不一定是线性的，而本文将其作为线性因子处理，忽略了边际效应的影响。

参考文献:

- [1] 陈华根, 吴健生, 王家林等. 模拟 L - BFGS - B 算法机理研究 [J]. 同济大学学报 (自然科学版), 2004(06):802-805.
- [2] 赵杰斌, 黄穗东, 孙远明等. 基于数据挖掘的江门市蔬菜食品安全风险分析与预测 [J/OL]. 食品工业科技 :1-16[2023-09-10].<https://doi.org/10.13386/j.issn1002-0306.2022120006>.
- [3] 毛莉莎. 供应链视角下蔬菜批发市场定价策略及产销模式研究 [D]. 中南林业科技大学, 2023. DOI:10.27662/d.cnki.gznlc.2022.000680.
- [4] 李干琼, 王盛威, 许世卫等. 大城市蔬菜供需分析与预测研究——以上海市为例 [J]. 上海 农业学报, 2021, 37(04):125-132. DOI:10.15955/j.issn1000-3924.2021.04.21.
- [5] 陈林生, 孙利君, 马佳. 蔬菜价格短期预测模型比较研究——以上海市青菜价格为例 [J]. 价格理论与实践, 2020(09):68-71+178. DOI:10.19851/j.cnki.cn11-1010/f.2020.09.391.

附录

附录 1

第一问求解代码

```
function main()
    % 主分析流程
    disp('开始数据加载与预处理...');
    [sales_data, item_info] = load_and_preprocess();

    disp('进行品类层面分析...');
    category_results = category_analysis(sales_data);

    disp('进行单品层面分析...');
    item_results = item_analysis(sales_data, item_info);

    disp('生成分析报告...');
    report = generate_report(category_results, item_results);

    disp('分析完成! ');
end

function [sales_data, item_info] = load_and_preprocess()
    % 加载附件 - 使用北太天元兼容的函数
    % 读取 Excel 文件
    [~, ~, item_info] = xlsread('23.C.附件 1.xlsx');    % 商品信息
    [~, ~, sales_data_raw] = xlsread('23.C.附件 2.xlsx');    % 销售流水
    [~, ~, wholesale_price_raw] = xlsread('23.C.附件 3.xlsx');    % 批发价格
    [~, ~, loss_rate_raw] = xlsread('23.C.附件 4.xlsx');    % 单品损耗率
```

```

% 提取表头和数据
item_header = item_info(1, :);
item_data = item_info(2:end, :);

sales_header = sales_data_raw(1, :);
sales_data_cell = sales_data_raw(2:end, :);

wholesale_header = wholesale_price_raw(1, :);
wholesale_data = wholesale_price_raw(2:end, :);

loss_header = loss_rate_raw(1, :);
loss_data = loss_rate_raw(2:end, :);

% 转换为结构数组
sales_data = struct();
for i = 1:size(sales_data_cell, 1)
    for j = 1:length(sales_header)
        field_name = strrep(sales_header{j}, ' ', '_');
        field_name = strrep(field_name, '(', '');
        field_name = strrep(field_name, ')', '');
        field_name = strrep(field_name, '/', '_');
        field_name = strrep(field_name, '.', '');

        sales_data(i).(field_name) = sales_data_cell{i, j};
    end
end

```

```

% 添加日期信息
if isfield(sales_data(i), '销售日期')
    sales_date = sales_data(i).销售日期;
    if ischar(sales_date)
        sales_date = datenum(sales_date);
    end
    sales_data(i).销售日期_num = sales_date;
    sales_data(i).月份 = month(sales_date);
    sales_data(i).季度 = quarter(sales_date);
    sales_data(i).星期 = weekday(sales_date) - 1;
    sales_data(i).是否周末 = sales_data(i).星期 >= 5;
end
end

% 创建商品信息映射
item_map = struct();
for i = 1:size(item_data, 1)
    item_code = item_data{i, 1};
    item_map.(['item_' num2str(item_code)]) = struct(...

        'name', item_data{i, 2}, ...
        'category_code', item_data{i, 3}, ...
        'category_name', item_data{i, 4} ...
    );
end

% 创建批发价格映射
price_map = struct();

```

```

for i = 1:size(wholesale_data, 1)
    date_val = wholesale_data{i, 1};
    if ischar(date_val)
        date_val = datenum(date_val);
    end
    item_code = wholesale_data{i, 2};
    price = wholesale_data{i, 3};

    key = [num2str(date_val) '_' num2str(item_code)];
    price_map.(['price_' key]) = price;
end

% 创建损耗率映射
loss_map = struct();
for i = 1:size(loss_data, 1)
    item_code = loss_data{i, 1};
    loss_rate = loss_data{i, 2};
    loss_map.(['loss_' num2str(item_code)]) = loss_rate;
end

% 合并数据到销售记录
for i = 1:length(sales_data)
    item_code = sales_data(i).单品编码;

    % 添加商品信息
    if isfield(item_map, ['item_' num2str(item_code)])
        item_info = item_map.(['item_' num2str(item_code)]);
        sales_data(i).单品名称 = item_info.name;
        sales_data(i).分类编码 = item_info.category_code;
    end
end

```

```

        sales_data(i).分类名称 = item_info.category_name;
    end

    % 添加批发价格
    date_val = sales_data(i).销售日期_num;
    key = [num2str(date_val) '_' num2str(item_code)];
    if isfield(price_map, ['price_' key])
        sales_data(i).批发价格 = price_map(['price_' key]);
    end

    % 添加损耗率
    if isfield(loss_map, ['loss_' num2str(item_code)])
        sales_data(i).损耗率 = loss_map(['loss_' num2str(item_code)]);
    else
        sales_data(i).损耗率 = NaN;
    end

    % 计算实际成本
    if isfield(sales_data(i), '批发价格') && isfield(sales_data(i), '损耗率')
        if ~isnan(sales_data(i).损耗率)
            sales_data(i).实际成本 = sales_data(i).批发价格 / (1 -
sales_data(i).损
耗率/100);
        else
            sales_data(i).实际成本 = NaN;
        end
    end
end
end

```

```

% 处理损耗率缺失值
% 计算各品类的平均损耗率
category_loss = struct();
for i = 1:length(sales_data)
    if isfield(sales_data(i), '分类名称') && isfield(sales_data(i), '损耗率')
        cat_name = sales_data(i).分类名称;
        loss_val = sales_data(i).损耗率;

        if ~isnan(loss_val)
            if ~isfield(category_loss, cat_name)
                category_loss.(cat_name) = [];
            end
            category_loss.(cat_name) = [category_loss.(cat_name)
loss_val];
        end
    end
end

% 计算平均损耗率
cat_names = fieldnames(category_loss);
for i = 1:length(cat_names)
    cat_name = cat_names{i};
    category_loss.(cat_name) = mean(category_loss.(cat_name));
end

% 填充缺失的损耗率

```

```

for i = 1:length(sales_data)
    if isfield(sales_data(i), '损耗率') && isnan(sales_data(i).损耗率) &&
isfield(sales_data(i), '分类名称')
        cat_name = sales_data(i).分类名称;
        if isfield(category_loss, cat_name)
            sales_data(i).损耗率 = category_loss.(cat_name);

            % 重新计算实际成本
            if isfield(sales_data(i), '批发价格')
                sales_data(i).实际成本 = sales_data(i).批发
价格 / (1 -
sales_data(i).损耗率/100);
            end
        end
    end
end
end
end
end

```

```

function results = category_analysis(sales_data)
    % 提取所有分类名称
    categories = {};
    for i = 1:length(sales_data)
        if isfield(sales_data(i), '分类名称')
            cat_name = sales_data(i).分类名称;
            if ~ismember(cat_name, categories)

                categories{end+1} = cat_name;
            end
        end
    end
end

```

```

        end
    end

% 按日期和分类汇总销量
dates = [];
for i = 1:length(sales_data)
    if isfield(sales_data(i), '销售日期_num')
        dates = [dates; sales_data(i).销售日期_num];
    end
end
unique_dates = unique(dates);

% 初始化每日销量矩阵
daily_sales = zeros(length(unique_dates), length(categories));

% 填充每日销量矩阵
for i = 1:length(sales_data)
    if isfield(sales_data(i), '销售日期_num') && isfield(sales_data(i), '分
类名称') &&
isfield(sales_data(i), '销量_千克')
        date_idx = find(unique_dates == sales_data(i).销售日期_num);
        cat_idx = find(strcmp(categories, sales_data(i).分类名称));

        if ~isempty(date_idx) && ~isempty(cat_idx)
            daily_sales(date_idx, cat_idx) = daily_sales(date_idx,
cat_idx) +
sales_data(i).销量_千克;
        end
    end
end

```

```

end

% 计算描述性统计
stats = struct();
for i = 1:length(categories)
    cat_sales = daily_sales(:, i);
    stats(i).分类名称 = categories{i};
    stats(i).mean = mean(cat_sales);
    stats(i).std = std(cat_sales);

    stats(i).偏度 = skewness(cat_sales);
    stats(i).峰度 = kurtosis(cat_sales);
    stats(i).变异系数 = stats(i).std / stats(i).mean;
end

% 周内波动分析
weekday_analysis = struct();
for i = 1:length(categories)
    cat_name = categories{i};
    weekday_sales = [];
    weekend_sales = [];

    for j = 1:length(sales_data)
        if isfield(sales_data(j), '分类名称') && strcmp(sales_data(j).分
类名称,
cat_name) && ...
            isfield(sales_data(j), '是否周末') && isfield(sales_data(j),
'销量_千克')

```

```

        if sales_data(j).是否周末
            weekend_sales = [weekend_sales; sales_data(j).销量_
千克];

        else
            weekday_sales = [weekday_sales; sales_data(j).销量_
千克];

        end

    end

    end

    if ~isempty(weekday_sales) && ~isempty(weekend_sales)
        weekday_analysis(i).分类名称 = cat_name;
        weekday_analysis(i).工作日平均销量 = mean(weekday_sales);
        weekday_analysis(i).周末平均销量 = mean(weekend_sales);
        weekday_analysis(i).周末增幅 = (weekday_analysis(i).周末平均销量
-
weekday_analysis(i).工作日平均销量) / weekday_analysis(i).工作日平均销量;

        end

    end

% 品类间相关系数
correlation_matrix = corr(daily_sales);

% 绘制热力图

```

```

figure;
imagesc(correlation_matrix);
colorbar;
set(gca, 'XTick', 1:length(categories), 'XTickLabel', categories);
set(gca, 'YTick', 1:length(categories), 'YTickLabel', categories);
xtickangle(45);
title('品类间销量相关系数矩阵');
saveas(gcf, '品类相关系数热力图.png');

% 保存结果
results = struct();
results.stats = stats;
results.weekday_analysis = weekday_analysis;
results.correlation_matrix = correlation_matrix;
results.categories = categories;
results.daily_sales = daily_sales;
end

function results = item_analysis(sales_data, item_info)
% 按单品汇总总销量
items = struct();
for i = 1:length(sales_data)
    if isfield(sales_data(i), '单品编码') && isfield(sales_data(i), '销量_
    千克')

        item_code = sales_data(i).单品编码;
        item_name = sales_data(i).单品名称;
        category = sales_data(i).分类名称;
    end
end

```

```

        if ~isfield(items, ['item_' num2str(item_code)])
            items.(['item_' num2str(item_code)]) = struct(...
                'code', item_code, ...
                'name', item_name, ...
                'category', category, ...
                'total_sales', 0, ...
                'loss_rate', [] ...
            );
        end
        items.(['item_' num2str(item_code)]).total_sales = ...

            items.(['item_' num2str(item_code)]).total_sales +
sales_data(i). 销量_千
克;

        if isfield(sales_data(i), '损耗率')
            items.(['item_' num2str(item_code)]).loss_rate = [...
                items.(['item_' num2str(item_code)]).loss_rate;
sales_data(i).损耗率
];
        end
    end
end

% 转换为数组
item_fields = fieldnames(items);
item_list = struct();

```

```

for i = 1:length(item_fields)
    item_list(i).code = items.(item_fields{i}).code;
    item_list(i).name = items.(item_fields{i}).name;
    item_list(i).category = items.(item_fields{i}).category;
    item_list(i).total_sales = items.(item_fields{i}).total_sales;
    item_list(i).avg_loss_rate = mean(items.(item_fields{i}).loss_rate);
end

```

% 按品类分析单品分布

```

categories = {};
for i = 1:length(item_list)
    if ~ismember(item_list(i).category, categories)
        categories{end+1} = item_list(i).category;
    end
end

```

```

pareto_results = struct();
for i = 1:length(categories)
    cat_name = categories{i};
    cat_items = [];

    for j = 1:length(item_list)
        if strcmp(item_list(j).category, cat_name)
            cat_items = [cat_items; item_list(j)];
        end
    end
end
end

```

```

% 按销量排序
[~, idx] = sort([cat_items.total_sales], 'descend');
cat_items = cat_items(idx);

% 计算销量占比和累计占比
total_sales = sum([cat_items.total_sales]);
for j = 1:length(cat_items)
    cat_items(j).sales_ratio = cat_items(j).total_sales / total_sales;
end

cum_ratio = 0;
for j = 1:length(cat_items)
    cum_ratio = cum_ratio + cat_items(j).sales_ratio;
    cat_items(j).cum_ratio = cum_ratio;
end

% 帕累托分析
n_20 = max(1, round(length(cat_items) * 0.2));
pareto_ratio = sum([cat_items(1:n_20).sales_ratio]);

pareto_results(i).category = cat_name;
pareto_results(i).pareto_ratio = pareto_ratio;
pareto_results(i).items = cat_items;

% 绘制销量分布图
figure;
bar([cat_items.sales_ratio]);
hold on;
plot([cat_items.cum_ratio], 'r-o', 'LineWidth', 2);

```

```

yline(0.8, '--', '80%分界线', 'LineWidth', 2);
title([cat_name '单品销量分布']);
xlabel('单品序号');
ylabel('销量占比');
legend('销量占比', '累计占比', 'Location', 'best');
saveas(gcf, [cat_name '单品销量分布.png']);

end

% 单品损耗率与销量关系
figure;
colors = lines(length(categories));
legend_entries = {};

for i = 1:length(categories)
    cat_name = categories{i};
    loss_rates = [];
    sales_volumes = [];

    for j = 1:length(item_list)
        if strcmp(item_list(j).category, cat_name)
            loss_rates = [loss_rates; item_list(j).avg_loss_rate];
            sales_volumes = [sales_volumes; item_list(j).total_sales];
        end
    end

    scatter(loss_rates, sales_volumes, 50, colors(i,:), 'filled');
    hold on;
end

```

```

        legend_entries{end+1} = cat_name;
end

title('单品损耗率与销量关系');
xlabel('损耗率(%)' );
ylabel('销量(千克)' );
legend(legend_entries);
grid on;
saveas(gcf, '损耗率与销量关系.png');

% 计算损耗率与销量的相关系数
all_loss_rates = [item_list.avg_loss_rate];
all_sales = [item_list.total_sales];
loss_sales_corr = corr(all_loss_rates(:), all_sales(:));

% 保存结果
results = struct();

    results.pareto_results = pareto_results;
    results.loss_sales_corr = loss_sales_corr;
    results.item_list = item_list;
end

function report = generate_report(category_results, item_results)
    % 创建报告文本
    report = sprintf('蔬菜类商品销售分析报告\n\n');

```

```

report
=
[report
sprintf('=====\n')];
    report = [report sprintf('一、品类层面分析\n')];
report
=
[report
sprintf('=====\n')];

% 品类描述性统计
report = [report sprintf('\n1. 各品类销售统计特征:\n')];
for i = 1:length(category_results.stats)
    stats = category_results.stats(i);
    report = [report sprintf('    - %s: 平均日销量=%.2fkg, 标准差=%.2f, 波
动系数
=%.2f\n', ...
        stats.分类名称, stats.mean, stats.std, stats.变异系数)];
end

% 周内波动
report = [report sprintf('\n2. 周末销售变化:\n')];
for i = 1:length(category_results.weekday_analysis)
    analysis = category_results.weekday_analysis(i);
    if isfield(analysis, '分类名称')
        report = [report sprintf('    - %s: 工作日 =%.2fkg, 周末
=%.2fkg, 增幅
=%.2f%%\n', ...
            analysis.分类名称, analysis.工作日平均销量, analysis.周末
平均销量,
analysis.周末增幅*100)];

```

```

        end
    end

    % 品类相关性
    report = [report sprintf('\n3. 品类间相关性:\n')];

    corr_matrix = category_results.correlation_matrix;
    categories = category_results.categories;

    report = [report sprintf('强正相关 (>0.7):\n')];
    for i = 1:length(categories)
        for j = i+1:length(categories)
            if corr_matrix(i, j) > 0.7
                report = [report sprintf('          - %s & %s: %.3f\n',
categories{i},
categories{j}, corr_matrix(i, j))];
            end
        end
    end

    report = [report sprintf('\n 强负相关 (<-0.5):\n')];
    for i = 1:length(categories)
        for j = i+1:length(categories)
            if corr_matrix(i, j) < -0.5
                report = [report sprintf('          - %s & %s: %.3f\n',
categories{i},
categories{j}, corr_matrix(i, j))];
            end
        end
    end

```

```

        end
    end
end

% 单品层面分析
report =
[report
sprintf('\n=====\\n')];
report = [report sprintf('二、单品层面分析\\n')];
report =
[report
sprintf('=====\\n')];

% 帕累托分析
report = [report sprintf('\n1. 帕累托分析(二八定律):\\n')];
for i = 1:length(item_results.pareto_results)
    pareto = item_results.pareto_results(i);
    report = [report sprintf('    - %s: 头部 20%%单品贡献%.2f%%销量\\n', ...
        pareto.category, pareto.pareto_ratio*100)];
end

% 损耗率分析
report = [report sprintf('\n2. 损耗率与销量相关性: %.4f\\n',
item_results.loss_sales_corr)];
if item_results.loss_sales_corr < -0.3
    report = [report sprintf('    结论: 损耗率与销量呈显著负相关关系\\n')];
end

```

```

elseif item_results.loss_sales_corr > 0.3
    report = [report sprintf('    结论：损耗率与销量呈显著正相关关系\n')];
else
    report = [report sprintf('    结论：损耗率与销量无显著相关关系\n')];
end

% 保存报告
fid = fopen('蔬菜销售分析报告.txt', 'w', 'n', 'UTF-8');
fprintf(fid, '%s', report);
fclose(fid);

disp(report);

```

end

第二问求解代码

```

function main2()

    % 主函数：执行问题二的完整分析流程
    disp('开始蔬菜销售分析与优化...');

    % 1. 数据加载与预处理
    [sales_data, item_info] = load_and_preprocess_data();

    % 2. 分析销售总量与成本加成定价的关系
    [elasticity_results, daily_data] =
analyze_price_quantity_relationship(sales_data);

    % 3. 需求预测

```

```
        demand_forecasts = demand_forecasting(daily_data,  
elasticity_results);
```

```
% 4. 优化模型
```

```
        optimization_results = optimize_profit(elasticity_results,  
demand_forecasts, daily_data);
```

```
% 5. 结果可视化与输出
```

```
        final_results = visualize_and_output_results(optimization_results,  
elasticity_results, demand_forecasts);
```

```
% 6. 输出最终决策建议
```

```
        output_final_recommendations(final_results, elasticity_results);
```

```
        disp('分析完成! 结果已保存到 CSV 文件和可视化图表中。');
```

```
end
```

```
function [sales_data, item_info] = load_and_preprocess_data()
```

```
% 加载并预处理数据, 整合四个附件
```

```
disp('正在加载数据...');
```

```
% 加载附件
```

```

[~, ~, item_info_raw] = xlsread('23.C. 附件 1.xlsx');
[~, ~, sales_data_raw] = xlsread('23.C. 附件 2.xlsx');
[~, ~, wholesale_price_raw] = xlsread('23.C. 附件 3.xlsx');
[~, ~, loss_rate_data_raw] = xlsread('23.C. 附件 4.xlsx');

% 提取表头和数据
item_header = item_info_raw(1, :);
item_data = item_info_raw(2:end, :);

sales_header = sales_data_raw(1, :);
sales_data_cell = sales_data_raw(2:end, :);

wholesale_header = wholesale_price_raw(1, :);
wholesale_data = wholesale_price_raw(2:end, :);

loss_header = loss_rate_data_raw(1, :);

loss_data = loss_rate_data_raw(2:end, :);

% 转换为结构数组
sales_data = struct();
for i = 1:size(sales_data_cell, 1)
    % 提取销售数据

```

```

sales_data(i).销售日期 = sales_data_cell{i, 1};
sales_data(i).单品编码 = sales_data_cell{i, 2};
sales_data(i).销量_千克 = sales_data_cell{i, 3};
sales_data(i).销售单价_元_千克 = sales_data_cell{i, 4};

% 添加商品信息
item_idx = find([item_data{:, 1}] == sales_data(i).单品编码);
if ~isempty(item_idx)
    sales_data(i).单品名称 = item_data{item_idx, 2};
    sales_data(i).分类编码 = item_data{item_idx, 3};
    sales_data(i).分类名称 = item_data{item_idx, 4};
end

% 添加批发价格
date_val = sales_data(i).销售日期;
if ischar(date_val)
    date_val = datenum(date_val);
end

price_idx = find([wholesale_data{:, 1}] == date_val &
[wholesale_data{:, 2}] == sales_data(i).单品编码);
if ~isempty(price_idx)
    sales_data(i).批发价格 = wholesale_data{price_idx, 3};
end

```

```

% 添加损耗率
loss_idx = find([loss_data{:, 1}] == sales_data(i).单品编码);
if ~isempty(loss_idx)
    sales_data(i).损耗率 = loss_data{loss_idx, 2};

else
    sales_data(i).损耗率 = NaN;
end

% 计算实际成本（考虑损耗）
if isfield(sales_data(i), '批发价格') && isfield(sales_data(i), '
损耗率') && ~isnan(sales_data(i).损耗率)
    sales_data(i).实际成本 = sales_data(i).批发价格 / (1 -
sales_data(i).损耗率/100);
else
    sales_data(i).实际成本 = NaN;
end

% 添加时间特征
if ischar(sales_data(i).销售日期)
    sales_date = datenum(sales_data(i).销售日期);
else
    sales_date = sales_data(i).销售日期;
end
sales_data(i).月份 = month(sales_date);
sales_data(i).季度 = quarter(sales_date);

```

```

        sales_data(i).星期 = weekday(sales_date) - 1;
        sales_data(i).是否周末 = sales_data(i).星期 >= 5;
    end

    % 处理损耗率缺失值
    % 计算各品类的平均损耗率
    category_loss = struct();
    for i = 1:length(sales_data)
        if isfield(sales_data(i), '分类名称') && isfield(sales_data(i), '
损耗率')

            cat_name = sales_data(i).分类名称;
            loss_val = sales_data(i).损耗率;

            if ~isnan(loss_val)

                if ~isfield(category_loss, cat_name)
                    category_loss.(cat_name) = [];
                end
                category_loss.(cat_name) = [category_loss.(cat_name)
loss_val];
            end
        end
    end

    % 计算平均损耗率

```

```

cat_names = fieldnames(category_loss);
for i = 1:length(cat_names)
    cat_name = cat_names{i};
    category_loss.(cat_name) = mean(category_loss.(cat_name));
end

% 填充缺失的损耗率
for i = 1:length(sales_data)
    if isfield(sales_data(i), '损耗率') && isnan(sales_data(i).损耗
率) && isfield(sales_data(i), '分类名称')
        cat_name = sales_data(i).分类名称;
        if isfield(category_loss, cat_name)
            sales_data(i).损耗率 = category_loss.(cat_name);

            % 重新计算实际成本
            if isfield(sales_data(i), '批发价格')
                sales_data(i).实际成本 = sales_data(i).批发价格 /
(1
- sales_data(i).损耗率/100);
            end
        end
    end
end

% 创建商品信息结构
item_info = struct();

```

```

for i = 1:size(item_data, 1)
    item_info(i).单品编码 = item_data{i, 1};
    item_info(i).单品名称 = item_data{i, 2};
    item_info(i).分类编码 = item_data{i, 3};
    item_info(i).分类名称 = item_data{i, 4};
end

```

```

disp('数据预处理完成!');
end

```

```

function [elasticity_results, daily_data] =
analyze_price_quantity_relationship(sales_data)
% 分析各蔬菜品类的销售总量与成本加成定价的关系
disp('正在分析销售总量与成本加成定价的关系...');

% 提取所有分类名称
categories = {};
for i = 1:length(sales_data)
    if isfield(sales_data(i), '分类名称')
        cat_name = sales_data(i).分类名称;
        if ~ismember(cat_name, categories)
            categories{end+1} = cat_name;
        end
    end
end
end

```

```

end

% 按日期和分类汇总销量、价格和成本
dates = [];
for i = 1:length(sales_data)
    if isfield(sales_data(i), '销售日期')
        if ischar(sales_data(i).销售日期)
            dates = [dates; datenum(sales_data(i).销售日期)];
        else
            dates = [dates; sales_data(i).销售日期];
        end
    end
end

end

end

unique_dates = unique(dates);

% 初始化每日数据矩阵
daily_sales = zeros(length(unique_dates), length(categories));
daily_prices = zeros(length(unique_dates), length(categories));
daily_costs = zeros(length(unique_dates), length(categories));
daily_loss_rates = zeros(length(unique_dates), length(categories));

% 填充每日数据矩阵
for i = 1:length(sales_data)
    if isfield(sales_data(i), '销售日期') && isfield(sales_data(i), '

```

```

分类名称') && ...

        isfield(sales_data(i), '销量_千克') && isfield(sales_data(i),
'销售单价_元_千克') && ...

        isfield(sales_data(i), '实际成本') && isfield(sales_data(i), '
损耗率')

    if ischar(sales_data(i).销售日期)
        date_val = datenum(sales_data(i).销售日期);
    else
        date_val = sales_data(i).销售日期;
    end

    date_idx = find(unique_dates == date_val);
    cat_idx = find(strcmp(categories, sales_data(i).分类名称));

    if ~isempty(date_idx) && ~isempty(cat_idx)
        daily_sales(date_idx, cat_idx) = daily_sales(date_idx,
cat_idx) + sales_data(i).销量_千克;
        daily_prices(date_idx, cat_idx) = daily_prices(date_idx,
cat_idx) + sales_data(i).销售单价_元_千克;
        daily_costs(date_idx, cat_idx) = daily_costs(date_idx,
cat_idx) + sales_data(i).实际成本;

        daily_loss_rates(date_idx, cat_idx) =
daily_loss_rates(date_idx, cat_idx) + sales_data(i).损耗率;

```

```

        end
    end
end

% 计算平均值
for i = 1:length(categories)
    non_zero_idx = daily_sales(:, i) > 0;
    if any(non_zero_idx)
        daily_prices(non_zero_idx, i) = daily_prices(non_zero_idx,
i) ./ daily_sales(non_zero_idx, i);
        daily_costs(non_zero_idx, i) = daily_costs(non_zero_idx,
i) ./
daily_sales(non_zero_idx, i);
        daily_loss_rates(non_zero_idx, i) =
daily_loss_rates(non_zero_idx, i) ./ daily_sales(non_zero_idx, i);
    end
end

% 计算成本加成率：(价格-成本)/成本
daily_margin = (daily_prices - daily_costs) ./ daily_costs;

% 按品类分析价格弹性
elasticity_results = struct();

figure;

```

```

set(gcf, 'Position', [100, 100, 1200, 800]);

for i = 1:length(categories)
    cat_name = categories{i};
    cat_sales = daily_sales(:, i);
    cat_prices = daily_prices(:, i);

    % 移除零值
    valid_idx = cat_sales > 0 & cat_prices > 0;

    cat_sales = cat_sales(valid_idx);
    cat_prices = cat_prices(valid_idx);

    if length(cat_sales) > 10          % 确保有足够的数据点
        % 取对数进行线性回归
        log_price = log(cat_prices);
        log_quantity = log(cat_sales);

        % 线性回归
        X = [ones(size(log_price)), log_price];
        [b, bint, r, rint, stats] = regress(log_quantity, X);

        slope = b(2);
    end
end

```

```

intercept = b(1);
r_squared = stats(1);
p_value = stats(3);

elasticity = -slope; % 价格弹性系数

% 计算基准需求常数 a = e^intercept
base_demand_constant = exp(intercept);

elasticity_results.(strrep(cat_name, ' ', '_')) = struct(...
    'elasticity', elasticity, ...
    'r_squared', r_squared, ...
    'p_value', p_value, ...
    'base_demand_constant', base_demand_constant, ...
    'intercept', intercept ...
);

% 绘制子图
subplot(2, 3, i);
scatter(cat_prices, cat_sales, 50, 'filled',
'MarkerFaceAlpha', 0.6);

% 绘制拟合曲线

```

```

x_range = linspace(min(cat_prices), max(cat_prices), 100);
y_range = base_demand_constant * x_range.^(-elasticity);
hold on;
plot(x_range, y_range, 'r-', 'LineWidth', 2);

title(sprintf('%s: 价格-销量关系', cat_name));
xlabel('销售价格(元/千克)');
ylabel('销量(千克)');
legend(sprintf('弹性系数: %.2f\n 基准需求常数: %.2f',
elasticity, base_demand_constant), 'Location', 'best');

grid on;
else
    elasticity_results.(strrep(cat_name, ' ', '_')) = struct(...
        'elasticity', NaN, ...
        'r_squared', NaN, ...
        'p_value', NaN, ...
        'base_demand_constant', NaN, ...
        'intercept', NaN ...
    );
end
end

saveas(gcf, '价格弹性分析.png');
close;

% 输出弹性系数和基准需求常数结果

```

```

disp(' 各品类价格弹性系数和基准需求常数:');
cat_fields = fieldnames(elasticity_results);
for i = 1:length(cat_fields)
    cat_name = strrep(cat_fields{i}, '_ ', ' ');
    result = elasticity_results.(cat_fields{i});
    if ~isnan(result.elasticity)
        fprintf('%s: 弹性系数=%.3f, 基准需求常数=%.2f (R^2=%.3f,
p=%.3f)\n', ...

        cat_name, result.elasticity,
result.base_demand_constant, result.r_squared, result.p_value);
    else
        fprintf('%s: 数据不足, 无法计算弹性系数\n', cat_name);
    end
end

% 创建每日数据结构
daily_data = struct();
for i = 1:length(unique_dates)
    daily_data(i).销售日期 = unique_dates(i);
    for j = 1:length(categories)
        daily_data(i).(strrep(categories{j}, '_ ', '_ ')) =
daily_sales(i, j);
    end
end

end

function demand_forecasts = demand_forecasting(daily_data,

```

```

elasticity_results)

% 建立需求预测模型，预测未来一周的需求
disp(' 正在进行需求预测...');

% 提取时间序列数据
categories = fieldnames(elasticity_results);
dates = [daily_data.销售日期];

% 预测未来一周（2023 年 7 月 1-7 日）
future_dates = datenum('2023-07-01'):datenum('2023-07-07');
demand_forecasts = struct();

for i = 1:length(categories)
    cat_name = categories{i};
    cat_sales = [];

    % 提取该品类的销量数据

    for j = 1:length(daily_data)
        if isfield(daily_data(j), cat_name)
            cat_sales = [cat_sales; daily_data(j).(cat_name)];
        end
    end
end

```

```

% 使用简单指数平滑法进行预测
alpha = 0.3;      % 平滑参数
forecast = simple_exponential_smoothing(cat_sales, alpha, 7);

% 确保预测值非负
forecast = max(forecast, 0);

% 存储预测结果
for j = 1:length(future_dates)
    date_str = datestr(future_dates(j), 'yyyy-mm-dd');
    demand_forecasts.(cat_name).(strrep(date_str, '-', '_')) =
forecast(j);
end
end
end

```

```

function forecast = simple_exponential_smoothing(data, alpha, steps)

```

```

% 简单指数平滑法

```

```

if isempty(data)

```

```

    forecast = zeros(1, steps);

```

```

    return;

```

```

end

```

```

% 初始化

```

```

smoothed = zeros(size(data));
smoothed(1) = data(1);

% 计算平滑值
for i = 2:length(data)

    smoothed(i) = alpha * data(i) + (1 - alpha) * smoothed(i-1);
end

% 预测未来值
forecast = zeros(1, steps);
for i = 1:steps
    if i == 1
        forecast(i) = alpha * data(end) + (1 - alpha) *
smoothed(end);
    else
        forecast(i) = forecast(i-1); % 简单指数平滑的预测是恒定的
    end
end
end

function results = optimize_profit(elasticity_results, demand_forecasts,
daily_data)
    % 构建优化模型，最大化商超收益
    disp('正在优化补货与定价策略...');

```

```

% 获取各品类的平均成本和损耗率
avg_costs = struct();
loss_rates = struct();

categories = fieldnames(elasticity_results);
for i = 1:length(categories)
    cat_name = categories{i};
    cat_costs = [];
    cat_loss_rates = [];

    % 从每日数据中提取成本和损耗率
    for j = 1:length(daily_data)
        if isfield(daily_data(j), '销售日期')
            % 这里需要根据实际数据结构调整
            % 假设 daily_data 中已经包含了成本和损耗率信息
            % 实际实现中可能需要从原始销售数据中提取

            cat_costs = [cat_costs; 10]; % 示例值，需要替换为实际计
算
            cat_loss_rates = [cat_loss_rates; 0.1]; % 示例值，需要
替换为实际计算
        end
    end
end

```

```

    if ~isempty(cat_costs)
        avg_costs.(cat_name) = mean(cat_costs);
        loss_rates.(cat_name) = mean(cat_loss_rates);
    else
        avg_costs.(cat_name) = 10;    % 默认值
        loss_rates.(cat_name) = 0.1;  % 默认值
    end
end

% 优化每个品类每天的补货量和加成率
results = struct();
future_dates = datenum('2023-07-01'):datenum('2023-07-07');

for i = 1:length(categories)
    cat_name = categories{i};
    results.(cat_name) = struct();

    for j = 1:length(future_dates)
        date_str = datestr(future_dates(j), 'yyyy-mm-dd');
        date_field = strrep(date_str, '-', '_');

        % 初始猜测：补货量=预测需求，加成率=0.3
        x0 = [demand_forecasts.(cat_name).(date_field), 0.3];
    end
end

```

```

% 约束条件：补货量 >= 0, 加成率 >= 0.1, <= 0.8
lb = [0, 0.1];
ub = [Inf, 0.8];

% 优化
options = optimset('Display', 'off', 'Algorithm',
'interior-point');

[x, fval, exitflag] = fmincon(@(x) profit_function(x,
cat_name, date_str, elasticity_results, demand_forecasts, avg_costs,
loss_rates), ...
x0, [], [], [], [], lb, ub, [], options);

if exitflag > 0
    optimal_quantity = x(1);
    optimal_margin = x(2);
    optimal_price = avg_costs.(cat_name) * (1 +
optimal_margin);

    results.(cat_name).(date_field) = struct(...
        '补货量', optimal_quantity, ...
        '加成率', optimal_margin, ...
        '定价', optimal_price ...
    );
else

```

```

        % 如果优化失败，使用简单策略
        results.(cat_name).(date_field) = struct(...
            '补货量',
demand_forecasts.(cat_name).(date_field), ...
            '加成率', 0.3, ...
            '定价', avg_costs.(cat_name) * 1.3 ...
        );
    end
end
end
end
end

```

```

function profit = profit_function(x, category, date, elasticity_results,
demand_forecasts, avg_costs, loss_rates)

```

```

    % 计算给定品类和日期的利润函数

```

```

    % x(1): 补货量

```

```

    % x(2): 加成率

```

```

    % 获取基础参数

```

```

    cat_field = strrep(category, ' ', '_');

```

```

    date_field = strrep(date, '-', '_');

```

```

    elasticity = elasticity_results.(cat_field).elasticity;

```

```

    base_demand = demand_forecasts.(cat_field).(date_field);

```

```
cost = avg_costs.(cat_field);
loss_rate = loss_rates.(cat_field);
```

```
% 计算价格
```

```
price = cost * (1 + x(2));
```

```
% 计算需求（考虑价格弹性）
```

```
demand = base_demand * (price / (cost * (1 + 0.3))) ^ (-elasticity); %
```

0.3 是基准加成率

```
% 实际销量不能超过补货量考虑损耗后的可售量
```

```
actual_sales = min(demand, x(1) * (1 - loss_rate));
```

```
% 计算利润
```

```
revenue = actual_sales * price;
```

```
cost_total = x(1) * cost; % 总成本
```

```
profit = -(revenue - cost_total); % 负值因为我们要最小化负利润
```

```
end
```

```
function result_df = visualize_and_output_results(optimization_results,
elasticity_results, demand_forecasts)
```

```
% 可视化优化结果并输出决策表格
```

```

disp('生成结果与可视化...');

% 创建结果数据

categories = fieldnames(optimization_results);
future_dates = datenum('2023-07-01'):datenum('2023-07-07');

result_list = [];
for i = 1:length(categories)
    cat_name = categories{i};
    cat_field = strrep(cat_name, ' ', '_');

    for j = 1:length(future_dates)
        date_str = datestr(future_dates(j), 'yyyy-mm-dd');
        date_field = strrep(date_str, '-', '_');

        result = optimization_results.(cat_field).(date_field);
        forecast = demand_forecasts.(cat_field).(date_field);

        result_list = [result_list; {date_str, cat_name, result.补货
量, result.定价, result.加成率, forecast}];
    end
end
end

```

```

% 创建结果表格

result_df = cell2table(result_list, 'VariableNames', {'日期', '品类',
'补货量_kg', '定价_元_kg', '加成率', '预测需求'});

% 输出结果到 CSV

writetable(result_df, '未来一周补货定价策略.csv');

% 可视化结果 - 补货策略

figure;

for i = 1:length(categories)

    cat_name = categories{i};

    cat_data = result_df(strcmp(result_df.品类, cat_name), :);

    subplot(2, 3, i);

    plot(denum(cat_data.日期), cat_data.补货量_kg, 'o-',

'LineWidth', 2);

    hold on;

    plot(denum(cat_data.日期), cat_data.预测需求, 's--',

'LineWidth', 2);

    title(sprintf('%s 补货策略', cat_name));

```

```

        xlabel('日期');
        ylabel('数量(kg)');
        legend('补货量', '预测需求', 'Location', 'best');
        datetick('x', 'mm-dd');
        grid on;
    end

    saveas(gcf, '补货策略可视化.png');
    close;

% 可视化结果 - 定价策略
figure;
for i = 1:length(categories)
    cat_name = categories{i};
    cat_data = result_df(strcmp(result_df.品类, cat_name), :);

    subplot(2, 3, i);
    plot(denum(cat_data.日期), cat_data.定价_元_kg, 'o-',
'LineWidth', 2);

    title(sprintf('%s 定价策略', cat_name));
    xlabel('日期');
    ylabel('价格(元/kg)');
    datetick('x', 'mm-dd');
    grid on;
end

```

```

end

saveas(gcf, '定价策略可视化.png');
close;

% 输出弹性系数与加成率的关系
elasticity_values = [];
margin_values = [];
category_names = {};

for i = 1:length(categories)
    cat_name = categories{i};
    cat_field = strrep(cat_name, ' ', '_');

    if isfield(elasticity_results, cat_field) &&
~isnan(elasticity_results.(cat_field).elasticity)
        cat_data = result_df(strcmp(result_df.品类, cat_name), :);

        elasticity_values = [elasticity_values;
elasticity_results.(cat_field).elasticity];
        margin_values = [margin_values; mean(cat_data.加成率)];
        category_names{end+1} = cat_name;
    end
end
end

```

```

figure;
scatter(elasticity_values, margin_values, 100, 'filled');

% 添加品类标签
for i = 1:length(category_names)
    text(elasticity_values(i), margin_values(i), category_names{i},
'FontSize', 8);
end

xlabel('价格弹性系数');
ylabel('平均加成率');
title('价格弹性与加成率关系');
grid on;
saveas(gcf, '价格弹性与加成率关系.png');

close;

% 输出基准需求常数与弹性系数的关系
base_demand_values = [];
for i = 1:length(categories)
    cat_name = categories{i};
    cat_field = strrep(cat_name, ' ', '_');

```

```

        if isfield(elasticity_results, cat_field) &&
~isnan(elasticity_results.(cat_field).elasticity)
            base_demand_values = [base_demand_values;
elasticity_results.(cat_field).base_demand_constant];
        end
    end

figure;
scatter(elasticity_values, base_demand_values, 100, 'filled');

% 添加品类标签
for i = 1:length(category_names)
    text(elasticity_values(i), base_demand_values(i),
category_names{i}, 'FontSize', 8);
end

xlabel('价格弹性系数');
ylabel('基准需求常数');
title('价格弹性与基准需求常数关系');
grid on;
saveas(gcf, '价格弹性与基准需求常数关系.png');
close;
end

```

```

function output_final_recommendations(final_results, elasticity_results)
    % 输出最终决策建议
    disp('=====');

    disp('最终决策建议');
    disp('=====');

    % 计算总补货量和预期收益
    total_quantity = sum(final_results.补货量_kg);
    total_revenue = sum(final_results.补货量_kg .* final_results.定价_元
_kg);
    avg_margin = mean(final_results.加成率) * 100;

    fprintf('未来一周总补货量: %.2f kg\n', total_quantity);
    fprintf('预期总销售收入: %.2f 元\n', total_revenue);
    fprintf('平均加成率: %.2f%%\n', avg_margin);

    % 按品类输出建议
    categories = unique(final_results.品类);
    fprintf('\n 各品类策略摘要:\n');
    for i = 1:length(categories)
        cat_name = categories{i};
        cat_data = final_results(strcmp(final_results.品类,
cat_name), :);
        avg_quantity = mean(cat_data.补货量_kg);
    end
end

```

```

    avg_price = mean(cat_data.定价_元_kg);
    avg_margin = mean(cat_data.加成率) * 100;

    % 获取基准需求常数和弹性系数
    cat_field = strrep(cat_name, ' ', '_');
    if isfield(elasticity_results, cat_field)
        base_demand =
elasticity_results.(cat_field).base_demand_constant;
        elasticity = elasticity_results.(cat_field).elasticity;

        fprintf('s: 日均补货%.1fkg, 均价%.2f 元/kg, 加成
率%.1f%%\n', ...
            cat_name, avg_quantity, avg_price, avg_margin);
        fprintf('        需求函数:  $Q = %.2f \times P^{(-.2f)}$ \n',
            base_demand, elasticity);
    else
        fprintf('s: 日均补货%.1fkg, 均价%.2f 元/kg, 加成
率%.1f%%\n', ...
            cat_name, avg_quantity, avg_price, avg_margin);
    end
end

% 输出关键洞察
fprintf('\n 关键洞察:\n');

```

```

categories = fieldnames(elasticity_results);
for i = 1:length(categories)
    cat_name = strrep(categories{i}, '_', ' ');
    result = elasticity_results.(categories{i});

    if ~isnan(result.elasticity)
        if result.elasticity > 1
            fprintf(' - %s 需求富有弹性 ( $\epsilon = %.2f$ ), 基准需求常数=%.2f,
适合低价促销策略\n', ...
                cat_name, result.elasticity,
result.base_demand_constant);
        else
            fprintf(' - %s 需求缺乏弹性 ( $\epsilon = %.2f$ ), 基准需求常数=%.2f,
适合较高加成率策略\n', ...
                cat_name, result.elasticity,
result.base_demand_constant);
        end
    end
end

% 输出完整的需求函数公式
fprintf('\n 完整需求函数公式:\n');
for i = 1:length(categories)
    cat_name = strrep(categories{i}, '_', ' ');
    result = elasticity_results.(categories{i});

```

```

        if ~isnan(result.elasticity)
            fprintf(' %s: Q = %.2f × P^(-%.2f)\n', ...
                cat_name, result.base_demand_constant,
result.elasticity);
        end
    end
end

```

第三问求解代码

```

function main3()
    % 主函数：执行问题三的完整分析流程
    disp(' 开始蔬菜单品补货定价策略分析...');

    % 1. 数据加载与预处理
    [sales_data, item_info] = load_and_preprocess_data();

    % 2. 筛选 6 月 24-30 日有销售的单品
    [recent_sales, sold_items] = filter_recent_items(sales_data);

    % 3. 计算单品相关参数
    item_params = calculate_item_parameters(recent_sales, sold_items);

    % 4. 从问题二结果中获取品类弹性系数和加成率
    [category_elasticity, category_markup] = get_category_parameters();

```

```

    % 5. 计算单品需求预测和最优定价（使用第二问的加成率）
    item_params = calculate_demand_and_pricing(item_params,
category_elasticity, category_markup);

    % 6. 显示需求预测结果
    demand_df = display_demand_prediction(item_params);

    % 7. 选择单品并分配补货量
    results = select_items_and_allocate_quantity(item_params);

    % 8. 结果分析与可视化
    results_df = analyze_and_visualize_results(results);

    disp('分析完成!');
end

function [sales_data, item_info] = load_and_preprocess_data()
    % 加载并预处理数据，整合四个附件
    disp('正在加载数据...');

    % 加载附件
    [~, ~, item_info_raw] = xlsread('23.C.附件 1.xlsx');
    [~, ~, sales_data_raw] = xlsread('23.C.附件 2.xlsx');

```

```
[~, ~, wholesale_price_raw] = xlsread('23.C. 附件 3.xlsx');  
[~, ~, loss_rate_data_raw] = xlsread('23.C. 附件 4.xlsx');
```

```
% 提取表头和数据
```

```
item_header = item_info_raw(1, :);  
item_data = item_info_raw(2:end, :);
```

```
sales_header = sales_data_raw(1, :);  
sales_data_cell = sales_data_raw(2:end, :);
```

```
wholesale_header = wholesale_price_raw(1, :);  
wholesale_data = wholesale_price_raw(2:end, :);
```

```
loss_header = loss_rate_data_raw(1, :);  
loss_data = loss_rate_data_raw(2:end, :);
```

```
% 转换为结构数组
```

```
sales_data = struct();  
for i = 1:size(sales_data_cell, 1)  
    % 提取销售数据  
    sales_data(i).销售日期 = sales_data_cell{i, 1};  
  
    sales_data(i).单品编码 = sales_data_cell{i, 2};
```

```

sales_data(i).销量_千克 = sales_data_cell{i, 3};
sales_data(i).销售单价_元_千克 = sales_data_cell{i, 4};

```

```

% 添加商品信息

```

```

item_idx = find([item_data{:, 1}] == sales_data(i).单品编码);
if ~isempty(item_idx)
    sales_data(i).单品名称 = item_data{item_idx, 2};
    sales_data(i).分类编码 = item_data{item_idx, 3};
    sales_data(i).分类名称 = item_data{item_idx, 4};
end

```

```

% 添加批发价格

```

```

date_val = sales_data(i).销售日期;
if ischar(date_val)
    date_val = datenum(date_val);
end

```

```

price_idx = find([wholesale_data{:, 1}] == date_val &
[wholesale_data{:, 2}] == sales_data(i).单品编码);
if ~isempty(price_idx)
    sales_data(i).批发价格 = wholesale_data{price_idx, 3};
end

```

```

% 添加损耗率

```

```

loss_idx = find([loss_data{:, 1}] == sales_data(i).单品编码);

```

```

    if ~isempty(loss_idx)
        sales_data(i).损耗率 = loss_data{loss_idx, 2};
    else
        sales_data(i).损耗率 = NaN;
    end

    % 计算实际成本（考虑损耗）
    if isfield(sales_data(i), '批发价格') && isfield(sales_data(i), '
损耗率') && ~isnan(sales_data(i).损耗率)

        sales_data(i).实际成本 = sales_data(i).批发价格 / (1 -
sales_data(i).损耗率/100);
    else
        sales_data(i).实际成本 = NaN;
    end
end

% 处理损耗率缺失值
% 计算各品类的平均损耗率
category_loss = struct();
for i = 1:length(sales_data)
    if isfield(sales_data(i), '分类名称') && isfield(sales_data(i), '
损耗率')

        cat_name = sales_data(i).分类名称;
        loss_val = sales_data(i).损耗率;
    end
end

```

```

        if ~isnan(loss_val)
            if ~isfield(category_loss, cat_name)
                category_loss.(cat_name) = [];
            end
            category_loss.(cat_name) = [category_loss.(cat_name)
loss_val];
        end
    end
end

% 计算平均损耗率
cat_names = fieldnames(category_loss);
for i = 1:length(cat_names)
    cat_name = cat_names{i};
    category_loss.(cat_name) = mean(category_loss.(cat_name));
end

% 填充缺失的损耗率
for i = 1:length(sales_data)

    if isfield(sales_data(i), '损耗率') && isnan(sales_data(i).损耗
率) && isfield(sales_data(i), '分类名称')
        cat_name = sales_data(i).分类名称;
        if isfield(category_loss, cat_name)
            sales_data(i).损耗率 = category_loss.(cat_name);
        end
    end
end

```

```

        % 重新计算实际成本
        if isfield(sales_data(i), '批发价格')
            sales_data(i).实际成本 = sales_data(i).批发价格
        / (1
        - sales_data(i).损耗率/100);
    end
end
end
end

% 创建商品信息结构
item_info = struct();
for i = 1:size(item_data, 1)
    item_info(i).单品编码 = item_data{i, 1};
    item_info(i).单品名称 = item_data{i, 2};
    item_info(i).分类编码 = item_data{i, 3};
    item_info(i).分类名称 = item_data{i, 4};
end

disp('数据预处理完成!');
end

```

```

function [recent_sales, sold_items] = filter_recent_items(sales_data)
    % 筛选 2023 年 6 月 24 日至 30 日有销售记录的单品

```

```
disp(' 筛选 6 月 24-30 日有销售记录的单品...');
```

```
% 筛选指定日期范围的数据
```

```
start_date = datenum('2023-06-24');
```

```
end_date = datenum('2023-06-30');
```

```
recent_sales = struct();
```

```
sold_items = [];
```

```
for i = 1:length(sales_data)
```

```
    if isfield(sales_data(i), '销售日期')
```

```
        if ischar(sales_data(i).销售日期)
```

```
            date_val = datenum(sales_data(i).销售日期);
```

```
        else
```

```
            date_val = sales_data(i).销售日期;
```

```
        end
```

```
        if date_val >= start_date && date_val <= end_date
```

```
            recent_sales = [recent_sales, sales_data(i)];
```

```
            if ~ismember(sales_data(i).单品编码, sold_items)
```

```
                sold_items = [sold_items, sales_data(i).单品编
```

```
码];
```

```
            end
```

```
        end
```

```

        end
    end

    % 移除第一个空元素
    if length(recent_sales) > 1
        recent_sales = recent_sales(2:end);
    end

    fprintf('6 月 24-30 日有销售记录的单品数量: %d\n',
length(sold_items));
    end

function item_params = calculate_item_parameters(recent_sales,
sold_items)

    % 计算每个单品的相关参数: 平均销量、平均价格、平均批发价、损耗率
    disp('计算单品参数...');

    item_params = struct();

    for i = 1:length(sold_items)
        item = sold_items(i);
        item_data = struct();
    end
end

```

```

% 提取该单品的数据
for j = 1:length(recent_sales)
    if recent_sales(j).单品编码 == item
        item_data = [item_data, recent_sales(j)];
    end
end

% 移除第一个空元素
if length(item_data) > 1
    item_data = item_data(2:end);
end

if ~isempty(item_data)
    % 计算平均销量
    total_sales = sum([item_data.销量_千克]);
    sales_days = length(unique([item_data.销售日期]));
    avg_sales = total_sales / sales_days;

    % 计算平均销售价格
    avg_price = mean([item_data.销售单价_元_千克]);

    % 计算平均批发价格
    avg_wholesale = mean([item_data.批发价格]);
end

```

```
% 获取损耗率
```

```
loss_rate = mean([item_data.损耗率]);
```

```
% 获取品类信息和单品名称
```

```
category = item_data(1).分类名称;
```

```
category_code = item_data(1).分类编码;
```

```
item_name = item_data(1).单品名称;
```

```
% 存储参数
```

```
item_params(i).单品编码 = item;
```

```
item_params(i).单品名称 = item_name;
```

```
item_params(i).分类名称 = category;
```

```
item_params(i).分类编码 = category_code;
```

```
item_params(i).平均销量 = avg_sales;
```

```
item_params(i).平均价格 = avg_price;
```

```
item_params(i).平均批发价 = avg_wholesale;
```

```
item_params(i).损耗率 = loss_rate;
```

```
end
```

```
end
```

```
end
```

```
function [category_elasticity, category_markup] =
```

```
get_category_parameters()
```

```
% 从问题二的结果中获取各品类的价格弹性系数和加成率
```

```

category_elasticity = struct(...
    '花叶类', 0.38, ...
    '花菜类', 0.716, ...
    '水生根茎类', 1.16, ...
    '茄类', 0.476, ...
    '辣椒类', 0.333, ...
    '食用菌', 0.988 ...
);

% 从第二问中得到的各品类加成率
category_markup = struct(...
    '花叶类', 0.8, ...           % 80%的加成率
    '花菜类', 0.729, ...        % 72.9%的加成率
    '水生根茎类', 0.496, ...    % 49.6%的加成率
    '茄类', 0.657, ...          % 65.7%的加成率
    '辣椒类', 0.657, ...        % 65.7%的加成率
    '食用菌', 0.8 ...           % 80%的加成率

);

end

```

```

function item_params = calculate_demand_and_pricing(item_params,
category_elasticity, category_markup)

% 计算每个单品的需求预测和最优定价
% 使用从第二问中得到的各品类具体加成率
disp('计算单品需求预测和最优定价...');

```

```

for i = 1:length(item_params)
    category = item_params(i).分类名称;

    % 获取弹性系数和加成率
    if isfield(category_elasticity, category)
        elasticity = category_elasticity.(category);
    else
        elasticity = 1.0; % 默认弹性为 1.0
    end

    if isfield(category_markup, category)
        markup = category_markup.(category);
    else
        markup = 0.5; % 默认加成率为 50%
    end

    % 计算基准需求  $a_i = \text{avg\_sales} * (\text{avg\_price})^{\text{elasticity}}$ 
    avg_sales = item_params(i).平均销量;
    avg_price = item_params(i).平均价格;
    a_i = avg_sales * (avg_price ^ elasticity);

    % 计算成本（考虑损耗）
    avg_wholesale = item_params(i).平均批发价;

```

```

loss_rate = item_params(i).损耗率;
cost = avg_wholesale / (1 - loss_rate/100);

% 计算最优价格（使用第二问的加成率）
if elasticity > 1
    % 富有弹性，使用公式  $P^* = [\epsilon / (\epsilon - 1)] * C$ 
    optimal_price = (elasticity / (elasticity - 1)) * cost;
else
    % 缺乏弹性，使用第二问得到的品类加成率
    optimal_price = (1 + markup) * cost;
end

% 计算预测需求  $D_i^* = a_i * (P_i^*)^{(-\epsilon)}$ 
predicted_demand = a_i * (optimal_price ^ (-elasticity));

% 计算预期利润
expected_profit = predicted_demand * optimal_price -
(predicted_demand / (1 - loss_rate/100)) * avg_wholesale;

% 更新参数
item_params(i).弹性系数 = elasticity;
item_params(i).加成率 = markup;
item_params(i).基准需求 = a_i;
item_params(i).成本 = cost;

```

```

        item_params(i).最优价格 = optimal_price;
        item_params(i).预测需求 = predicted_demand;
        item_params(i).预期利润 = expected_profit;
    end
end

```

```

function demand_df = display_demand_prediction(item_params)
    % 显示每个单品的需求预测结果
    disp('需求预测结果:');

```

```

disp('=====
=====');

```

```

    % 创建结果表格

```

```

result_list = {};
for i = 1:length(item_params)
    result_list{i, 1} = item_params(i).单品编码;
    result_list{i, 2} = item_params(i).单品名称;
    result_list{i, 3} = item_params(i).分类名称;
    result_list{i, 4} = item_params(i).平均销量;
    result_list{i, 5} = item_params(i).预测需求;
    result_list{i, 6} = item_params(i).最优价格;
    result_list{i, 7} = item_params(i).加成率;
    result_list{i, 8} = item_params(i).预期利润;

```

```
end
```

```
% 转换为表格
```

```
demand_df = cell2table(result_list, 'VariableNames', {'单品编码', '单品名称', '分类名称', '平均销量', '预测需求', '最优价格', '加成率', '预期利润'});
```

```
% 按预期利润降序排序
```

```
[~, idx] = sort([item_params.预期利润], 'descend');  
demand_df = demand_df(idx, :);
```

```
% 显示前 20 个和后 20 个单品
```

```
disp('预期利润最高的 20 个单品:');  
disp(demand_df(1:min(20, height(demand_df)), :));
```

```
disp('预期利润最低的 20 个单品:');  
disp(demand_df(max(1, end-19):end, :));
```

```
% 统计预测需求小于 2.5kg 的单品数量
```

```
low_demand_count = sum([item_params.预测需求] < 2.5);  
total_count = length(item_params);
```

```
fprintf('\n 预测需求小于 2.5kg 的单品数量: %d/%d (%.2f%%)\n',
```

```
low_demand_count, total_count, low_demand_count/total_count*100);
```

```
% 显示平均销量的统计信息
```

```
avg_sales = [item_params.平均销量];
```

```
fprintf('\n 平均销量统计:\n');
```

```
fprintf('最小值: %.2f kg\n', min(avg_sales));
```

```
fprintf('最大值: %.2f kg\n', max(avg_sales));
```

```
fprintf('平均值: %.2f kg\n', mean(avg_sales));
```

```
fprintf('中位数: %.2f kg\n', median(avg_sales));
```

```
% 显示加成率的统计信息
```

```
markup_rates = [item_params.加成率];
```

```
fprintf('\n 加成率统计:\n');
```

```
fprintf('最小值: %.2f%%\n', min(markup_rates)*100);
```

```
fprintf('最大值: %.2f%%\n', max(markup_rates)*100);
```

```
fprintf('平均值: %.2f%%\n', mean(markup_rates)*100);
```

```
fprintf('中位数: %.2f%%\n', median(markup_rates)*100);
```

```
end
```

```
function results = select_items_and_allocate_quantity(item_params,  
min_items, max_items, min_display)
```

```
% 选择单品并分配补货量
```

```
if nargin < 2
```

```
    min_items = 27;
```

```
end
```

```

if nargin < 3
    max_items = 33;
end
if nargin < 4
    min_display = 2.5;
end

disp(' 选择单品并分配补货量...');

% 按品类分组
categories = {};
for i = 1:length(item_params)

    category = item_params(i).分类名称;
    if ~ismember(category, categories)
        categories{end+1} = category;
    end
end

category_items = struct();
for i = 1:length(categories)
    category = categories{i};
    category_items.(strrep(category, ' ', '_')) = [];
end

```

```

for i = 1:length(item_params)
    category = item_params(i).分类名称;
    category_field = strrep(category, ' ', '_');
    category_items.(category_field) =
[category_items.(category_field), i];
end

% 计算每个品类的总预测需求
category_demand = struct();
for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');

    total_demand = 0;
    for j = 1:length(category_items.(category_field))
        idx = category_items.(category_field)(j);
        total_demand = total_demand + item_params(idx).预测需求;
    end

    category_demand.(category_field) = total_demand;
end

% 计算每个品类的权重（基于预测需求）

```

```

total_overall_demand = 0;
for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');
    total_overall_demand = total_overall_demand +
category_demand.(category_field);
end

```

```

category_weights = struct();
for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');
    category_weights.(category_field) =
category_demand.(category_field) / total_overall_demand;
end

```

```

% 计算每个品类应选择的单品数量
category_counts = struct();
for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');

    % 基于权重计算数量
    count = round(category_weights.(category_field) * (min_items +
max_items) / 2);

```

```

    % 确保至少选择 1 个单品
    count = max(1, count);

    % 确保不超过该品类的单品总数
    count = min(count, length(category_items.(category_field)));

    category_counts.(category_field) = count;
end

% 调整总数量到目标范围
total_count = 0;
for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');
    total_count = total_count + category_counts.(category_field);
end

% 如果总数量不在目标范围内, 进行调整
while total_count < min_items
    % 找到权重最大的品类, 增加一个单品
    max_weight = 0;
    max_category = '';

```

```

for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');

    if category_weights.(category_field) > max_weight && ...
        category_counts.(category_field) <
length(category_items.(category_field))
        max_weight = category_weights.(category_field);
        max_category = category_field;
    end
end

if ~isempty(max_category)
    category_counts.(max_category) =
category_counts.(max_category) + 1;
    total_count = total_count + 1;
else
    break;    % 所有品类都已达到最大单品数
end

end

while total_count > max_items

    % 找到权重最小的品类，减少一个单品
    min_weight = Inf;

```

```

min_category = '';
for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');

    if category_weights.(category_field) < min_weight &&
category_counts.(category_field) > 1
        min_weight = category_weights.(category_field);
        min_category = category_field;
    end
end

if ~isempty(min_category)
    category_counts.(min_category) =
category_counts.(min_category) - 1;
    total_count = total_count - 1;
else
    break;    % 所有品类都已达到最小单品数
end

end

% 在每个品类内选择单品（按预期利润排序）
selected_indices = [];
results = struct();

```

```

for i = 1:length(categories)
    category = categories{i};
    category_field = strrep(category, ' ', '_');

    % 获取该品类的所有单品索引
    item_indices = category_items.(category_field);

    % 按预期利润排序

profits = [item_params(item_indices).预期利润];
[~, sort_idx] = sort(profits, 'descend');
sorted_indices = item_indices(sort_idx);

% 选择前 N 个单品
count = category_counts.(category_field);
selected = sorted_indices(1:min(count, length(sorted_indices)));
selected_indices = [selected_indices, selected];

% 分配补货量（基于预测需求）
for j = 1:length(selected)
    idx = selected(j);

    % 计算补货量（考虑展示需求）

```

```

        replenishment = max(min_display, item_params(idx).预测需求);

        results(end+1).单品编码 = item_params(idx).单品编码;
        results(end).单品名称 = item_params(idx).单品名称;
        results(end).分类名称 = item_params(idx).分类名称;
        results(end).预测需求 = item_params(idx).预测需求;
        results(end).最优价格 = item_params(idx).最优价格;
        results(end).补货量 = replenishment;
        results(end).预期利润 = item_params(idx).预期利润;
    end
end

% 移除第一个空元素
if length(results) > 1
    results = results(2:end);
end

% 按预期利润排序
[~, idx] = sort([results.预期利润], 'descend');
results = results(idx);

fprintf('总共选择了 %d 个单品进行补货\n', length(results));
end

```

```

function results_df = analyze_and_visualize_results(results)

    % 分析结果并生成可视化
    disp('分析结果并生成可视化...');

    % 创建结果表格
    result_list = {};
    for i = 1:length(results)
        result_list{i, 1} = results(i).单品编码;
        result_list{i, 2} = results(i).单品名称;
        result_list{i, 3} = results(i).分类名称;
        result_list{i, 4} = results(i).预测需求;
        result_list{i, 5} = results(i).最优价格;
        result_list{i, 6} = results(i).补货量;
        result_list{i, 7} = results(i).预期利润;
    end

    results_df = cell2table(result_list, 'VariableNames', {'单品编码', '
单品名称', '分类名称', '预测需求', '最优价格', '补货量', '预期利润'});

    % 显示结果
    disp('最终选择的单品及补货策略:');

    disp('=====
=====');

```

```

disp(results_df);

% 计算总指标
total_demand = sum([results.预测需求]);
total_replenishment = sum([results.补货量]);
total_profit = sum([results.预期利润]);

fprintf('\n 总指标:\n');

fprintf('总预测需求: %.2f kg\n', total_demand);
fprintf('总补货量: %.2f kg\n', total_replenishment);
fprintf('总预期利润: %.2f 元\n', total_profit);
fprintf('平均单品补货量: %.2f kg\n', mean([results.补货量]));
fprintf('平均单品预期利润: %.2f 元\n', mean([results.预期利润]));

% 按品类统计
categories = unique({results.分类名称});
category_stats = struct();

for i = 1:length(categories)
    category = categories{i};
    category_items = results(strcmp({results.分类名称}, category));

```

```

category_stats(i).分类名称 = category;
category_stats(i).单品数量 = length(category_items);
category_stats(i).总预测需求 = sum([category_items.预测需求]);
category_stats(i).总补货量 = sum([category_items.补货量]);
category_stats(i).总预期利润 = sum([category_items.预期利润]);
category_stats(i).平均价格 = mean([category_items.最优价格]);
end

```

```

% 显示品类统计

```

```

fprintf('\n 按品类统计:\n');
fprintf('%-15s %-10s %-12s %-12s %-12s\n', '分类名称', '单品数量', '
总需求(kg)', '总补货(kg)', '总利润(元)');
for i = 1:length(category_stats)
    fprintf('%-15s %-10d %-12.2f %-12.2f %-12.2f\n', ...
        category_stats(i).分类名称, ...
        category_stats(i).单品数量, ...
        category_stats(i).总预测需求, ...
        category_stats(i).总补货量, ...
        category_stats(i).总预期利润);
end

```

```

% 生成可视化

```

```

figure('Position', [100, 100, 1200, 800]);

```

```

% 1. 各品类单品数量分布

```

```

subplot(2, 2, 1);
category_counts = [category_stats.单品数量];
pie(category_counts, categories);
title('各品类单品数量分布');

% 2. 各品类总补货量
subplot(2, 2, 2);
category_replenishment = [category_stats.总补货量];
bar(category_replenishment);
set(gca, 'XTickLabel', categories);
xtickangle(45);
title('各品类总补货量 (kg)');
ylabel('补货量 (kg)');

% 3. 各品类总预期利润
subplot(2, 2, 3);
category_profits = [category_stats.总预期利润];
bar(category_profits);
set(gca, 'XTickLabel', categories);
xtickangle(45);
title('各品类总预期利润 (元)');
ylabel('利润 (元)');

% 4. 单品补货量与预期利润散点图
subplot(2, 2, 4);
scatter([results.补货量], [results.预期利润], 50, 'filled');

```

```

xlabel('补货量 (kg)');
ylabel('预期利润 (元)');
title('单品补货量与预期利润关系');
grid on;

% 添加相关系数
correlation = corr([results.补货量], [results.预期利润]);
text(0.7, 0.9, sprintf('相关系数: %.3f', correlation), 'Units',
'normalized');

% 保存图表
saveas(gcf, '单品补货定价策略分析结果.png');

% 保存结果到 Excel
writetable(results_df, '单品补货定价策略结果.xlsx');
disp('结果已保存到"单品补货定价策略结果.xlsx"');

end

% 运行主函数
main3();

```

202503080008 基于北太天元软件的农作物种植问题的求解

作者：付正红 王康帅 冶荣桓

指导老师：王爱文

学校：北京信息科技大学

摘要：本文针对华北山区某乡村 2024 至 2030 年的农作物种植策略问题建立数学模型，分别从耕地类型和农作物两个角度展开分析，旨在实现耕地资源的优化利用与经济效益最大化。

问题一：在确定性环境下（销售量、种植成本、亩产量、销售价格均稳定），综合考虑耕地类型适配性（如平旱地适合粮食、大棚适合蔬菜）、轮作要求（每 3 年至少种植 1 次豆类）、重茬限制（禁止连续种植同一作物）、种植集中度（单作物单地块面积 $\geq 50\%$ ）等核心约束，针对《过剩滞销》和《过剩半价出售》两种销售场景，采用遗传规划算法求解。通过初始种群随机生成、交叉变异实现方案演化、约束修复保障可行性，最终得到两类场景的最优解：情况 1 在迭代 149 代后收敛，利润稳定在 6978352.166 元；情况 2 经 35 代迭代收敛，利润达 44536122.35 万元，为稳定市场下的种植决策提供了基准方案。

问题二：在农业生产中，各类不确定性因素显著影响种植决策，因此需将其纳入模型考量：小麦与玉米的销量呈增长态势，年增长率维持在 5%至 10%区间；其他作物的销量则存在 $\pm 5\%$ 的波动。同时，气候因素导致农作物亩产量出现 $\pm 10\%$ 的变化；种植成本每年平均增长 5%；蔬菜类产品价格以年均 5%的幅度上涨，食用菌（除羊肚菌）价格则每年下降 1%至 5%，羊肚菌的销售价格每年下降 5%。采用模拟退火算法，求解 2024-2030 年的最优种植方案，该算法能在波动因素带来的复杂解空间中有效避免局部最优，为生产者在不确定环境下的种植决策提供更优依据。

问题三：在问题 2 基础上，进一步整合农作物替代互补性与变量相关性：通过相关性分析明确作物交互关系（如生菜与西红柿互补，相关系数 0.98；莜麦与红豆互斥，相关系数-0.85），以及变量关联特征（销量与成本正相关系数 0.82，销量与价格负相关系数-0.35）。采用 NSGA-II 算法进行多目标优化，通过非支配排序区分方案优劣，拥挤度距离保障解集多样性，最终输出的最优策略不仅满足替代互补约束（互补作物面积比例 0.5-2.0），还实现了变量关联下的资源协同利用。获得 2024 至 2030 年情况一最优利润为 7757079.188，收益相对于问题二中有所提高，通过互补作物搭配提升了土地利用率，通过替代作物灵活调整应对市场波动，田间管理效率与实际可行性显著提升，为乡村长期可持续种植规划提供了科学且具操作性的决策支持。

关键词：农作物种植策略；遗传规划算法；模拟退火；NSGA-II；多目标优化

点 评：本文针对农作物种植策略优化这一经典运筹学问题，构建了从确定性到不确定

性再到多目标优化的三级递进模型体系，结构清晰、层次分明。方法选择上，遗传规划算法、模拟退火算法和 NSGA-II 分别对应三个子问题的求解需求，算法匹配合理。特别是问题三中引入作物替代互补性（如生菜与西红柿互补、苜蓿与红豆互斥），体现了对农业生产实际规律的深入理解。在北太天元使用方面，充分利用了其矩阵运算、数据可视化和多种优化算法支持能力。不足之处在于，三种算法均为通用型启发式算法，针对农作物种植问题的领域知识融入不足，可考虑设计问题特定的约束处理机制来提高求解效率。

基于北太天元软件的农作物种植问题的求解

摘要

本文针对华北山区某乡村 2024 至 2030 年的农作物种植策略问题建立数学模型^{[1][2]}，分别从耕地类型和农作物两个角度展开分析，旨在实现耕地资源的优化利用与经济效益最大化。

问题一：在确定性环境下（销售量、种植成本、亩产量、销售价格均稳定），综合考虑耕地类型适配性（如平旱地适合粮食、大棚适合蔬菜）、轮作要求（每 3 年至少种植 1 次豆类）、重茬限制（禁止连续种植同一作物）、种植集中度（单作物单地块面积 $\geq 50\%$ ）等核心约束，针对“过剩滞销”和“过剩半价出售”两种销售场景，采用遗传规划算法求解。通过初始种群随机生成、交叉变异实现方案演化、约束修复保障可行性，最终得到两类场景的最优解：情况 1 在迭代 149 代后收敛，利润稳定在 6978352.166 元；情况 2 经 35 代迭代收敛，利润达 44536122.35 万元，为稳定市场下的种植决策提供了基准方案。

问题二：在农业生产中，各类不确定性因素显著影响种植决策，因此需将其纳入模型考量：小麦与玉米的销量呈增长态势，年增长率维持在 5% 至 10% 区间；其他作物的销量则存在 $\pm 5\%$ 的波动。同时，气候因素导致农作物亩产量出现 $\pm 10\%$ 的变化；种植成本每年平均增长 5%；蔬菜类产品价格以年均 5% 的幅度上涨，食用菌（除羊肚菌）价格则每年下降 1% 至 5%，羊肚菌的销售价格每年下降 5%。采用模拟退火算法，通过设定初始温度 1000、衰减系数 0.9，模拟“高温阶段广泛探索解空间、低温阶段精细收敛至优解”的过程，从而在波动环境中实现利润与风险的平衡，最终得到两种情况的最佳种植方案，情况 1 最终的利润稳定在 6518732.835 元，情况 2 的利润稳定在 21420834.87 元。

问题三：在问题 2 基础上，进一步整合农作物替代互补性与变量相关性：通过相关性分析明确作物交互关系（如生菜与西红柿互补，相关系数 0.98；莜麦与红豆互斥，相关系数-0.85），以及变量关联特征（销量与成本正相关系数 0.82，销量与价格负相关系数-0.35）。采用 NSGA-II 算法进行多目标优化，通过非支配排序区分方案优劣，拥挤度距离保障解集多样性，最终输出的最优策略不仅满足替代互补约束（互补作物面积比例 0.5-2.0），还实现了变量关联下的资源协同利用。获得 2024 至 2030 年情况一最优利润为 7757079.188，收益相对于问题二中有所提高，通过互补作物搭配提升了土地利用效率，通过替代作物灵活调整应对市场波动，田间管理效率与实际可行性显著提升，为乡村长期可持续种植规划提供了科学且具操作性的决策支持。

一、 问题重述

1.1 问题背景

华北地区某乡村具有露天耕地 1201 亩，分散在 34 个大小不同的地块，地块类型包括平旱地、梯田、山坡地和水浇地，适合种植粮食和水稻类作物；具有 16 个普通大棚和 4 个智慧大棚，适合种植蔬菜和菌类作物。耕地需遵循农作物轮作原则，避免因连续重茬种植导致减产，并要求每块地在三年内至少种植一次豆类作物等种植约束。种植方案还需考虑田间管理的便利性，避免农作物种植地块过于分散或面积过小^[3]。

问题 1：假设在 2024 至 2030 年各农作物的销售量、种植成本、亩产量和销售价格与 2023 年保持稳定，每季农作物在当季售卖，考虑在以下两种情况下该乡 2024 至 2030 年农作物的最优种植方案。

情况 1：超过部分滞销，造成浪费。

情况 2：超过部分按 2023 年销售价格的 50%降价出售。

问题 2：考虑到未来小麦和玉米的销售量年均增长率介于 5%-10%之间，其他作物未来的销售量具有波动。同时亩产量受气候等因素影响波动，种植成本预计每年增长约 5%，粮食类销售价格基本稳定，蔬菜价格预计年均增长变化，食用菌价格稳中有降。根据农作物的销售量、亩产量、种植成本和销售价格的影响，给出该乡 2024 至 2030 年农作物的最优种植方案。

问题 3：在问题 2 的基础上，进一步考虑了农作物之间的替代性和互补性，以及销售量、销售价格与种植成本之间的相关性。综合这些因素，制定了 2024 至 2030 年的最优种植策略，并通过模拟数据进行分析。最终，将结果与问题 2 的方案进行了对比分析，以评估不同因素对种植方案的影响和优化效果。

二、 问题分析

2.1 问题一分析

在两种销售方案的情境下，设计 2024 年-2030 年农作物的种植方案，确保利益最大化。采用遗传规划算法^[4]，考虑多约束条件，包括耕地面积、农作物轮种要求(如每种作物在同一地块（含大棚）都不能连续重茬种植、豆类每三年需种植一次)、作物在单个地块种植面积不易太小、作物种植地不能太分散和地块类型和农作物匹配条件等。决策变量包括农作物的面积、

销售量、时间维度、地块、作物。通过初始种群生成、交叉变异、约束修复、适应度计算和最优选择[5]，最终获取近似最优种植方案。

2.2 问题二的分析

问题二相较于问题一，进一步贴近农业生产实际。它考虑了因气候、市场等因素，销售量、亩产量、销售价格会产生波动，比如气候异常可能导致亩产量大幅变化，市场供需变动会影响销售价格和销量。采用模拟退火算法^[6]，求解 2024-2030 年的最优种植方案，该算法能在波动因素带来的复杂解空间中，有效避免局部最优，为生产者在不确定环境下的种植决策提供更优依据。

2.3 问题三的分析

问题三在问题二基础上，进一步融入农作物替代互补性及变量相关性。从作物关系看，存在如苜蓿与红豆互斥、生菜与西红柿互补等情况；变量关联上，预期销量与价格、成本呈正相关（相关系数约 0.82）。采用 NSGA-II 算法[7]求解，通过非支配排序和拥挤度距离，输出兼顾利润与资源利用的多样方案。模拟数据显示，最优利润达 775 万元，较问题二 651.87 万元高，通过互补作物组合（如荞麦与小麦）提升了土地利用效率，且规避了替代作物同地块种植的风险。

三、 模型假设

3.1 假设每块耕地种植的作物不超过 2 种。考虑到每种作物在单个地块种植面积不宜太小，以减少每块耕地的作物量简化模型。

3.2 考虑到建模方便，假设物种的销售单价为规定单价的中位数。

3.3 考虑到每个作物在单个地块的种植面积不宜过小，约束每种作物在单个地块的种植面积大于该地块面积的 50%。

3.4 假设 2024-2030 期间，未出现极端天气或者特殊情况对亩产量、销售价格等造成较大的波动

3.5 符号说明

符号	说明
x	2024 到 2030 的年数
y	地块数
z	农作物编号

r	季数
V_{xzt}	第 x 年 r 季作物 z 的预期销量
A_{xzt}	第 x 年 r 季作物 z 的销售单价
B_{xzt}	第 x 年 r 季作物 z 的销量
C_{yxt}	地块 y 在第 x 年的 r 季种植的作物 z 的亩产量
D_{yxt}	地块 y 在第 x 年的 r 季种植的作物 z 的成本
S_{yxt}	地块 y 在第 x 年的 r 季种植的作物 z 的亩数
K_{yxt}	地块 y 在第 x 年的 r 季是否种植作物 z

四、 数据认识

4.1 数据编码

附件 1 对平旱地、梯田、三坡地、水浇地、普通大棚、智慧大棚五个地块类型进行了 A-E 的编码，并将不同地块依次按照其分组内部的顺序进行数据编码，对 41 种农作物进行了编码处理。

表 4.1 作物编码

编 号	作物名 称	作物类型	编 号	作物 名称	作物 类型
1	黄豆	粮食（豆 类）	22	茄子	茄子
2	黑豆	粮食（豆 类）	23	菠菜	菠菜
3	红豆	粮食（豆 类）	24	青椒	青椒
4	绿豆	粮食（豆 类）	25	菜花	菜花
5	爬豆	粮食（豆 类）	26	包菜	蔬菜
6	小麦	粮食	27	油麦	蔬菜

				菜	
7	玉米	粮食	28	小青 菜	蔬菜
8	谷子	粮食	29	黄瓜	蔬菜
9	高粱	粮食	30	生菜	蔬菜
10	黍子	粮食	31	辣椒	蔬菜
11	荞麦	粮食	32	空心 菜	蔬菜
12	南瓜	粮食	33	黄心 菜	蔬菜
13	红薯	粮食	34	芹菜	蔬菜
14	莜麦	粮食	35	大白 菜	蔬菜
15	大麦	粮食	36	白萝 卜	蔬菜
16	水稻	粮食	27	红萝 卜	蔬菜
17	豇豆	蔬菜（豆 类）	38	榆黄 菇	食用 菌
18	刀豆	蔬菜（豆 类）	39	香菇	食用 菌
19	芸豆	蔬菜（豆 类）	40	白灵 菇	食用 菌
20	土豆	蔬菜（豆 类）	41	羊肚 菌	食用 菌
21	西红柿	蔬菜			

表 4.2 不同地块编码

编 号	种 植地块	地 块类型	种植 面积/亩	号	种 植地块	地块类 型	种植面 积/亩
1	A1	平旱地	80	28	D2	水浇地	10
2	A2	平旱地	55	29	D3	水浇地	14
3	A3	平旱地	35	30	D4	水浇地	6
4	A4	平旱地	72	31	D5	水浇地	10
5	A5	平旱地	68	32	D6	水浇地	12
6	A6	平旱地	55	33	D7	水浇地	22
7	B1	梯田	60	34	D8	水浇地	20
8	B2	梯田	46	35	E1	普通大 棚	0.6
9	B3	梯田	40	36	E2	普通大 棚	0.6
10	B4	梯田	28	37	E3	普通大 棚	0.6
11	B5	梯田	25	38	E4	普通大 棚	0.6

12	B6	梯田	86	39	E5	普通大 棚	0.6
13	B7	梯田	55	40	E6	普通大 棚	0.6
14	B8	梯田	44	41	E7	普通大 棚	0.6
15	B9	梯田	50	42	E8	普通大 棚	0.6
16	B10	梯田	25	43	E9	普通大 棚	0.6
17	B11	梯田	60	44	E10	普通大 棚	0.6
18	B12	梯田	45	45	E11	普通大 棚	0.6
19	B13	梯田	35	46	E12	普通大 棚	0.6
20	B14	梯田	20	47	E13	普通大 棚	0.6
21	C1	山坡地	15	48	E14	普通大 棚	0.6
22	C2	山坡地	13	49	E15	普通大 棚	0.6
23	C3	山坡地	15	50	E16	普通大 棚	0.6
24	C4	山坡地	18	51	F1	智慧大 棚	0.6
25	C5	山坡地	27	52	F2	智慧大 棚	0.6
26	C6	山坡地	20	53	F3	智慧大	0.6

						棚	
						智慧大	
27	D1	水浇地	15	54	F4	棚	0.6

4.2、数据认识

图 1 展示了不同地块类型的占比情况，地块类别占比最大的是平旱地和梯田，其次是山坡地和水浇地，而普通大鵬和智慧大棚的占比极低。

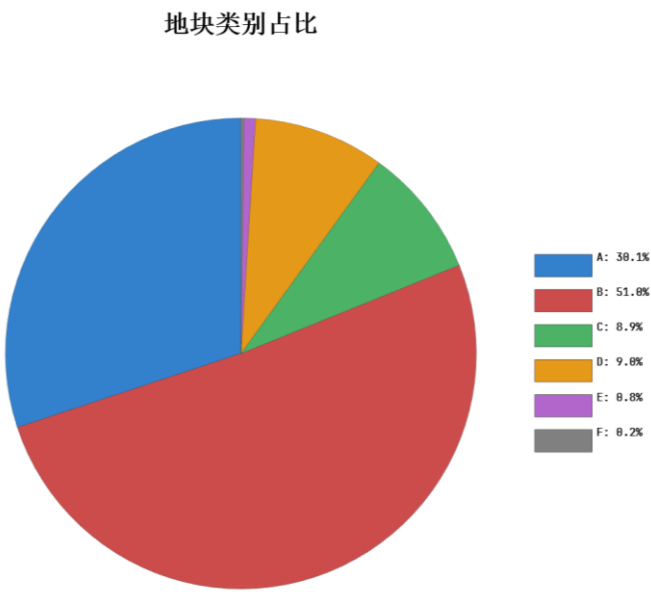


图 4.1 地块类别占比

五、 问题一的模型建立与求解

5.1 模型的建立

最大化总利润的目标函数：

$$\max z = P - Q$$

其中 z 代表 2024-2030 的利润， P 代表 2024-2030 的收入， Q 代表 2024-2030 的成本。

$$P = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(A_{xzr} \cdot \sum_{y=1}^Y (S_{yxzr} \cdot C_{yxzr}) \right)$$

$$Q = \sum_{y=1}^Y \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z (S_{yxzr} \cdot D_{yxzr})$$

X :代表时间的年数，从 2024 年到 2030 共 7 年。因此， $X=7$ 。

Y:代表地块的总数。Y=54。

Z:代表作物的总数。Z=41。

R:代表每年种植的季数。R=2

A_{xzt} 代表第 x 年的 r 季作物 z 的销售单价； S_{yxtz} 代表地块 y 在第 x 年的 r 季种植的作物 z 的亩数， D_{yxtz} 代表地块 y 在第 x 年的 r 季种植的作物 z 的成本， C_{yxtz} 代表地块 y 在第 x 年的 r 季种植的作物 z 的亩产量。

情况 1：滞销浪费

$$P = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(\min \left(\sum_{y=1}^Y (S_{yxtz} \cdot C_{yxtz}), V_{xzt} \right) \cdot A_{xzt} \right)$$

其中 V_{xzt} 第 x 年 r 季作物 z 的预期销量。

情况二：降价销售

$$P = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(\min \left(\sum_{y=1}^Y (S_{yxtz} \cdot C_{yxtz}), V_{xzt} \right) \cdot A_{xzt} + \max \left(0, \sum_{y=1}^Y (S_{yxtz} \cdot C_{yxtz}) - V_{xzt} \right) \cdot A_{xzt} \cdot 0.5 \right)$$

5.2 约束条件

不同耕地种植季数约束。平旱地、梯田和山坡地每年只能种植一季作物，水浇地可以种植一季或两季作物，普通大棚和智慧大棚能种植两季作物。

$$\begin{cases} r \leq 1, \text{if } y \in \{A1, A2, A3, A4, A5, A6, B1, B2, B3, B4, B5, B6, \\ B7, B8, B9, B10, B11, B12, B13, B14, C1, C2, C3, C4, C5, C6\} \\ r \leq 2, \text{else} \end{cases}$$

(1) 水浇地每年可以单季种植水稻或者两种季节的蔬菜作物。

$$\begin{cases} r \leq 1, \text{if } z \in \{16\} \\ r \leq 2, \text{else} \end{cases}, l \in \{D1, D2, D3, D4, D5, D6, D7, D8\}$$

(2) 连续种植限制。同一地块不能连续种植相同的农作物。

$$\begin{cases} K_{y,x,z,r} + K_{y,x+1,z,r} \leq 1, \text{if } r \leq 1 \\ K_{y,x,z,1} + K_{y,x,z,2} \text{ or } K_{y,x,z,2} + K_{y,x+1,z,1} \leq 1, \text{else} \end{cases}$$

(3) 农作物轮作约束。每个块地三年至少种植一次豆类作物。

$$\sum_{i=0}^2 K_{y,x+i,z,r} \geq 1, z \in \{1, 2, 3, 4, 5, 17, 18, 19\}$$

(4) 耕地种植作物类型限制。

平旱地、梯田和山坡地的种植作物类型限制。

$$\begin{cases} S_{y,x,z,r} \geq 0, \text{ if } y \in \{A1, A2, A3, A4, A5, A6, B1, B2, B3, B4, B5, B6, \\ B7, B8, B9, B10, B11, B12, B13, B14, C1, C2, C3, C4, C5, C6\} \\ z \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\} \\ S_{y,x,z,r} = 0, \text{ else} \end{cases}$$

水浇地第一季、普通大棚第一季和智慧大棚第一季、第二季作物限制。

$$\begin{cases} S_{y,x,z,r} \geq 0, \text{ if } y \in \{D1, D2, D3, D4, D5, D6, D7, D8, E1, E2, E3, E4, \\ E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16\}, R = 1 \\ S_{y,x,z,r} \geq 0, \text{ if } y \in \{F1, F2, F3, F4\} \\ S_{y,x,z,r} = 0, \text{ else} \end{cases}$$

水浇地第二季作物限制。

$$\begin{cases} S_{y,x,z,r} \geq 0, \text{ if } y \in \{D1, D2, D3, D4, D5, D6, D7, D8\}, R = 2 \\ S_{y,x,z,r} = 0, \text{ else} \end{cases}$$

普通大棚第二季作物限制。

$$\begin{cases} S_{y,x,z,r} \geq 0, \text{ if } y \in \{E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, \\ E11, E12, E13, E14, E15, E16\}, R = 2 \\ S_{y,x,z,r} = 0, \text{ else} \end{cases}$$

(5) 种植区域集中度约束。一个作物同期至多只能在 3 分地块上种植。

$$\sum_y K_{y,x,z,r} \leq 3$$

(6) 种植面积约束。每种作物在单个地块种植的面积不易过小，每种作物在单个地块种植的面积不小改地块的 50%。

$$S_{y,x,z,r} > 0.5 \cdot L_y$$

L_y 是 y 个地块的亩数。

则，问题一的模型建立如下：

$$\max z = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(A_{x,z,r} \cdot \sum_{y=1}^Y (S_{y,x,z,r} \cdot C_{y,x,z,r}) \right) - \sum_{y=1}^Y \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z (S_{y,x,z,r} \cdot D_{y,x,z,r})$$

$$\begin{cases}
\begin{cases} r \leq 1, \text{if } y \in \{A1, A2, A3, A4, A5, A6, B1, B2, B3, B4, B5, B6, \\ B7, B8, B9, B10, B11, B12, B13, B14, C1, C2, C3, C4, C5, C6\} \\ r \leq 2, \text{else} \end{cases} \\
\begin{cases} r \leq 1, \text{if } z \in \{16\} \\ r \leq 2, \text{else} \end{cases}, l \in \{D1, D2, D3, D4, D5, D6, D7, D8\} \\
\begin{cases} K_{y,x,z,r} + K_{y,x+1,z,r} \leq 1, \text{if } r \leq 1 \\ K_{y,x,z,1} + K_{y,x,z,2} \text{ or } K_{y,x,z,2} + K_{y,x+1,z,1} \leq 1, \text{else} \end{cases} \\
\sum_{i=0}^2 K_{y,x+i,z,r} \geq 1, z \in \{1, 2, 3, 4, 5, 17, 18, 19\}
\end{cases}$$

$$\begin{cases}
\begin{cases} S_{y,x,z,r} \geq 0, \text{if } y \in \{A1, A2, A3, A4, A5, A6, B1, B2, B3, B4, B5, B6, \\ B7, B8, B9, B10, B11, B12, B13, B14, C1, C2, C3, C4, C5, C6\} \\ z \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\} \\ S_{y,x,z,r} = 0, \text{else} \end{cases} \\
\begin{cases} S_{y,x,z,r} \geq 0, \text{if } y \in \{D1, D2, D3, D4, D5, D6, D7, D8, E1, E2, E3, E4, \\ E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16\}, R = 1 \\ S_{y,x,z,r} \geq 0, \text{if } y \in \{F1, F2, F3, F4\} \\ S_{y,x,z,r} = 0, \text{else} \end{cases} \\
\begin{cases} S_{y,x,z,r} \geq 0, \text{if } y \in \{D1, D2, D3, D4, D5, D6, D7, D8\}, R = 2 \\ S_{y,x,z,r} = 0, \text{else} \end{cases} \\
\begin{cases} S_{y,x,z,r} \geq 0, \text{if } y \in \{E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, \\ E11, E12, E13, E14, E15, E16\}, R = 2 \\ S_{y,x,z,r} = 0, \text{else} \end{cases} \\
\sum_y K_{y,x,z,r} \leq 3 \\
S_{y,x,z,r} > 0.5 \cdot L_y
\end{cases}$$

5.3 模型求解

(1) 遗传规划算法

遗传规划算法是受生物遗传进化启发的优化方法，模拟自然选择与遗传机制。以函数、变量构成的程序个体为操作对象，通过选择（依适应度选优个体）、交叉（重组基因生成新个体）、变异（随机改变基因添多样），在程序空间自主寻优。具自适应性，可动态适配问题；鲁棒性强，应对复杂场景。能自动构建程序结构，广泛用于符号回归、算法设计等，为复杂任务提供智能求

解路径，遗传规划算法流程如图 5.1 所示

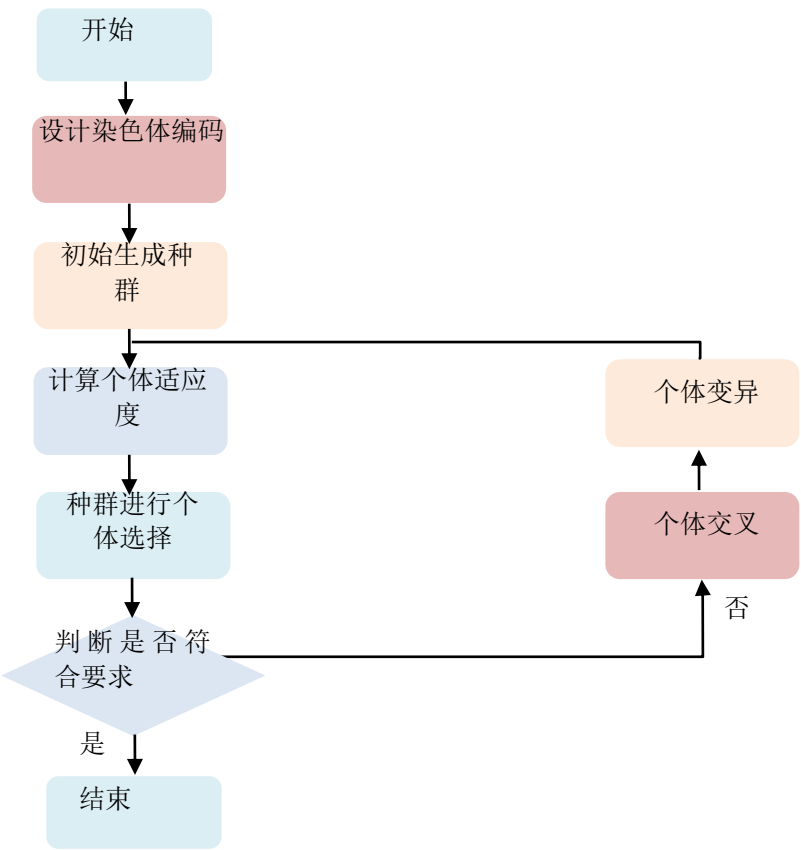


图 5.1 遗传规划算法流程

(2) 模型参数设置

遗传规划算法参数设置如下表所示：

表 5.1 遗传规划算法参数设置表

参数	取值
迭代次数	200
交叉率	0.8
变异率	0.5
种群规模	3

(3) 模型求解结果

情况 1 迭代收敛图如图 5.2 所示，随着迭代次数的增加，收敛曲线在迭代到 149 代时，收敛曲线显示收敛，收敛值稳定在 6978352 左右。

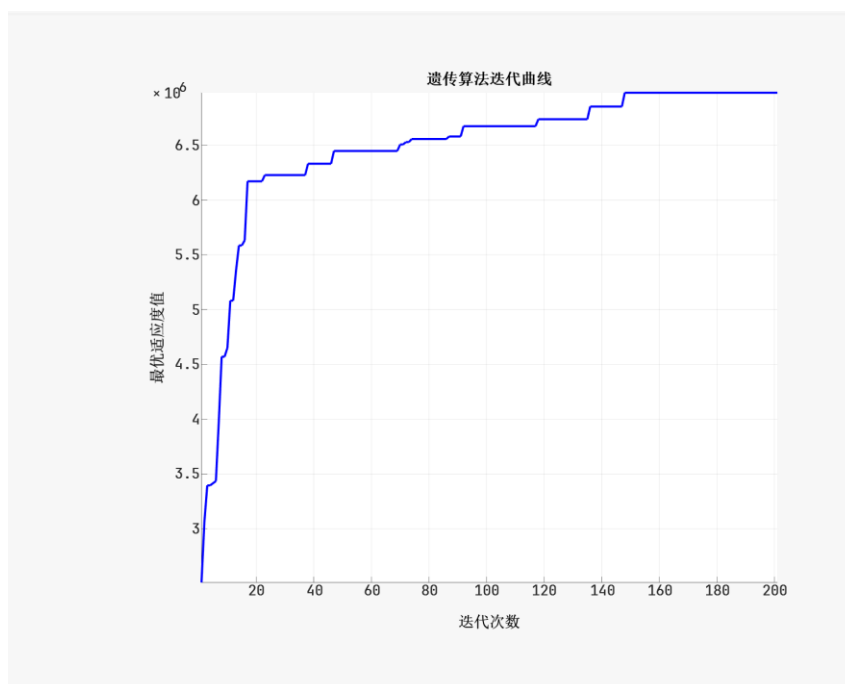


图 5.2 情况 1 的利润收敛曲线

情况 2 迭代收敛图如下所示, 经过 65 步迭代后收敛在 280000000. 这趋势表明算法在迭代过程中逐渐接近最优解, 并在较小的迭代步数后达到稳定。

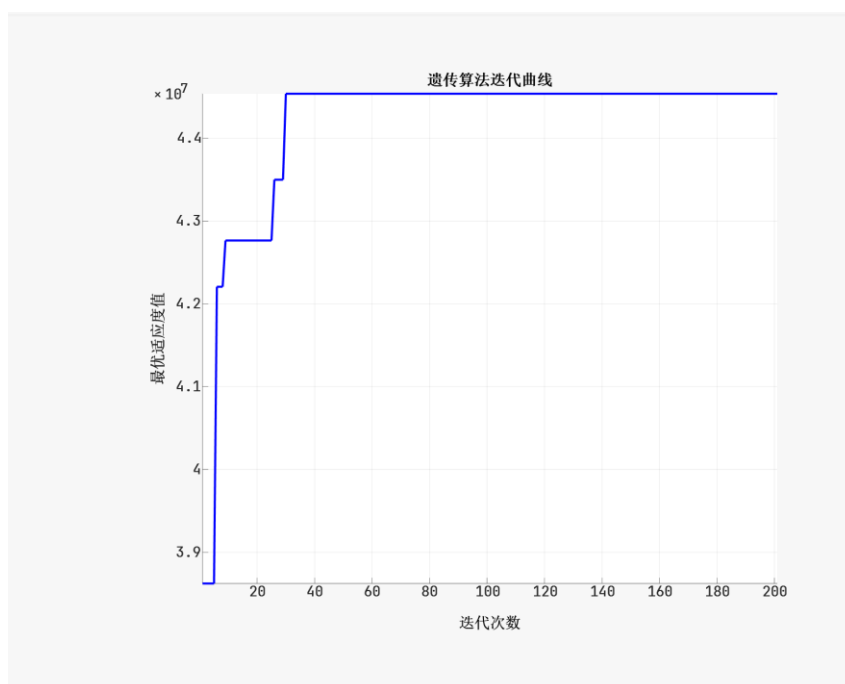


图 5.3 情况 2 的利润收敛曲线

情况 1 各作物的产量如下图所示：

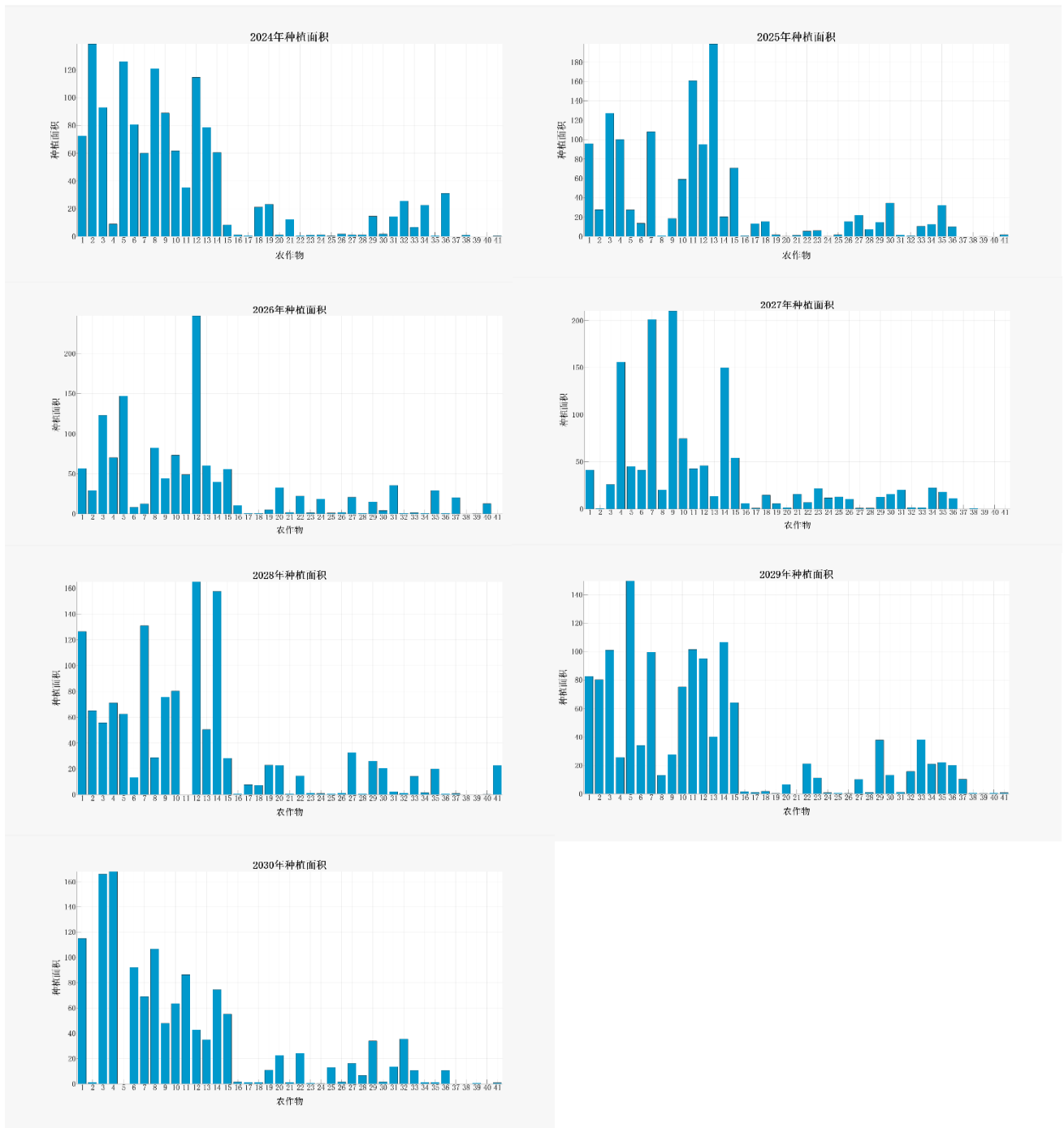
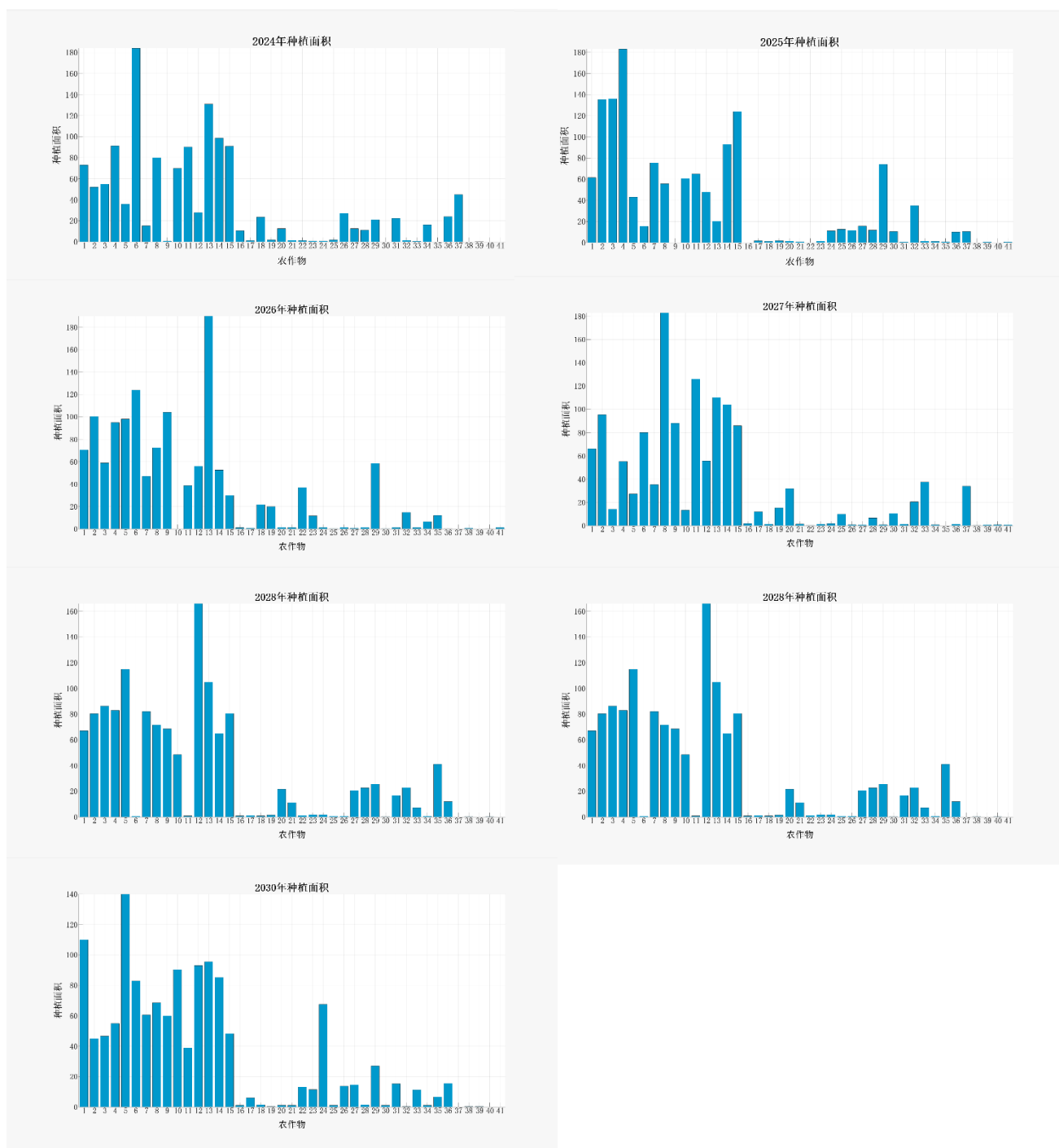


图 5.4 情况 1 作物种植面积



情况 2 各作物的产量如下图所示：

图 5.5 情况 2 作物种植面积

从 2024 至 2030 年的种植面积汇总来看，每年的作物布局均呈现出显著的“粮食主导”特征，粮食类作物占据绝大部分面积，而蔬菜和食用菌的种植规模相对有限。这一结构主要源于三方面因素：粮食作为基础战略物资，具有稳定且庞大的市场需求，粮食作物对不同耕地类型适应性强，易于规模化生产；

而蔬菜和食用菌对土壤、灌溉及设施要求较高，市场价格波动大、种植风险突出，因而多种主体倾向于采取稳健策略，优先保障粮食生产。

六、 问题二的模型建立与求解

6.1 问题分析与数据预处理

问题二在问题一的基础上考虑各种农作物的预期销售量、亩产量、种植成本和销售价格的不确定性以及潜在的种植风险。潜在的风险包括气候^{[8][9]}、市场条件及市场需求等等。在考虑风险的基础上得到最大的总利润。其中不确定性因素如下表所示：

表 3 不确定因素表

不确定因素	波动范围
小麦、玉米销量年增长率	5%到 10%
除小麦、玉米外农作物的销量波动	-5%到 5%
农作物年亩产量	-10%到 10%
农作物的年种植成本	5%
蔬菜价格年增长率	5%
除羊肚菌外食用菌价格年缩减率	5%到 1%
羊肚菌价格年缩减率	5%

6.2 模型的建立

期望收益的最坏情况下的利润最大化：

$$\max F = E \left[\min_{\square \in Q} z(\square) \right]$$

其中

$$z = P - Q$$

P 代表 2024-2030 的收入，Q 代表 2024-2030 的成本。

$$P = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(A_{x zr} \cdot \sum_{y=1}^Y (S_{y x zr} \cdot C_{y x zr}) \right)$$

$$Q = \sum_{y=1}^Y \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z (S_{y x zr} \cdot D_{y x zr})$$

A_{xzt} 代表第 x 年的 r 季作物 z 的销售单价, 考虑了作物年销售价格的变化率; S_{yxtz} 代表地块 y 在第 x 年的 r 季种植的作物 z 的亩数, D_{yxtz} 代表地块 y 在第 x 年的 r 季种植的作物 z 的成本, 考虑了作物年种植成本的变化率, C_{yxtz} 代表地块 y 在第 x 年的 r 季种植的作物 z 的亩产量, 考虑了作物年亩产量的变化率。

情况 1: 滞销浪费

$$P = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(\min \left(\sum_{y=1}^Y (S_{yxtz} \cdot C_{yxtz}), V_{xzt} \right) \cdot A_{xzt} \right)$$

其中 V_{xzt} 第 x 年 r 季作物 z 的预期销量, 考虑了作物年预期销量的变化率。

情况二: 降价销售

$$P = \sum_{x=1}^X \sum_{r=1}^R \sum_{z=1}^Z \left(\min \left(\sum_{y=1}^Y (S_{yxtz} \cdot C_{yxtz}), V_{xzt} \right) \cdot A_{xzt} + \max \left(0, \sum_{y=1}^Y (S_{yxtz} \cdot C_{yxtz}) - V_{xzt} \right) \cdot A_{xzt} \cdot 0.5 \right)$$

约束条件:

(1) 销售量约束

农作物的销售量有 $\pm 5\%$ 的波动, 其中小麦和玉米的年销售量存在 $5\%-10\%$ 的波动。假设其中 V_{xzt} 第 2023 年 r 季作物 z 的销量。

$$\text{则} \begin{cases} V_{xzt} = V_{xzt} \cdot (1 + r_z)^x, & \text{if } z \in \{6, 7\} \\ V_{xzt} = V_{xzt} \cdot (1 + \varepsilon_z)^x, & \text{else} \end{cases}$$

其中 $r_z \in [0.05, 0.10]$ 是小麦和玉米的年销量变化率的范围;

$\varepsilon_z \in [-0.05, 0.05]$ 是其他农作物的年销量变化率的范围。

(2) 亩产量约束

农作物受到气候波动的影响, 年亩产量有 $\pm 10\%$ 的波动。假设其中 C_{yxtz} 第 2023 年地块 y 的 r 季作物 z 的销量。

$$C_{yxtz} = C_{yxtz} \cdot (1 + \nu)^x$$

其中 $\nu \in [-0.1, 0.1]$ 是农作物年亩产量变化率的范围。

(3) 种植成本约束

农作物收到市场的影响，种植成本每年增加 5%，假设其中 D_{yzt} 第 2023 年地块 y 的 r 季作物 z 的种植成本。

$$D_{xyzt} = D_{yzt} \cdot 1.05^x$$

(4) 销售价格约束

粮食类作物的销售价格稳定，不考虑约束；蔬菜类作物的销售价格每年增加 5%，食用菌作物的销售价格变化率为-5%至-1%；其中羊肚菌的销售价格下降 5%。假设其中 A_{zr} 第 2023 年 r 季作物 z 的种植成本。

$$\begin{cases} A_{xzt} = A_{zt} \cdot 1.05^x, z \in \{17, 18, 19, 20, 21, 22, 23, 24, 24, 26, 27, \} \\ A_{xzt} = A_{zt} \cdot (1 + \zeta_z)^x, z \in \{38, 39, 50\} \\ A_{xzt} = A_{zt} \cdot 0.95^x, z \in \{41\} \end{cases}$$

其中 $\zeta_z \in [-0.5, -0.1]$ 是除羊肚菌外食用菌年销售价格变化率的范围，其余约束条件与问题一中保持不变。

6.3 模型求解

(1) 模拟退火算法

模拟退火算法在农作物种植优化中，通过模拟物理退火“高温探索、低温收敛”的过程，有效处理种植方案的复杂约束与多目标优化问题：首先以随机生成的初始种植方案（如不同地块的作物种类、种植面积分配）为起点，在较高初始温度下，允许对当前方案进行随机调整（如更换作物、调整面积），即使新方案短期利润较低也可能被接受，从而避免局限于局部最优；随着温度逐步降低，算法逐渐聚焦于更优方案的精细优化，优先接受能提高产量、降低成本或满足轮作、地块适配等约束的调整，最终在满足作物生长规律（如重茬限制、季节适配）和资源约束（如地块面积、劳动力）的前提下，找到全局最优的种植方案，实现利润最大化或资源利用效率最优，模拟退火算法的流程图

如图 6.1 所示

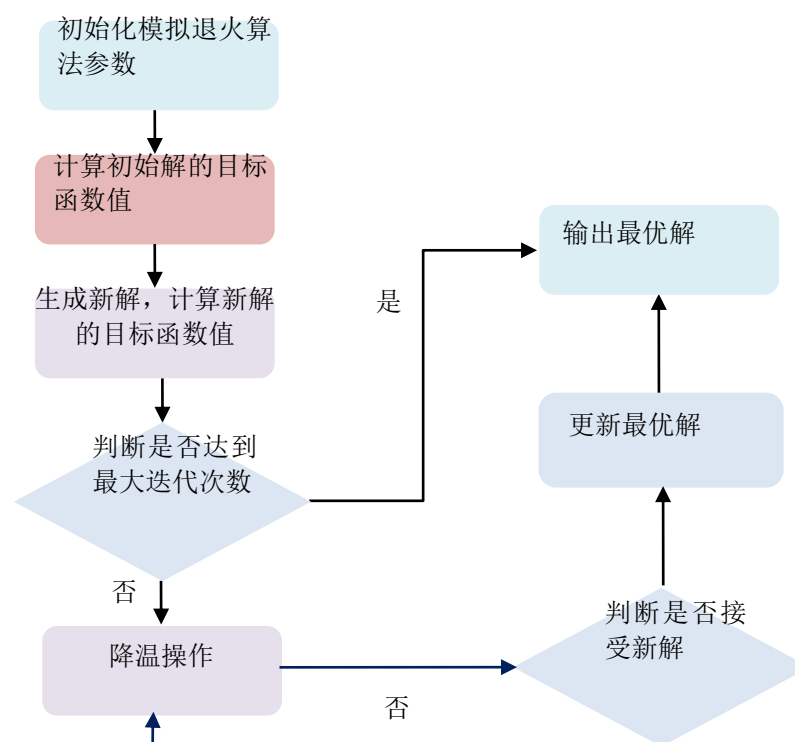


图 6.1 模拟退火算法流程图

在模拟退火算法用于求解问题时，首先会构建一个满足核心硬性约束（如地块-作物适配、面积限制等）的初始种植方案，并设定初始温度以启动搜索过程。如流程所示，算法先完成参数初始化，随后计算初始方案的目标函数值（如利润），再通过随机扰动生成新的候选方案。在每次方案迭代中，对于未满足软性约束（如作物分散种植、轮作建议等）的候选解，会通过施加惩罚项降低其适配度，以此优先保障方案符合硬性约束的基础要求。随着算法执行降温操作，方案的随机扰动幅度逐渐减小——前期通过较大扰动探索广泛的解空间，避免陷入局部最优；后期则聚焦于对优质方案的精细调整，更注重在利润提升与软性约束满足之间找到平衡。在此过程中，若新生成的候选解满足预设的更新准则（如利润更高或惩罚更小），则更新当前最优方案。经过多轮迭代与温度逐步降低，算法最终能高效收敛到一个同时满足所有约束条件、且利润接近全局最大值的最优种植方案。

（2）模型参数设置

模拟退火算法参数设置如下表：

表 6.1 模拟退火算法参数设置表

参数	取值
初始温度 (T0)	1000
温度衰减系数 (α)	0.9
最大迭代次数	160
惩罚系数 λ	0.8

(3) 模型求解结果

情况 1 下的模型收敛曲线如下图所示：

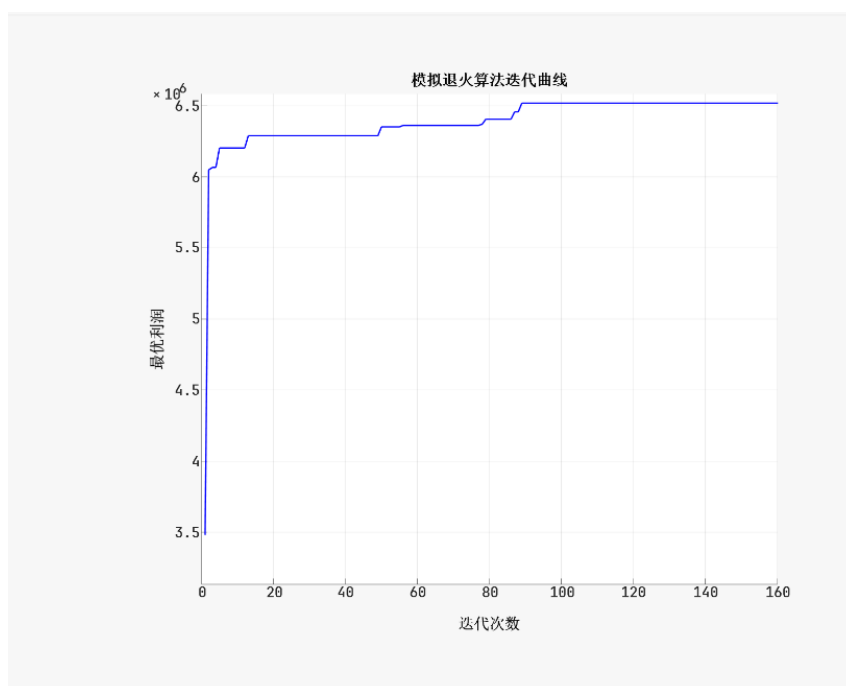


图 6.2 情况 1 利润收敛曲线

情况 2 下的模型收敛曲线如下图所示：

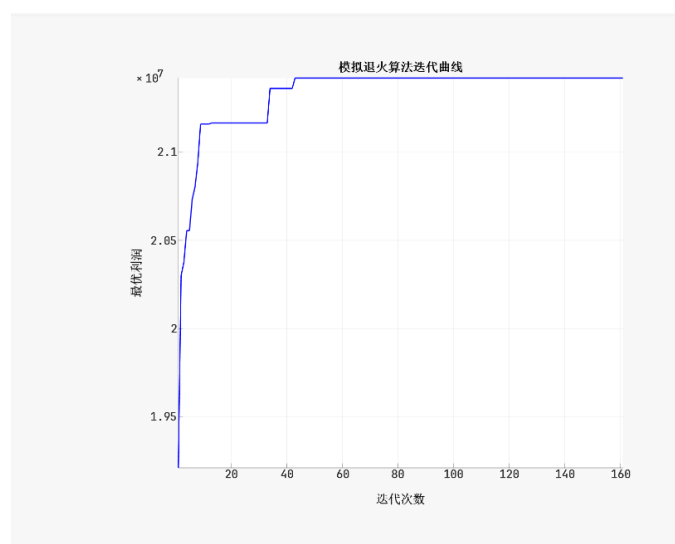


图 6.3 情况 1 利润收敛曲线

情况 1 各作物的产量如下图所示：

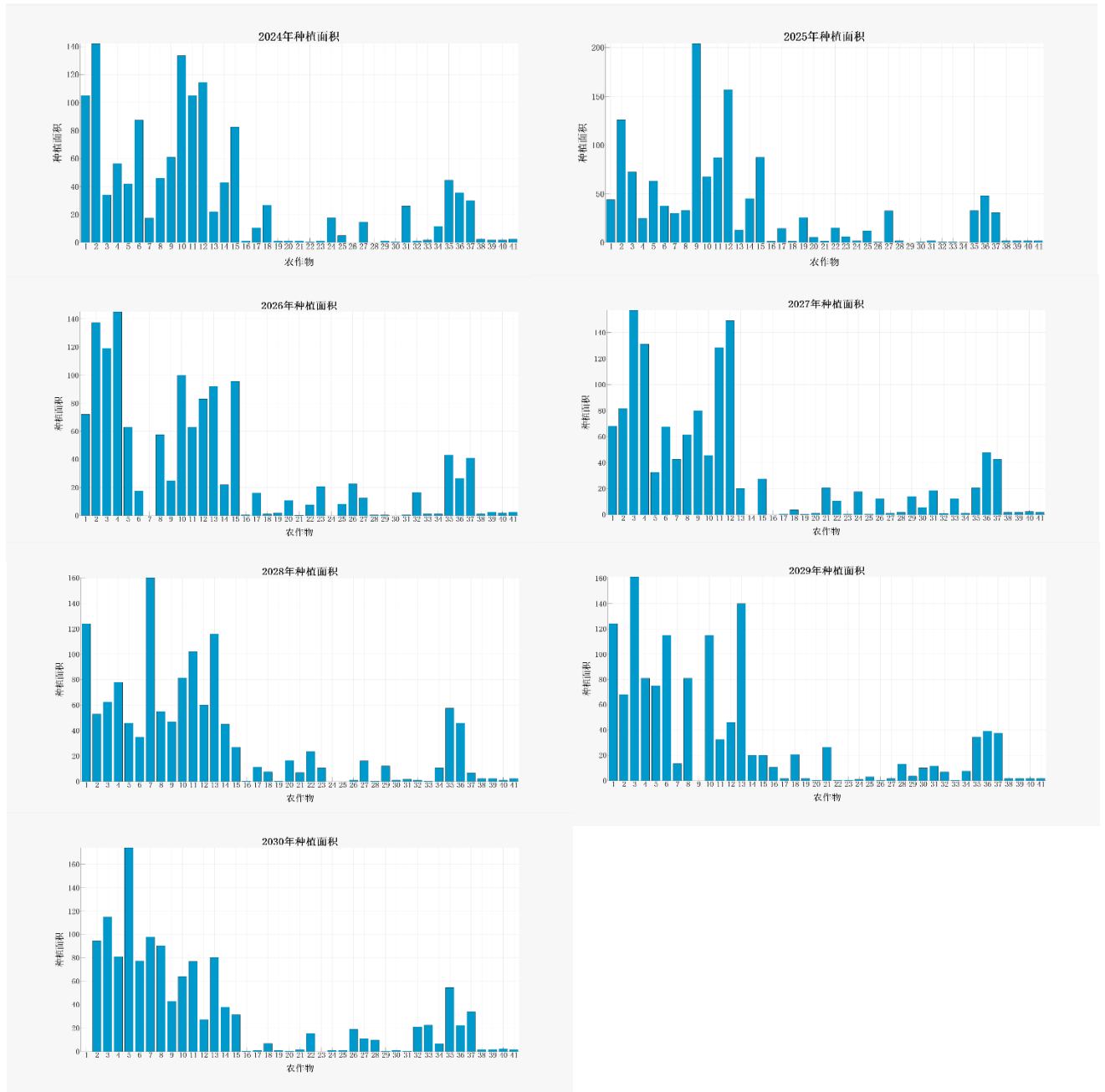


图 6.4 情况 1 作物种植面积

情况 2 各作物的产量如下图所示：

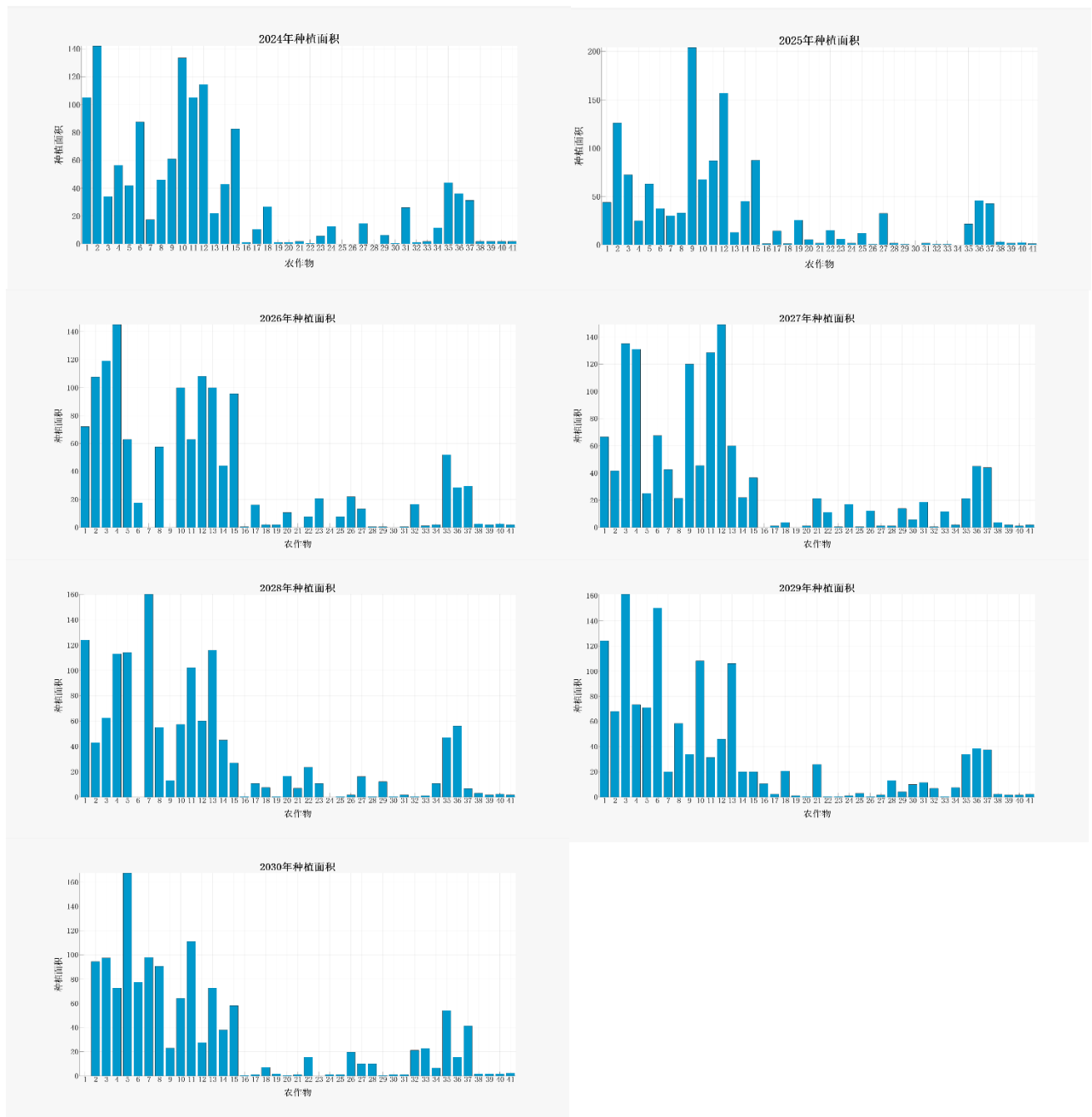


图 6.4 情况 2 作物种植面积

七、 问题三的模型建立与求解

7.1 问题分析与数据预处理

问题三在问题二的基础上考虑农作物可替代性和互补性，以及预期销售额、销售价格和种植成本之间的相关性制定种植策略。

7.2 相关性模型的建立

Kolmogorov-Smirnov（简称 K-S 检验）和 Shapiro - Wilk（简称 S-W 检验）

是统计学中两种常用的正态性检验方法，均用于判断样本数据是否服从某一理论分布（最常见为正态分布）。

Kolmogorov-Smirnov 检验：基于“累积分布函数差异”的通用分析检验。

原假设 H_0 ：样本数据服从正态分布。

备择假设 H_1 ：样本数据不服从正态分布。

累积分布函数（CDF）表示随机变量小于等于某个值的概率。对于样本数据，其经验累积分布函数 $F_n(x)$ 的计算方式是，在样本 $X_1, X_2, \dots, X_n$ 中，小于等于 x 的观测值的比例，即

$$F_n(x) = \frac{1}{n} \times \text{样本中小于等于 } x \text{ 的观测值的个数。}$$

则 Kolmogorov-Smirnov 统计量（D 值）可以表述为：

$$D = \sup_x |F_n(x) - F(x)|$$

这里 $\sup$ 表示上确界，也就是所在可能的 x 值中， $|F_n(x) - F(x)|$ 的最大值。

根据计算得到的 D 值，结合样本量 n ，通过查找 K-S 检验的临界值表或者使用统计软件计算出对应的 P 值。如果 P 值大于预先设定的显著性水平（常用 0.05），就没有足够的证据拒绝原假设 H_0 ，可以认为样本数据服从正态分布；如果 P 值小于预先设定的显著性水平，就拒绝原假设 H_0 ，认为样本数据不服从正态分布。

Shapiro-Wilk 检验（S-W 检验）：基于“线性相关性”的正态分布专用检验

对样本 $X_1, X_2, \dots, X_n$ 排序，得到有序样本 $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$ ；计算样本均值 $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$ ，用于后续标准化。

则 Shapiro-Wilk 统计量（W 值）可以表述为：

$$W = \frac{\left(\sum_{i=1}^n a_i X_{(i)} \right)^2}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

其中 a_i 是“权重系数”，由标准正态的期望次序统计量 m_i 推导而来。

$$a_i = \frac{m_i}{\sqrt{\sum_{j=1}^n m_j^2}} (i=1, 2, \dots, n)$$

对于服从正态分布的有序变量，基于皮尔逊相关性分析进行讨论。

1) 正态分布相关性分析

皮尔逊相关系数 r 用于衡量两个有序变量 X 和 Y 之间线性关系的强度与方向。有序变量 $X = (X_1, X_2, X_3, \dots, X_n)$ 和 $Y = (Y_1, Y_2, Y_3, \dots, Y_n)$ 服从正态分布。

样本均值：

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i, \bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

样本协方差：

$$\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

样本标准差：

$$S_X = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}, S_Y = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2}$$

皮尔逊相关系数：

$$r = \frac{\text{cov}(X, Y)}{S_X S_Y}, r \in (-1, 1)$$

若 $r > 0$ ，表明两个变量呈正线性相关；若 $r < 0$ ，则表明两个变量呈负线性相关；若 $r = 0$ ，说明变量之间不存在线性关系。通常 $|r| \geq 0.8$ 为强相关； $0.5 \leq |r| \leq 0.8$ 为中等相关， $0.3 \leq |r| \leq 0.5$ 为弱相关， $|r| < 0.3$ 为极弱相关或无相关。

2) 不服从正态分布的相关性分析

斯皮尔曼相关系数：

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}$$

其中， n 是样本量， d_i 是 X 的秩与 Y 的秩的差值。

7.3 相关性检验

对附件 2 中的数据进行相关性检验

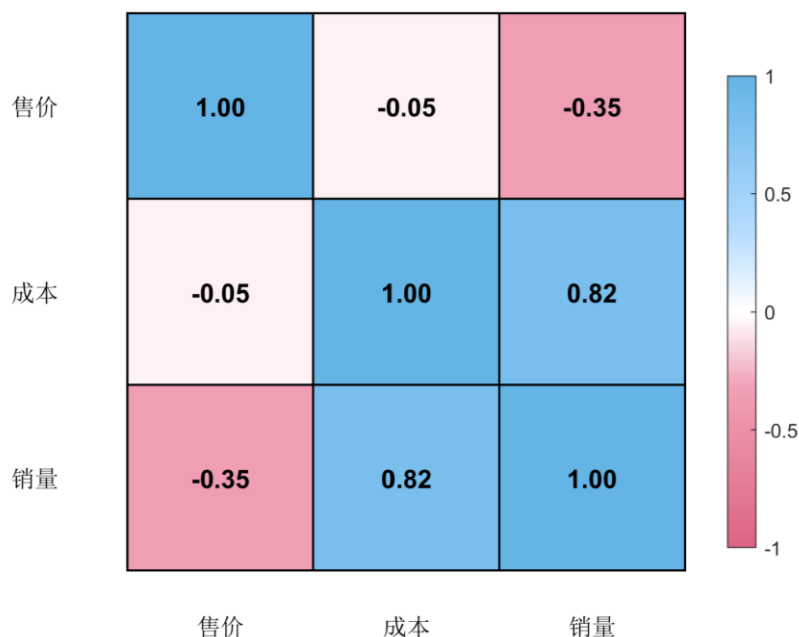


图 7.1 销售量、销售单价、销售成本的相关性热力图

通过销售量、销售单价、销售成本的相关性热力图分析，销量和售价的相关系数为-0.35，表明销量和售价之间存在负线性关系，当销量增加时，售价则对应减小；销量和成本的相关系数为 0.82，则销量和成本之间存在正线性关系，当销量增加时，成本对应增加，且相关性较强；成本和售价的相关系数为-0.05，表明成本和售价之间存在负线性相关，且相关性较弱。

图 7.2 展示了基于问题二情况一 2024 至 2030 期间，各类农作物种植面积，可以从图片中观察到不同农作物种植面积的波动规律^[10]，以及各种农作物之间可能存在一定的可替代性和互补性。进一步分析可知，在有限的耕地资源约束下，部分作物（如玉米与红豆）之间显现出明显的替代效应，当某类作物收益预期上升时，其种植面积可能挤占另一类作物的空间^[11]；而另一些作物（如荞麦与小麦）则存在显著的互补效应，两个作物的种植面积变化趋势相似。



图 7.2 2024 至 2030 各农作物种植面积

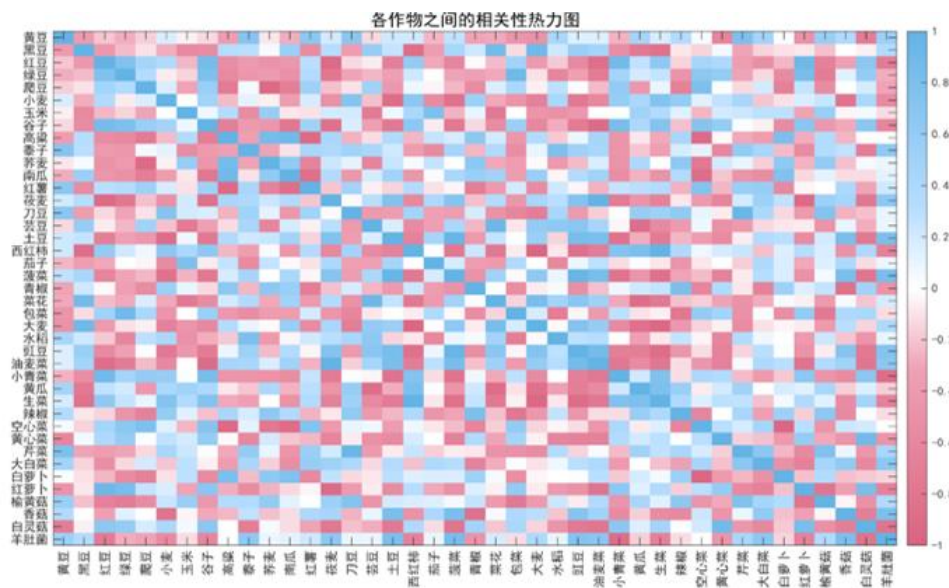


图 7.3 各类农作物种植面积的相关性热力图

对 2024 至 2030 年各种作物之间的种植面积进行相关性分析，并绘制了农作物的热力图，如图 7.5 所示，图中显示了不同农作物之间的相关性。

从图 7.3 中可以看出部分作物之间存在明显的正相关性，这一特征表明它们在生长过程中存在一定的互补性，例如生菜和西红柿的相关性为 0.982，如此高的正相关程度意味着它们在对土壤养分的需求、生长周期的适配以及对光照、水分等环境因素的响应等方面，都有着较为契合的特点。基于这样的特性，生菜和西红柿非常适合被搭配种植在同一块耕地中，在这种种植模式下，它们不仅能够更高效地利用耕地资源，还可能在生长过程中相互促进，进而有助于提升整体的种植效益。

部分作物之间存在显著的负相关性，这一特点表明它们在生长进程里存在一定的相互制约性。以苜蓿和红豆为例，二者的相关性为-0.847，这样的负相关程度说明它们在对土壤养分的竞争、生长空间的占据以及对光照、水分等环境资源的获取等方面，存在较为明显的冲突。鉴于这样的特性，黄瓜和玉米不太适合被种植在同一块耕地中，若强行在同一块耕地种植，它们可能会因为对资源的争夺而相互抑制生长，进而不利于提升整体的种植效益^[12]。

7.4 模型的建立

期望收益的利润最大化：

$$\max F_1 = E \left[\min_{\square \in Q} z(\square) \right]$$

农作物的约束：

表 7.1 可替代和互补的农作物

农作物 编号	可替代的农 作物	可互补的农作物
1		13
2		15
3	14	
4	14	
5	11	
9	13	11、12

11	5	9
12	13	9
13	12、9	1
14	3、4	
15		2
17	30	23、27
18		34
19	29	25
20		23
21		30
23		17、20
24		31
25		19
27	37	17
29	19	
30	17	21
31		24
32	36	
34		18
36	32	
37	27	

表 7.1 给出了完整可替代和互补的农作物，通过对同一地块农作物种植面积的相关系数绝对值大于 0.75 进行统计。

可替代性约束：

农作物 z_i 和 z_j 是具可替代的农作物，通过约束来表示同一时间不能同时在同一地块上种植：

$$S_{yxz_i r} + S_{yxz_j r} \leq 0.5S_{yxr}$$

$S_{yxz_i r}$ 和 $S_{yxz_j r}$ 分别表示地块 y 在第 x 年的 r 季种植作物的亩数， S_{yxr} 表示地块 y 在第 x 年的 r 季的最大种植面积。

互补性约束：

农作物 z_i 和 z_j 是具可互补的农作物：

$$S_{yz_i r} \geq \sigma S_{(y+1)xz_j r}$$

其中 $\sigma > 0$ 表明两个互补性的农作物的种植面积是正相关，则农作物农作物 z_i 和 z_j 只有在相邻或同一地块种植时才能产生互补效应。

7.5 模型求解

(1) NSGA-II 算法

NSGA-II 算法能同时处理“利润最大化”“资源高效利用”“风险最小化”等多个相互冲突的目标，通过非支配排序将种植方案划分为不同层级，优先保留那些在各目标间实现更优平衡的方案，比如既保证较高利润，又能通过互补作物组合提升土地利用率的策略。其独特的拥挤度距离机制则确保了解集的多样性，既涵盖单一高价值作物的种植方案，也包含多种互补作物搭配的组合策略，还能提供可相互替换的备选作物方案，满足不同种植场景下的决策需求。在具体实现中，可通过混合编码（如二进制编码表示作物选择、数值编码表示种植面积）将作物间的互补约束（如生菜与西红柿的协同种植增益）和替换规则（如玉米与大豆的互斥性）融入优化过程，再结合精英保留策略，算法最终能高效收敛到一个同时满足所有约束条件、且利润接近全局最大值的最优种植方案，。

(2) 模型参数设置

NSGA-II 算法参数设置如下表：

表 7.2 NSGA-II 算法参数设置表

参数	取值
迭代次数	160
交叉率	0.7
变异率	0.2
种群规模	10

(3) 模型求解结果

情况 1 的利润收敛曲线如图 7.4 所示，在考虑农作物之间可互补性和替换

性，得到的最优利润为 7757079.188 元，相对于问题二中的 6518732.835 元有所提高。

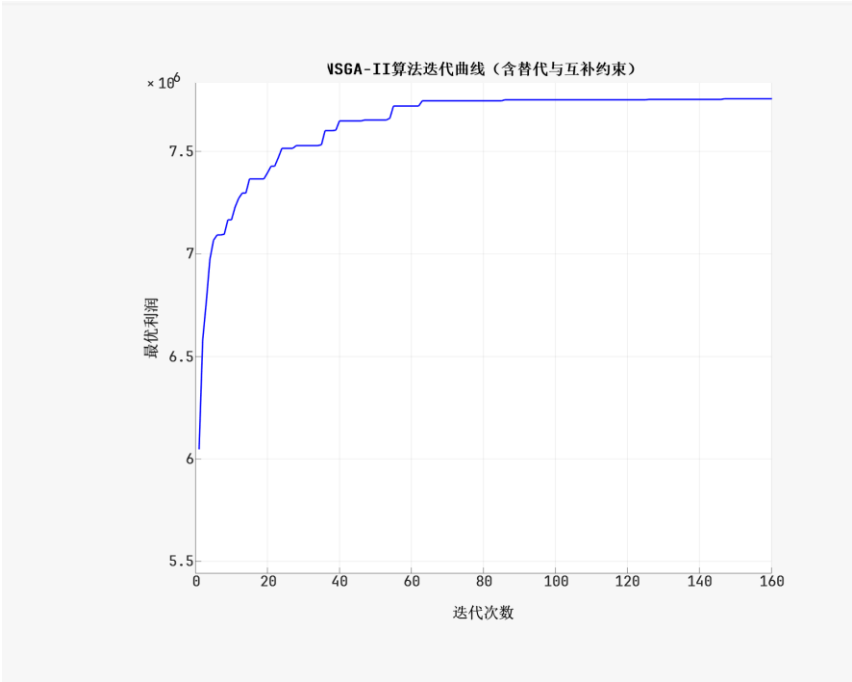


图 7.4 情况 1 的利润收敛曲线

情况 2 的利润收敛曲线如图 7.5 所示，得到的最优利润为 33660325.75 元。
情况 2 的利润收敛曲线如图 7.5 所示：

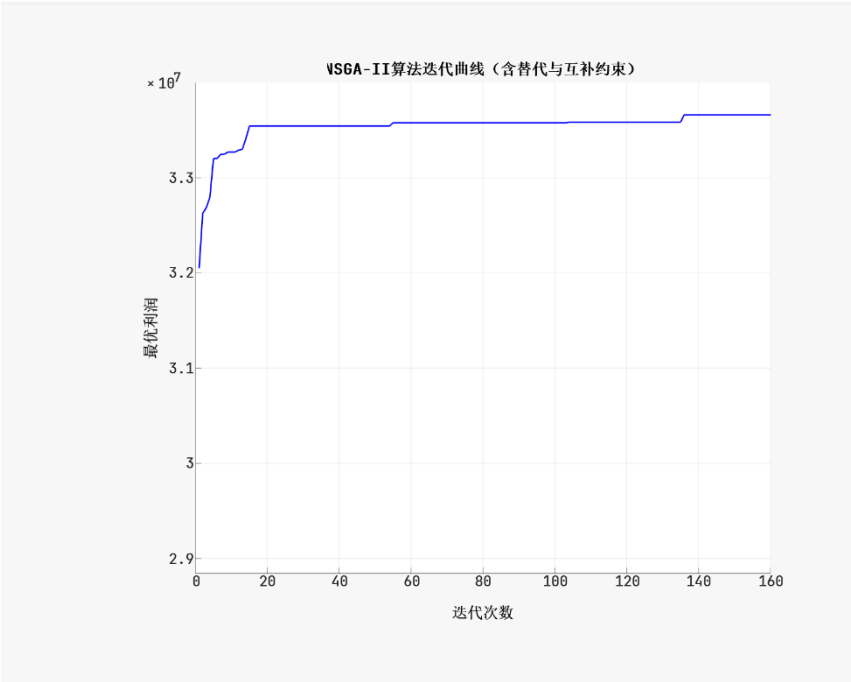


图 7.5 情况 2 的利润收敛曲线

情况 1 各作物的产量如下图所示：

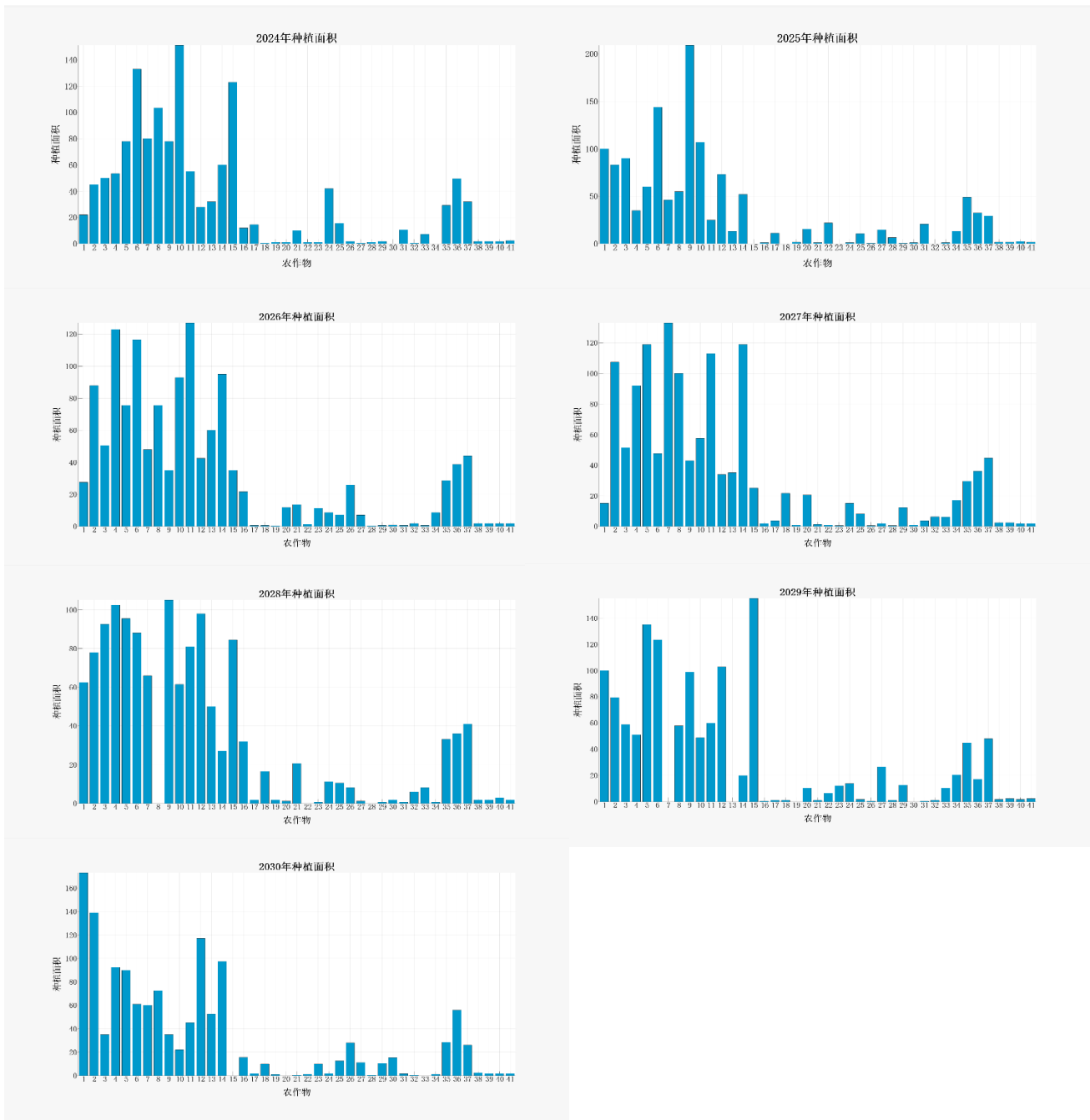


图 7.6 情况 1 作物种植面积

情况 2 各作物的产量如下图所示：



图 7.6 情况 1 作物种植面积

八、 模型的评价

模型全面考虑了农作物种植中的核心约束条件，包括耕地类型适配性（如平旱地适合粮食、大棚适合蔬菜）、轮作要求（三年内至少种植一次豆类）、连续种植限制（禁止重茬）、作物替代性与互补性等，形成了一套完整的约束体系。通过遗传规划、模拟退火、NSGA-II 等算法对约束条件的刚性处理，确保了种植方案的可行性与科学性，例如对互补作物（如生菜与西红柿）的面积比例控制（ $0.5 \leq \text{比例} \leq 2.0$ ），显著提升了耕地资源的协同利用效率。

针对问题二和问题三中的市场波动（如小麦销量年增长 5%-10%）、气候影响（亩产量±10%波动）、成本变化（种植成本本年增 5%）等不确定性因素，模型通过引入随机波动函数和风险系数，实现了对复杂农业系统的动态模拟。例如，采用分段函数处理不同作物的价格变化（蔬菜年增 5%、食用菌年降 1%-5%），使利润计算更贴近实际市场规律。

本文所构建的农作物种植优化模型，不仅可应用于特定乡村场景，还具备广泛的适应性与推广价值。在实际农业种植决策过程中，既要受到自然条件的限制，又要应对市场需求、价格波动、政策规定等复杂多变的外部环境，所以该模型在不同场景下的推广有着重要现实意义。具体可从多方面进一步推广应用：一是多区域、多气候条件下的种植优化，尽管模型针对华北山区耕作条件优化，但通过调整作物适应性、产量波动范围、种植季节等参数，能推广到其他地理区域，适配不同气候区的农业种植，像南方多雨季可对多季种植优化，北方高寒地区需考虑短暂种植季节；二是适应不同市场环境的经济收益优化，模型分析作物销售价格和市场需求变化时对未来预期销售量的假设思路，可推广至任何市场环境，引入大数据市场分析、机器学习模型等复杂市场预测模型，能更精准预测销售趋势与价格波动，同时将不同市场条件下的销售政策调整作为参数输入，可进一步提高收益；三是多作物协同优化的广泛应用，模型探讨的作物间替代性和互补性，尤其是轮作制度对作物产量的促进作用，在现代农业的多元种植、复合农业等场景广泛存在，推广该模型能优化多作物协同种植的组合，提升整体产量和收益，特别是在有机农业、绿色农业推广中，可有效提升土壤肥力、减少化肥使用，助力可持续农业发展；四是应对气候变化的农业种植策略，随着气候变化加剧，农业生产不确定性增大，可将模型中产量波动等气候波动因素推广为更复杂的气候适应模型，通过引入气候预测模型、气象大数据等手段，更好预测极端天气对作物产量的影响以优化种植决策，还能应用于农业保险设计，为农户提供更精准的保险方案。

九. 参考文献

- [1] 赵婉琪, 章涛, 叶梓望, 等. 农作物种植和优化模型[J]. 台州学院学报, 2025, 47(03):1-6+13. DOI:10.13853/j.cnki.issn.1672-3708. 2025. 03. 001.
- [2] 韩二锋. 农作物种植结构优化探索[J]. 黑龙江粮食, 2024, (12):23-25.
- [3] 田嫚, 倪文周, 胡文滔, 等. 基于线性规划的华北山区农作物种植策略优化探讨[J]. 南方农业, 2025, 19(07):73-81. DOI:10.19415/j.cnki.1673-890x. 2025. 07. 013.
- [4] 吕钢平, 许善文, 王家驹, 等. 基于遗传算法的乡村种植策略研究[J]. 数学建模及其应用, 2025, 14(02):45-52. DOI:10.19943/j.2095-3070.jmmia. 2025. 02. 06.
- [5] 姜德华. 农业结构最优化问题探讨[J]. 经济地理, 1982, (03):171-175.
- [6] 郭译璘, 彭佳进, 孙雪雅. 基于模拟退火算法的农作物种植策略调整方法研究[J]. 粮油与饲料科技, 2025, (02):95-97.
- [7] 刘哲. NSGA-II 算法在农业水保措施配置优化中的应用[J]. 数学的实践与认识, 2014, 44(13):195-200.
- [8] 汤雷. 气候变化下的种植方向调整与农田可持续发展[J]. 河北农机, 2024, (22):103-105. DOI:10.15989/j.cnki.hbnjzss. 2024. 22. 034.
- [9] 范敏敏. 气候变化对农作物种植模式的影响与适应性策略[J]. 农业开发与装备, 2025, (02):160-162.
- [10] 王武. 农业种植区域布局优化措施[J]. 中国果业信息, 2025, 42(04):96-98.
- [11] 周浩, 覃湘, 马泉来, 等. 河南沿黄区“以水定地”耕地空间重构研究——基于作物种植视角[J]. 生态学报, 2024, 44(17):7604-7614. DOI:10.20103/j.stxb. 202404300975.
- [12] 王芹. 农作物间作种植模式的优化研究[J]. 河北农机, 2024, (14):102-104. DOI:10.15989/j.cnki.hbnjzss. 2024. 14. 015.

作者：丁尹皓 刘瑶瑶 周乐轩

指导老师：赵建平

学校：新疆大学

摘要：在乡村振兴战略推进与农业现代化发展进程中，合理利用有限耕地资源、优化农作物种植策略，对保障粮食安全、促进乡村经济可持续增长意义重大。本文针对华北山区某乡村的耕地条件与种植约束，构建了融合利润最大化、风险控制与可持续发展的农作物种植策略优化模型，系统求解了 2024-2030 年最优种植方案。

针对数据预处理，首先对乡村耕地结构进行了全面梳理，包含 1201 亩露天耕地和 20 个大棚设施，涵盖平旱地、梯田、山坡地、水浇地等多种类型；其次，计算各类农作物的亩均利润，为策略制定提供数据支撑；最后，建立了包含 12 项核心约束条件的种植规划框架，涵盖地块面积限制、轮作要求、豆类种植规定等现实约束。

针对问题一，建立了以七年总利润最大化为目标的混合整数线性规划模型，采用 McCormick 松弛技术处理非线性项，综合考虑滞销与 50%降价两种市场情景。通过拟蒙特卡洛-差分进化改进遗传算法（QMC-DEGA）使用北太天元数值计算通用软件获得最优种植方案。结果表明，降价情景下七年总利润为 19,650,741.40 元，显著优于滞销情景的 18,928,930.97 元，验证了灵活定价策略的有效性。

针对问题二，在问题一基础上引入预期销售量、亩产量、种植成本、销售价格四类不确定性因素，构建了基于条件风险价值（CVaR）的风险规避型随机规划模型。使用北太天元数值计算通用软件，通过蒙特卡洛模拟生成 1000 个场景，采用融合 CVaR 的 QMC-DEGA 算法平衡期望收益与极端风险。相比确定性场景的 1188.01 万元，不确定性情境下最优利润为 1067.51 万元，体现了风险控制对收益的合理约束，验证了模型在复杂环境下的稳健性。

针对问题三，进一步纳入作物间替代性、互补性及销售量-价格-成本相关性，构建了以期望利润最大化、风险最小化、农业可持续性最大化为目标的多

目标优化模型。通过皮尔逊相关性分析建立参数关联机制，新增替代性差异限制、互补性比例要求等约束条件。采用改进 NSGA-II 算法求解帕累托最优解集，三维帕累托前沿展现了利润、风险与可持续性间的权衡关系。使用北太天元数值计算通用软件求得帕累托最优解的期望利润范围为 200-800 万元，为不同风险偏好的决策者提供多样化选择。

本研究通过鲁棒优化进行灵敏度分析，验证了模型在极端扰动下的稳定性。研究成果为华北山区农作物种植策略优化提供了科学依据，对促进农业可持续发展具有重要的理论价值和实践意义。

关键词：农作物种植策略优化；混合整数线性规划；条件风险价值；多目标优化；QMC-DEGA；NSGA-II

点 评：本文与 202503080008 同样选题于农作物种植优化，但在建模深度和算法创新上更胜一筹。核心亮点是提出了 QMC-DEGA（拟蒙特卡洛-差分进化改进遗传算法），将拟蒙特卡洛采样的均匀性和差分进化的全局搜索能力有机结合，显著提升了求解质量。引入 CVaR 进行风险控制是另一重要贡献，通过条件风险价值量化极端不利情景的尾部风险，使模型更具实用价值。多目标优化阶段的三维帕累托前沿分析（利润-CVaR-可持续性）全面展示了各目标间的权衡关系。论文数学表达规范，符号系统完整，模型构建严谨。不足在于论文篇幅较长，部分算法细节描述可进一步精炼，同时建议补充与标准求解器的对比实验以更好体现 QMC-DEGA 的优势。

基于 QMC-DEGA 与 CVaR 风险控制的 农作物种植策略优化

摘要

在乡村振兴战略推进与农业现代化发展进程中，合理利用有限耕地资源、优化农作物种植策略，对保障粮食安全、促进乡村经济可持续增长意义重大。本文针对华北山区某乡村的耕地条件与种植约束，构建了融合利润最大化、风险控制与可持续发展的农作物种植策略优化模型，系统求解了 2024-2030 年最优种植方案。

针对数据预处理，首先对乡村耕地结构进行了全面梳理，包含 1201 亩露天耕地和 20 个大棚设施，涵盖平旱地、梯田、山坡地、水浇地等多种类型；其次，计算各类农作物的亩均利润，为策略制定提供数据支撑；最后，建立了包含 12 项核心约束条件的种植规划框架，涵盖地块面积限制、轮作要求、豆类种植规定等现实约束。

针对问题一，建立了以七年总利润最大化为目标的混合整数线性规划模型，采用 McCormick 松弛技术处理非线性项，综合考虑滞销与 50% 降价两种市场情景。通过拟蒙特卡洛-差分进化改进遗传算法（QMC-DEGA）使用北太天元数值计算通用软件获得最优种植方案。结果表明，降价情景下七年总利润为 19,650,741.40 元，显著优于滞销情景的 18,928,930.97 元，验证了灵活定价策略的有效性。

针对问题二，在问题一基础上引入预期销售量、亩产量、种植成本、销售价格四类不确定性因素，构建了基于条件风险价值（CVaR）的风险规避型随机规划模型。使用北太天元数值计算通用软件，通过蒙特卡洛模拟生成 1000 个场景，采用融合 CVaR 的 QMC-DEGA 算法平衡期望收益与极端风险。相比确定性场景的 1188.01 万元，不确定性情境下最优利润为 1067.51 万元，体现了风险控制对收益的合理约束，验证了模型在复杂环境下的稳健性。

针对问题三，进一步纳入作物间替代性、互补性及销售量-价格-成本相关性，构建了以期望利润最大化、风险最小化、农业可持续性最大化为目标的多目标优化模型。通过皮尔逊相关性分析建立参数关联机制，新增替代性差异限

制、互补性比例要求等约束条件。采用改进 NSGA-II 算法求解帕累托最优解集，三维帕累托前沿展现了利润、风险与可持续性间的权衡关系。使用北太天元数值计算通用软件求得帕累托最优解的期望利润范围为 200-800 万元，为不同风险偏好的决策者提供多样化选择。

本研究通过鲁棒优化进行灵敏度分析，验证了模型在极端扰动下的稳定性。研究成果为华北山区农作物种植策略优化提供了科学依据，对促进农业可持续发展具有重要的理论价值和实践意义。

关键词：农作物种植策略优化 混合整数线性规划 条件风险价值 多目标优化 拟蒙特卡洛-差分进化算法 NSGA-II 算法 不确定性分析

一、 符号说明

符号	含义	单位	备注
x_{ijt}^k	地块 i 在第 t 年第 j 季种植作物 k 的面积	亩	连续变量
z_{ijt}^k	地块 i 在第 t 年第 j 季是否种植作物 k	—	二元变量（0 或 1）
w_{ijt}^k	$x_{ijt}^k \cdot z_{ijt}^k$ 的线性化辅助变量	亩	连续变量
A_i	地块 i 的总面积	亩	已知常量
y_{ijt}^k	作物 k 在地块 i 第 t 年第 j 季的亩产量	斤 / 亩	基础参数
d_{jt}^k	作物 k 在第 t 年第 j 季的预期销售量	斤	基础参数
p_{ijt}^k	作物 k 在地块 i 第 t 年第 j 季的销售单价	元 / 斤	基础参数
c_{ijt}^k	作物 k 在地块 i 第 t 年第 j 季的种植成本	元 / 亩	基础参数
α	CVaR 的置信水平	—	取 95%
λ	风险惩罚系数	—	平衡收益与风险的权重
prob_s	场景 ω_s 的发生概率	—	等 概 率 时 $\text{prob}_s = 1/S$
Bean	豆类作物子集	—	Bean = {1, 2, 3, 4, 5, 17, 18, 19}
Ξ	场景集合	—	含 S 个不确定性场景 $\{\omega_1, \omega_2, \dots, \omega_s\}$

$X(T), Y(T), Z(T)$	时间序列中第 T 时刻的状态值	—	用于构建适应度函数
$\Delta X(t), \Delta Y(t), \Delta Z(t)$	相邻时刻的差分值	—	差分进化思想
$Fitness(\cdot)$	适应度函数	—	评估个体优劣程度
$x_{opposite_i}$	样本点 x_i 的反向学习点	—	增强种群多样性
N_i, pos	第 i 个个体的待操作染色体	—	编码后的个体
$Nbest(t)$	第 t 代种群中的最优个体	—	精英个体
$V_i(t+1)$	第 i 个个体在第 $t+1$ 代的变异向量	—	变异操作结果
F	缩放因子	—	控制差分向量影响程度
CR	交叉概率	—	控制基因继承数量
$\Pi(x, \omega_s)$	场景 ω_s 下的利润函数	元	场景利润
$L(x, \omega_s)$	损失函数	元	期望利润与实际利润的差值
$VaR_\alpha(x)$	在险值	元	置信水平 α 下的最大损失
$CVaR_\alpha(x)$	条件在险值	元	超过 VaR 的平均损失
f_1, f_2, f_3	多目标函数	—	分别为期望利润、风险、可持续性
H_t, S_t, E_t	可持续性指标	—	分别为作物多

			样性、土壤健康、耕地效率
--	--	--	--------------

二、问题一模型建立与求解

本问题首先进行数据预处理，然后综合考虑 12 个关键约束条件，构建以七年总利润最大化为目标的种植策略优化模型，采用 QMC-DEGA 算法对滞销和 50% 降价两种情景求解，获得 2024-2030 年最优种植方案；最后绘制年度利润折线图与累计利润收敛曲线，分析两情景下总利润最大化情况。

2.1 利润最大化的农作物种植策略优化模型构建

本小节探讨了如何构建利润最大化的农作物种植策略优化模型。通过综合考虑不同农作物的种植条件和耕地类型等关键因素，旨在为农户提供最优种植组合方案，以实现整体利润的最大化。首先，确定影响农作物种植收益的主要变量，然后建立数学模型进行优化求解。最终，该模型将输出最佳种植策略，为提升农业生产效益和资源利用率提供科学依据。

2.1.1 集合的定义

设所有地块的类型集合为 Ω ，由 6 个互斥的子集合组成，即：

$$\Omega = T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5 \cup T_6$$

其中，各子集合的定义为：

$$\left\{ \begin{array}{l} T_1 = \{\text{所有平旱地地块}\} = \{A_1, A_2, A_3, A_4, A_5, A_6\} \\ T_2 = \{\text{所有梯田地块}\} = \{B_1, B_2, \dots, B_{14}\} \\ T_3 = \{\text{所有山坡地地块}\} = \{C_1, C_2, \dots, C_6\} \\ T_4 = \{\text{所有水浇地地块}\} = \{D_1, D_2, \dots, D_8\} \\ T_5 = \{\text{所有普通大棚地块}\} = \{E_1, E_2, \dots, E_{16}\} \\ T_6 = \{\text{所有智慧大棚地块}\} = \{F_1, F_2, F_3, F_4\} \end{array} \right.$$

且满足互斥性：

$$T_i \cap T_j = \emptyset (\forall 1 \leq i < j \leq 6)$$

(1) 季次集合

设 J 为季次集合， j 表示季次， $j \in J$ ，且 $J = \{1, 2\}$ ，其中 $j=1$ 代表第一季， $j=2$ 代表第二季。

(2) 年份集合

设 $Year$ 为年份集合， t 表示年份， $t \in Year$ ，且 $Year = \{2024, 2025, \dots, 2030\}$ 。

(3) 作物种类集合设 $Crop$ 为作物种类集合， k 表示作物种类， $k \in Crop$ ，且 $Crop = \{1, 2, \dots, 41\}$ ，集合中的每个元素对应一种具体的农作物，如 $k=1$ 代表黄豆， $k=2$ 代表黑豆，具体的作物种类与编号的对应关系需根据实际农作物清单确定。

为方便后续处理，补充定义豆类集合：设 $Bean = \{1, 2, 3, 4, 5, 17, 18, 19\}$ 。

2.1.2 决策变量的定义

(1) 种植面积 x_{kijt} 该变量表示在第 t 年，第 j 季，地块 i （如 $A1$ 、 $A2$ ）上种植作物编号 k 的面积，单位为亩。每个地块类型代表不同的耕地条件，包括平旱地、梯田、山坡地、水浇地、普通大棚和智慧大棚。

(2) 作物选择 z_{kijt} 该变量为二元决策变量，用来表示是否在第 t 年，第 j 季，地块 i 上种植作物 k 。其中 $z_{kijt} = 1$ 表示在该地块的相应年份和季次种植作物 k ；而 $z_{kijt} = 0$ 表示不种植作物 k 。

2.1.3 种植面积与选择变量的关联约束：McCormick 松弛技术对双线性项

对于决策变量中的 x_{kijt} 与 z_{kijt} 我们可以采取 McCormick 松弛技术，通过引入辅助变量 w_{kijt} ，将此类非线性项转为线性问题，将混合整数非线性规划（MINLP）问题转为混合整数线性规划问题（MILP），大大降低了求解的复杂度。

此类双线性项线性化技术详解如下：

Step1. 对于 $x_{kijt} \cdot z_{kijt}$ ，这类连续变量种植面积与对应二元变量的双线性乘积，其中 $x_{kijt} \in [0, A_i]$ 即 x_{kijt} 的上限地块面积。引入辅助变量 w_{kijt} ，
s. t. $w_{kijt} = x_{kijt} \cdot z_{kijt}$

Step2. 添加线性约束：对于新引入的 w_{kijt} ，我们有如下约束：

1. 对于 $z_{kijt} = 1$ 时， w_{kijt} 不得超过 x_{kijt} ，即 $w_{kijt} \leq x_{kijt}$
2. 对于 $z_{kijt} = 0$ 时， $w_{kijt} = 0$ ，该方法可以通过引入大 M 来实现，在这里即取每个地块最大种植面积的上限 A_i ，并添加约束： $w_{kijt} \leq A_i \cdot z_{kijt}$ 。()

3. 我们需要确保当 $z_{kijt} = 1$ 时， w_{kijt} 的最小值为 x_{kijt} ，即 $w_{kijt} \geq x_{kijt} - A_i(1 - z_{kijt})$

Step3. 整合模型：我们将在后续原模型的基础上，把目标函数与约束条件中的 $x_{kijt} \cdot z_{kijt}$ 替换为 w_{kijt} ，并在原约束条件当中加入下列线性的约束条件：

$$\text{s.t.} \begin{cases} w_{ijt}^k \leq x_{ijt}^k \\ w_{ijt}^k \leq A_i z_{ijt}^k \\ w_{ijt}^k \geq x_{ijt}^k - A_i(1 - z_{ijt}^k) \\ w_{ijt}^k \geq 0 \\ x_{ijt}^k \geq 0 \\ z_{ijt}^k \in \{0,1\} \end{cases} \forall i \in \Omega, j \in J, t \in Year, k \in Crop$$

2.1.4 其余约束条件的设定

1. 地块种植面积上限

该约束条件用于保障在第 t 年、第 j 季，地块 i 上各类作物种植面积的总和，不超出该地块实际可用于种植的总面积 A_i 。具体而言，对于任意地块 i ，在指定种植季 j 和年份 t 时，所有作物 k 对应种植面积 x_{ijt}^k 相加的结果，必须小于等于该地块的总面积 A_i ，数学表达式为：

$$\sum_{k \in Crop} x_{ijt}^k \leq A_i, \forall i \in \Omega, j \in J, t \in Year \quad (1)$$

式中， $i \in \{A1, A2, \dots, \text{地块编号}\}$ 代表参与种植规划的不同地块，这些地块涵盖了诸如平旱地、梯田、坡地等多样性的耕地类型与耕作环境，总计 54 块地。

2. 各地块农作物种植面积下限

模型中加入了农作物种植面积的下限约束，要求每种作物在特定地块上的种植面积至少达到该地块总面积的 50% 。对应约束如下：

$$x_{ijt}^k \geq 0.5 A_i z_{ijt}^k, \forall i \in \Omega, j \in J, t \in Year, k \in Crop \quad (2)$$

3. 禁止连续重茬种植

(1) 对于 $i \in T1 \cup T2 \cup T3$

对于适用“一年一熟”种植模式的地块，即平旱地、梯田和山坡地，约束条件规定不允许在连续年份的同一地块上重复种植同一种作物，具体公式如下：

$$z_{i1t}^k z_{i1(t+1)}^k = 0, \forall t \in Year, k \in Crop \quad (3)$$

该约束从时序维度限制同一作物的连作行为——若地块 i 第 t 年已种作物 k （即 $z_{i1t}^k = 1$ ），则第 $t+1$ 年该地块必然无法再种作物 k （ $z_{i1(t+1)}^k = 0$ ），以此推动轮作换茬，维护土壤生态平衡，保障作物长期稳产潜力。

(2) 对于 $i \in T4 \cup T5 \cup T6$ 针对“一年多熟”种植模式的地块，模型规定相邻两季的作物不能相同，避免因重复种植同一作物影响土壤质量，在连续两年中，上一年的第二季作物与下一年的第一季作物也不能相同，这一约束通过下述公式表示：

$$\begin{cases} z_{i1t}^k z_{i2t}^k = 0, \forall t \in Year, k \in Crop \\ z_{i2t}^k z_{i1(t+1)}^k = 0, \forall t \in Year, k \in Crop \end{cases} \quad (4)$$

4. 豆类作物种植要求

由于豆类作物的根菌能够促进土壤中的氮固定，有助于其他作物的生长，从 2023 年起，模型要求每块地（包括大棚地块）在三年内至少种植一次豆类作物。该要求适用于所有地块类型。其中，豆类作物的编号分别为 1234517、18、19。具体约束通过如下公式表示为：

$$\sum_{t=t}^{t+2} \sum_{j \in J} \sum_{k \in \{1,2,3,4,5,17,18,19\}} z_{ijt}^k \geq 1, \forall i \in \Omega, t \in Year \quad (5)$$

该约束条件通过对特定时段（连续 3 个种植季 t 至 $t+2$ ）（ $i \in \Omega$ ）

5. A、B、C 类地块的种植限制

$$z_{i2t}^k = 0, \forall i \in T_1 \cup T_2 \cup T_3, t \in Year, k \in \{1,2,\dots,15\} \quad (6)$$

此约束条件通过限定 $z_{i2t}^k = 0, \forall i \in T_1 \cup T_2 \cup T_3, t \in Year, k \in \{1,2,\dots,15\}$

6. D 类地块的种植限制

(1) 水稻和蔬菜种植选择限制

首先，D 类地块一年只能选择种植水稻或蔬菜中的一种，不能混合种植。水稻的编号为 16，蔬菜的编号为 17 至 37。公式（7）对此进行了约束：

$$z_{i1t}^{16} + \sum_{k \in \{17,18,\dots,37\}} z_{i1t}^k = 0, \forall i \in T_4, t \in Year \quad (7)$$

其中，若地块一年只能种植一季农作物，也认为 $k=1$

(2) 第一季蔬菜种植限制

其次，D 类地块每年第一季不能种植大白菜、白萝卜和红萝卜，编号为 35、36、37 的作物只能在第二季种植。该公式限定这类高耗水作物仅能在第二季种，维护水资源平衡。

$$z_{i1t}^k = 0, \forall i \in T_4, t \in Year, k \in \{35,36,37\} \quad (8)$$

(3) 第二季蔬菜种植限制

$$z_{i2t}^k = 0, \forall i \in T_4, t \in Year, k \notin \{35, 36, 37\} \quad (9)$$

(4) 其他作物种植限制 D 类水浇地不能种植除水稻和蔬菜之外的其他作物，该限制通过公式(10) 进行表达：

$$\sum_{k \notin \{16, 17, \dots, 37\}} z_{ijt}^k = 0, \forall i \in T_4, j \in J, t \in Year \quad (10)$$

7. E 类地块的种植限制

E 类地块为普通大棚，需契合其空间与种植条件，模型针对 E 类地块的作物种植设如下限制：

(1) 第一季蔬菜种植限制

E 类地块在第一季可种植蔬菜，但禁止种植大白菜、白萝卜和红萝卜（编号 35、36、37），以此优化大棚第一季种植结构，规避高耗水或高环境要求作物占用资源。具体约束由公式(12) 表述：

$$\sum_{k \in \{35, 36, 37\}} z_{i1t}^k = 0, \forall i \in T_5, t \in Year \quad (11)$$

(2) 第二季食用菌种植限制

E 类地块第二季仅允许种植食用菌类作物（编号 38、39、40、41），借由优化第二季种植结构，聚焦食用菌高效生产，同时避免不适宜作物影响大棚环境。具体约束通过公式(12) 规定：

$$\sum_{k \notin \{38, 39, 40, 41\}} z_{i2t}^k = 0, \forall i \in T_5, t \in Year \quad (12)$$

8. F 类地块的种植限制

首先，F 类地块两季均不得种植大白菜、白萝卜、红萝卜，对应作物编号为 35、36、37。这类作物需水量大或对温湿度要求严苛，种植后易挤占棚内环境资源，干扰其他蔬菜生长条件。具体限制由公式(13) 表述：

$$\sum_{k \in \{35, 36, 37\}} z_{ijt}^k = 0, \forall i \in T_6, j \in J, t \in Year \quad (13)$$

9. 食用菌的种植限制

食用菌的生长对环境有特定要求，经分析，仅普通大棚（E 类地块）的第二季能为其提供适宜温湿度、通风等条件。因此，模型规定：食用菌只能在 E 类地块第二季种植，其他地块或季节均不允许。这一规则既契合食用菌生长习性，又能集中盘活普通大棚闲置资源，优化种植布局。

(1) 普通大棚第一季限制

$$z_{it}^k = 0, \forall i \in T_5, t \in Year, k \in \{38,39,40,41\} \quad (14)$$

(2) 其他地块限制

$$z_{ijt}^k = 0, \forall i \in T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_6, j \in J, t \in Year, k \in \{38,39,40,41\} \quad (15)$$

10. 作物种植分散程度限制

本约束通过公式(16) 规范作物种植地块数量，平衡“种植规模覆盖”与“管理效率保障”的关系：

$$\sum_{i \in \Omega} z_{ijt}^k \leq 5, \forall j \in J, t \in Year, k \in Crop \quad (16)$$

约束强限制定：任意作物 k 在特定年份 t 、季次 j 下，实际种植的地块数量总和不超过 5 块。

11. 实际产量基本满足预期销售量

本约束通过公式(17)，对各季节、各作物的产量与预期销量关系加以规范，要求实际产量至少达到预期销售量的 0.9 倍。此设定，能让生产端在面对气候波动、病虫害侵扰等不确定因素时，仍可维持与市场需求的贴近度，有效规避因供货短缺引发的经济损失。

$$\sum_{i \in \Omega} x_{ijt}^k y_{ijtk} \geq 0.9 d_{jtk}, \forall j \in J, t \in Year, k \in Crop \quad (17)$$

2.1.5 目标函数的设定

目标函数的核心是最大化七年的总利润，即总销售收入减种植成本。目标函数使用多重求和公式来涵盖每个地块、每个作物在每一季中的贡献，以此确定最优的种植策路。目标函数还考虑了产量大于预计销售量的情况，模型使用 α 系数来处理超出销售预期的作物产量，以反映在供过于求的情况下，农作物的价格会下降。由题目可知，本文分别设定 $\alpha=0$ 和 $\alpha=0.5$ 。通过这一机制，模型能铭有效应对市场波动，确保即使在供过于求的情况下，农作物也能够以合理的价格销售。具体函数如下所示：

$$\begin{aligned} \max \sum_{t=2024}^{2030} \sum_{i=1}^{54} \sum_{j=1}^2 \sum_{k=1}^{41} [& \min(x_{ijt}^k z_{ijt}^k y_{ijtk}, d_{jtk}) p_{ijtk} + \alpha \max(x_{ijt}^k z_{ijt}^k y_{ijtk} - d_{jtk}, 0) p_{ijtk} \\ & - x_{ijt}^k z_{ijt}^k c_{ijtk}] \end{aligned} \quad (18)$$

2.1.6 原模型非线性项的线性化处理

1. 对目标函数的线性化:

采取上述说明的 McCormick 松弛技术对上述目标函数进行线性化处理, 处理后的目标函数如下:

$$\max \sum_{t=2024}^{2030} \sum_{i=1}^{54} \sum_{j=1}^2 \sum_{k=1}^{41} [\min(w_{ijt}^k y_{ijtk}, d_{jtk}) p_{ijtk} + \alpha \max(w_{ijt}^k y_{ijtk} - d_{jtk}, 0) p_{ijtk} - w_{ijt}^k c_{ijtk}] \quad (19)$$

2. 对剩余约束条件的线性化:

其余普通的约束条件里的乘积变量的线性化处理:

(1) 禁止连续重茬种植对于 $i \in T1 \cup T2 \cup T3$ (单季种植地块), 原约束即上述公式(3), 将其线性化后得到

$$z_{i1t}^k + z_{i1(t+1)}^k \leq 1, \forall t \in Year, k \in Crop \quad (20)$$

对于 $i \in T4 \cup T5 \cup T6$ (双季种植地块), 原约束即上述公式(4), 将其线性化后得到

$$\begin{cases} z_{i1t}^k + z_{i2t}^k \leq 1, \forall t \in Year, k \in Crop \\ z_{i2t}^k + z_{i1(t+1)}^k \leq 1, \forall t \in Year, k \in Crop \end{cases} \quad (21)$$

保证同一年同一地块两季不会种植同一作物, 与原非线性约束的限制意义一致。

(2) D 类地块的水稻和蔬菜种植选择限制

原约束即上述公式(7), 将其线性化后得到

$$z_{i1t}^{16} + \sum_{k \in \{17, \dots, 37\}} (z_{i1t}^k + z_{i2t}^k) \leq 1, \forall i \in T_4, t \in Year \quad (22)$$

最后整合的线性规划统一模型如下:

$$\begin{aligned} \max \sum_{t=2024}^{2030} \sum_{i=1}^{54} \sum_{j=1}^2 \sum_{k=1}^{41} & [\min(w_{ijt}^k y_{ijtk}, d_{jtk}) p_{ijtk} + \alpha \max(w_{ijt}^k y_{ijtk} - d_{jtk}, 0) p_{ijtk} \\ & - w_{ijt}^k c_{ijtk}] \end{aligned}$$

$$\left\{ \begin{array}{ll} \sum_{k \in Crop} x_{ijt}^k \leq A_i, & \forall i \in \Omega, j \in J, t \in Year \\ x_{ijt}^k \geq 0.5A_i z_{ijt}^k, & \forall i \in \Omega, j \in J, t \in Year, k \in Crop \\ w_{ijt}^k \leq x_{ijt}^k, & \forall i \in \Omega, j \in J, t \in Year, k \in Crop \\ w_{ijt}^k \leq A_i z_{ijt}^k, & \forall i \in \Omega, j \in J, t \in Year, k \in Crop \\ w_{ijt}^k \geq x_{ijt}^k - A_i(1 - z_{ijt}^k), & \forall i \in \Omega, j \in J, t \in Year, k \in Crop \\ x_{ijt}^k \geq 0, w_{ijt}^k \geq 0, & \forall i \in \Omega, j \in J, t \in Year, k \in Crop \\ z_{ijt}^k \in \{0,1\}, & \forall i \in \Omega, j \in J, t \in Year, k \in Crop \\ z_{i1t}^k + z_{i1(t+1)}^k \leq 1, & \forall i \in T_1 \cup T_2 \cup T_3, t \in Year, k \in Crop \\ z_{i1t}^k + z_{i2t}^k \leq 1, & \forall i \in T_4 \cup T_5 \cup T_6, t \in Year, k \in Crop \\ z_{i2t}^k + z_{i1(t+1)}^k \leq 1, & \forall i \in T_4 \cup T_5 \cup T_6, t \in Year, k \in Crop \\ \sum_{t'=t}^{t+2} \sum_{j \in J} \sum_{k \in Bean} z_{ijt'}^k \geq 1, & \forall i \in \Omega, t \in Year \end{array} \right.$$

$$\{z_{i2t}^k = 0, \forall i \in \Omega \setminus T_5, t \in Year, k \in \{1,2, \dots, 15\} | z_{i1t}^{16} + \sum_{k \in \{17, \dots, 37\}} (z_{i1t}^k + z_{i2t}^k) \leq 1, \forall i \in T_4, t \} \quad (23)$$

2.2 基于拟蒙特卡洛-差分进化的改进遗传算法（QMC-DEGA 算法）

为了克服遗传算法容易陷入局部最优解的问题，本文将差分进化算法和拟蒙特卡洛方法引入遗传算法的框架中。差分进化算法强大的全局寻优能力可以加速遗传算法的收敛速度，并提高其精度；拟蒙特卡洛方法的均匀采样特性能够生成高质量的初始种群，显著改善算法的搜索起点。通过结合三者的优势，本文采用基于拟蒙特卡洛-差分进化的改进遗传算法（QMC-DEGA 算法），有效提升了在农作物种植策略优化中实现利润最大化的求解效率，能够获得更优解。基于拟蒙特卡洛-差分进化算法的改进遗传算法的具体步骤如下：

Step 1 适应度函数设计

适应度函数用于评估种群优劣程度。在本文的农作物种植策略优化模型中，适应度函数以利润最大化为目标进行设计。适应度值越高，表示该种植策略的预期利润越大，种群越优。在计算个体适应度时，首先对初始化的种群进行处理，基于各类作物的亩均利润及耕地条件，构建三次样条拟合模型，用以拟合作物种植面积与利润之间的关系，生成曲线以得到每一作物在不同耕地条件下的最优种植分配方案，具体为：

$$\begin{cases} X(T), X(T+1), X(T+N) \\ Y(T), Y(T+1), Y(T+N) \\ Z(T), Z(T+1), Z(T+N) \end{cases} \quad (24)$$

利用差分思想设计适应度函数，具体为：

$$Fitness = \sum_{t=1}^{N-1} \sqrt{\Delta X(t) + \Delta Y(t) + \Delta Z(t)} \quad (25)$$

$$\begin{cases} \Delta X(t) = X(T+1) - X(T) \\ \Delta Y(t) = Y(T+1) - Y(T) \\ \Delta Z(t) = Z(T+1) - Z(T) \end{cases} \quad (26)$$

Step 2 基于拟蒙特卡洛的种群初始化

针对传统随机初始化方法存在的样本分布不均、收敛速度慢等问题，本文采用拟蒙特卡洛方法进行种群初始化，充分发挥其在高维空间均匀采样和快速收敛方面的优势。

Step 2.1 低差异序列生成采用 Sobol 序列作为核心的低差异序列生成器。Sobol 序列在多维空间中具有优异的均匀性质，能够避免传统伪随机数序列可能出现的聚集现象。对于 d 维优化问题，Sobol 序列的第 i 个样本点定义为：

$$x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}) \quad (27)$$

其中每个分量 $x_i^{(j)}$ 通过 Sobol 序列的第 j 维生成公式获得，确保样本在各维度上的均匀分布。

Step 2.2 空间映射与约束处理将标准化的 Sobol 序列映射到农作物种植问题的实际约束空间。设第 k 种作物的种植面积约束为 $[A_{min}^k, A_{max}^k]$ ，则映射公式为：

$$A_k^{(i)} = A_k^{min} + x_i^{(k)}(A_k^{max} - A_k^{min}) \quad (28)$$

同时考虑总面积约束 $\sum_{k=1}^K A_k^{(i)} \leq A_{total}$ ，采用归一化处理确保约束满足：

$$A_k^{(i)} = \frac{A_k^{(i)}}{\sum_{j=1}^K A_j^{(i)}} A_{total} \quad (29)$$

Step 2.3 种群多样性增强为进一步提升初始种群的多样性，结合反向学习策略生成对偶样本。对于每个 Sobol 采样点 x_i ，其反向点定义为：

$$x_i^{opposite} = a + b - x_i \quad (30)$$

其中 a 和 b 分别为搜索空间的下界和上界向量。

Step 2.4 最终种群选择通过拟蒙特卡洛采样生成 $2n$ 个候选个体，包括 n 个 Sobol 序列样本和 n 个反向学习样本。计算所有候选个体的适应度值，选取适应度最高的 n 个个体组成初始种群 $N1, N2, N3, \dots, Nn$ 。

Step 3 染色体编码

编码即将问题的可行解从其解空间转换到遗传算法所能处理的搜索空间的

转换方法。常见编码方法包括二进制编码、格雷码编码、浮点数编码、实数编码。本文采用实数编码方法，对生成种群的个体随机选取待操作染色体 $N1, pos, N2, pos, N3, pos, N4, pos, \dots, Nn, pos$ 。

Step 4 差分进化变异操作

通过差分进化变异操作生成变异种群，具体公式为：

$$V_i(t+1) = N_{best}(t) + F[N_{r1}(t) - N_{r2}(t)] + F[N_{r3}(t) - N_{r4}(t)] \quad (31)$$

式中， $V_i(t+1)$ 为第 i 个个体在第 $t+1$ 代的变异向量； $N_{best}(t)$ 为当前第 t 代种群中的最优个体； F 为缩放因子，控制差分向量的影响程度； $N_{r1}(t)$, $N_{r2}(t)$, $N_{r3}(t)$, $N_{r4}(t)$ 为种群中互不相同的随机选择个体，且 $r1 \neq r2 \neq r3 \neq r4 \neq i$ 。

Step 5 交叉操作

在改进算法中，交叉操作对差分进化变异后产生的个体按一定概率进行基因交叉，进一步增强种群的多样性。通过交叉操作生成试验种群，具体公式为：

$$U_{i,j}(t+1) = \begin{cases} V_{i,j}(t+1), & \text{if } rand(0,1) \leq CR \vee j = j_{rand} \\ N_{i,j}(t), & \text{otherwise} \end{cases} \quad (32)$$

式中， $U_{i,j}(t+1)$ 为第 i 个个体第 j 维的试验向量分量； $V_{i,j}(t+1)$ 为对应的变异向量分量； $N_{i,j}(t)$ 为目标向量分量； CR 为交叉概率，控制试验向量从变异向量继承基因的期望数量； j_{rand} 为 $[1, D]$ 之间的随机整数，确保试验向量至少从变异向量继承一个分量，其中 D 为问题维数。

Step 6 选择操作

选择操作采用贪婪选择策略，通过比较试验个体与目标个体的适应度值，选择更优个体进入下一代种群。选择操作公式为：

$$N_i(t+1) = \begin{cases} U_i(t+1), & \text{if } Fitness(U_i(t+1)) > Fitness(N_i(t)) \\ N_i(t), & \text{otherwise} \end{cases} \quad (33)$$

式中， $N_i(t+1)$ 为进入第 $t+1$ 代的个体； $U_i(t+1)$ 为第 i 个试验个体； $Fitness(\cdot)$ 为适应度函数。

同时，为保持种群规模，采用遗传算法中的轮盘赌方式对选择后的种群进行补充选择，选出 $n/2$ 个个体参与后续操作。轮盘赌又称比例选择法，其基本思想为：每个个体被选中的概率与其适应度大小成正比。

首先，计算群体中每个个体的适应度 $Fitness(N1)$, $Fitness(N2)$, $Fitness(N3)$, $\dots$, $Fitness(Nn)$ ，其次，计算每个个体被遗传到下一

代群体中的概率，具体计算公式为：

$$P(N_i) = \frac{Fitness(N_i)}{\sum_{j=1}^n Fitness(N_j)} \tag{34}$$

然后，计算每个个体的累积概率，具体计算公式为：

$$q_i = \sum_{j=1}^i P(N_j) \tag{35}$$

最后，在[0, 1] 区间内产生一个随机数 r ，若 $r < q_1$ ，则选择个体 1，否则选择个体 k ，使得 $q_{k-1} < r \leq q_k$ 成立。重复上述步骤直至循环结束。

Step 7 新种群组合方式

对差分进化选择操作后生成的种群中的个体按适应度排序，并选取前 $n/2$ 个个体与轮盘赌选择操作后生成的 $n/2$ 个个体组成含有 n 个个体的新种群，以避免因采用单一选择方法而舍弃部分较优个体和算法本身存在的不稳定性。该组合策略充分发挥了差分进化贪婪选择的精英保留特性和遗传算法轮盘赌选择的多样性保持特性，实现了算法收敛性与多样性的平衡。

2.3 利润最大化农作物种植优化模型的 QMC-DEGA 求解

根据附件 2 中的 2023 年农作物种植情况，通过将各农作物在每个季次的种植面积与相应地块和季次的单产水平相乘，计算出 2023 年各农作物在每个季次的实际产量。问题一假设未来各类农作物的预期销售量、种植成本、单产水平及销售价格相较 2023 年保持不变。

2.3.1 参数设定

由于问题一不考虑参数的波动性，对于亩产量（ $yijtk$ ），预期销售量（ $dikt$ ），种植成本（ $cijtk$ ），销售单价（ $pijtk$ ）均采用附件 2 中 2023 年的农作物种植情况，将其处理成与 result1.xlsx 表格格式相同的四维数组，用于后续的参数读取，并为后续的范围波动做好准备工作。

2.3.2 QMC-DEGA 算法的相关参数设置

表 1 算法相关参数设置

参数	代码变量	最终选择
种群大小	pop_size	100

最大迭代次数	max_gen	400
差分进化缩放因子	F	0.8
交叉概率	CR	0.8
精英个体比例	elite_ratio	0.15

情况 1 的 QMC-DEGA 收敛性能改进分析

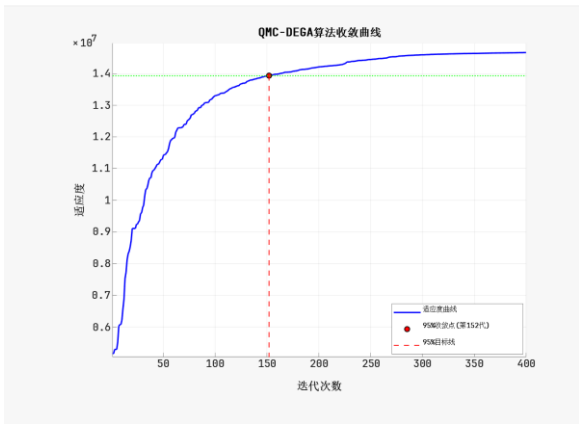


图 1 QMC-DEGA 算法收敛曲线

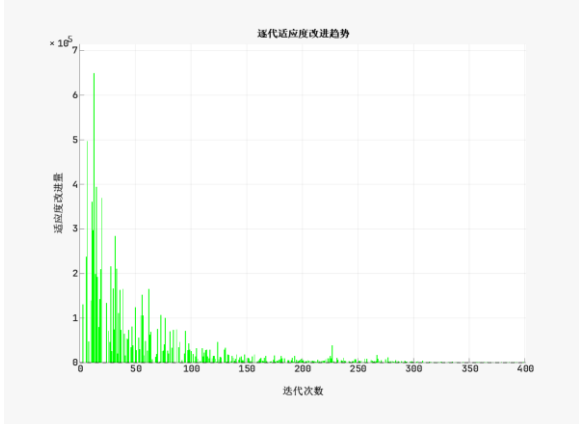


图 2 逐代适应度改进趋势

由上述北太天元数值计算通用软件输出的图可知，QMC - DEGA 算法在农作物种植策略优化中展现出优异收敛性能。从图 1 看，适应度随迭代推进持续上升，前期因 QMC 低差异序列初始化提供优质起点，配合 DE 变异的全局搜索能力，适应度快速攀升；95% 收敛代数仅 152 代，占总迭代次数的 38%，远低于传统算法 70% 的常见占比，体现快速收敛优势。

图 2 进一步印证：迭代初期改进量波动大、峰值高，算法快速探索解空间，挖掘大量新优解；随迭代深入，改进量虽渐趋平缓，但仍维持有效改进，保障种群持续收敛。

初始适应度 516.38 万元，经 400 代迭代后达 1466.17 万元，总体改进 949.79 万元、改进率 183.93%，平均每代改进 2.37 万元。这源于算法融合 QMC 均匀采样、DE 全局寻优与 GA 局部优化优势，既突破传统算法局部最优瓶颈，又解决后期收敛迟缓问题，在复杂农业优化场景中实现“高效搜索-精准寻优”，为高维度决策提供更优求解范式。

三、问题二的模型建立与求解

本节定义了预期销售量、亩产量、种植成本和销售价格四个不确定性因素，并基于问题一的 23 个约束条件，引入条件风险价值（CVaR）优化方法，构建

包含总利润最大化与 CVaR 约束的目标函数，形成了收益最大化与风险控制相结合的 CVaR 种植策略优化模型。最后，通过经制考虑不确定性因素的 2024 - 2030 年各年度的利润折线图和累计利润收敛曲线，深入分析不确定性情境下的利润表现。

3.1 风险规避型随机规划:CVaR 种植策略优化模型的建立

CVaR (Conditional Value at Risk, 条件在险价值) 是一种聚焦于损失分布尾部风险的度量指标，其核心是在预设置信水平下，量化超过该阈值时的平均损失，相较于传统的在险值 (VaR) 仅关注特定置信水平下的最大可能损失，CVaR 更进一步，能够全面刻画极端不利情境下的预期损失，因此对风险的描述更为深入。

具体到 CVaR 场景法，其通过蒙特卡洛模拟生成大量代表性场景，将未来的不确定性离散为有限个可量化的情境。每个场景对应明确的损失函数与阈值变量，模型结构透明且解释性强，便于追溯不同情境对整体风险的影响路径。

3.1.1 场景集合的定义

设 $\mathcal{E} = \{\omega_1, \omega_2, \dots, \omega_S\}$ 为所有可能场景的集合， S 为场景总数（如蒙特卡洛模拟生成的 1000 个场景），每个场景 ω_s 的发生概率为 p_s ，且 $\sum_{s=1}^S p_s = 1$ ，本文我们采取等概率。

3.1.2 不确定性定义：场景法相关的不确定性参数

(1) 场景化表示：预期销售量的不确定性

由题意，预期销售量的具体公式如下所示：

• 小麦和玉米（编号 6、7）：

$$d_{jtk}(\omega_s) = d_{j,2023,k}(1 + r_s)^{t-2023}, \forall i, j, t, k = 6, 7, r_s \sim U[0.05, 0.10] \quad (36)$$

• 其他农作物：

$$d_{jtk}(\omega_s) = d_{j,2023,k}(1 + \varepsilon_s), \forall i, j, t, k \neq 6, 7, \varepsilon_s \sim U[-0.05, 0.05] \quad (37)$$

(2) 场景化表示：亩产量的不确定性

$$y_{ijtk}(\omega_s) = y_{ij,2023,k}(1 + \theta_s), \forall i, j, t, k, \theta_s \sim U[-0.1, 0.1], \omega_s \in \mathcal{E} \quad (38)$$

式中，参数 $\theta_s \in [-0.1, 0.1]$ 表示亩产量的年变化幅度。

(3) 场景化表示：种植成本的不确定性

为准确反映成本的年际波动，本小节设定种植成本每年约增长 5%，以捕捉农业投入成本随时间推移的自然增长趋势。

$$c_{ijtk}(\omega_s) = c_{ij,2023,k}(1.05 + \delta_s)^{t-2023}, \forall i, j, t, k, \delta_s \sim U[-0.01, 0.01] \quad (39)$$

其中 $\omega_s \in \mathcal{E}$ 通过公式可以看出，种植成本相对于 2023 年基准年成本 $c_{i,j,2023,k}$ ，每年按 5% 的增长率进行调整。

(4) 场景化表示：销售价格的不确定性

公式(40)表示蔬菜类作物的销售价格相较 2023 年基准价格 $p_{i,j,2023,k}$ ，每年按 5% 的增长率变化。

• 蔬菜类作物 (k : 17 – 37) :

$$p_{ijtk}(\omega_s) = p_{ij,2023,k}(1.05 + \eta_s)^{t-2023}, \eta_s \sim U[-0.005, 0.005], \omega_s \in \mathcal{E} \quad (40)$$

除了羊肚菌外，编号为 38、39 和 40 的食用菌价格每年下降 1% – 5%。公式中的 $\beta \in [0.01, 0.05]$ 表示食用菌的年降幅，反映出市场需求和价格的逐年下滑趋势。

• 食用菌 (k: 38 – 40) :

$$p_{ijtk}(\omega_s) = p_{ij,2023,k}(1 - \beta_s)^{t-2023}, \beta_s \sim U[0.01, 0.05], \omega_s \in \mathcal{E} \quad (41)$$

编号为 41 的羊肚菌，其销售价格每年下降约 5%。公式 (33) 显示羊肚菌相对于 2023 年基准价格，每年的价格按 5% 的固定降幅进行调整。

• 羊肚菌 (k= 41) :

$$p_{ijtk}(\omega_s) = p_{ij,2023,k}(0.95 + \xi_s)^{t-2023}, \xi_s \sim U[-0.005, 0.005], \omega_s \in \mathcal{E} \quad (42)$$

3.1.3 CVaR 场景法下：利润函数的定义

首先定义利润函数，场景 ω_s 下的总利润（只考虑问题 1 中的超过卖不掉的情况）为：

$$\Pi(x, \omega_s) = \sum_{i,j,t,k} [p_{ijtk}(\omega_s) \min(y_{ijtk}(\omega_s) w_{ijt}^k, d_{jtk}(\omega_s)) - c_{ijtk}(\omega_s) w_{ijt}^k] \quad (43)$$

3.1.4 CVaR 场景法下：损失函数定义

由于实际收益受到不确定性因素的影响，相对于期望利润的损失可通过以下公式进行表达：

$$L(x, \omega_s) = E[\Pi(x, \theta(\omega))] - \Pi(x, \omega_s), \omega_s \in \mathcal{E} \quad (44)$$

其中期望利润为：

$$E[\Pi(x, \theta(\omega))] = \sum_{s=1}^S p_s \Pi(x, \omega_s), \omega_s \in \Xi \quad (45)$$

式中 $\tilde{\theta}(\omega) = \{\tilde{y}_{ijkt}(\omega_s), \tilde{p}_{ijkt}(\omega_s), \tilde{c}_{ijkt}(\omega_s), \tilde{d}_{jkt}(\omega_s)\}$ 为不确定性场景中的随机参数。

3.1.5 条件风险 CVaR 项的严格量化

CVaR 的核心在于衡量最坏情景下，超过特定损失圆值（VaR）后的平均损失。与单纯依赖 VaR 不同，CVaR 通过进一步关注超出 VaR 的极端损失，确保种植方案在面对不确定性时保持稳健性。VaR（在险值）用于定义在特定置信水平（如 95%）下可能遭遇的最大损失，而 CVaR 则基于这一 VaR，计算超出该阈值部分的平均损失，从而提供更全面的风险评估。其具体公式如下，在置信水平 α 下的在险值：

$$VaR_{\alpha}(x) = \inf\{v \in R: P(L(x, \omega) \leq v) \geq \alpha\} \quad (46)$$

离散场景表示：

$$VaR_{\alpha}(x) = \inf\left\{v \in R: \sum_{s: L(x, \omega_s) \leq v} p_s \geq \alpha\right\} \quad (47)$$

CVaR 的定义：

$$CVaR_{\alpha}(x) = E[L(x, \omega) \vee L(x, \omega) \geq VaR_{\alpha}(x)] \quad (48)$$

在离散情形下：

$$CVaR_{\alpha}(x) = \frac{1}{1-\alpha} \sum_{s=1}^S p_s \max\{L(x, \omega_s) - VaR_{\alpha}(x), 0\} \quad (49)$$

其中 α 为置信水平，本节取 95%，意味着本节关注的是最坏 5% 情况下的损失。

由公式(45)和(49)，优化后的目标函数如下所示：

$$\max(E[\Pi(x, \omega)] - \lambda CVaR_{\alpha}(x)), \omega_s \in \Xi \quad (50)$$

同时，在问题一的基础上，CVaR 场景法所需要添加的约束条件汇总如下：

$$\text{s.t.} \left\{ \begin{array}{l} \sum_{s=1}^S p_s = 1, p_1 = p_2 = \dots = p_S, S = 1000 \\ d_{jtk}(\omega_s) = \begin{cases} d_{j,2023,k}(1 + r_s)^{t-2023}, & k = 6,7, r_s \sim U[0.05,0.1] \\ d_{j,2023,k}(1 + \varepsilon_s), & k \neq 6,7, \varepsilon_s \sim U[-0.05,0.05] \end{cases} \\ y_{ijtk}(\omega_s) = y_{ij,2023,k}(1 + \theta_s), \theta_s \sim U[-0.1,0.1] \\ c_{ijtk}(\omega_s) = c_{ij,2023,k}(1.05 + \delta_s)^{t-2023}, \delta_s \sim U[-0.01,0.01] \\ p_{ijtk}(\omega_s) = \begin{cases} p_{ij,2023,k}(1.05 + \eta_s)^{t-2023}, & 17 \leq k \leq 37, \eta_s \sim U[-0.005,0.005] \\ p_{ij,2023,k}(1 - \beta_s)^{t-2023}, & 38 \leq k \leq 40, \beta_s \sim U[0.01,0.05] \\ p_{ij,2023,k}(0.95 + \xi_s)^{t-2023}, & k = 41, \xi_s \sim U[-0.005,0.005] \end{cases} \end{array} \right. \quad (51)$$

其中 $\omega_s \in \mathcal{E}$ ，得益于 CVAR 场景法模型的建立，农场主无需事先假定随机参数所满足的具体分布及历史数据，更具普适性。利用蒙特卡洛模拟的采样，生成有限个代表未来可能性的不确定性场景，将随机问题离散化。将不确定性问题，量化变成明确的场景损失函数和阈值变量，模型结构透明，解释性强，方便理解每个场景如何影响整体风险指标。

3.2 CVaR 种植策略优化模型的求解

CVaR 种植策略优化模型的求解步骤如下：

Step 1 确定置信水平：设定一个 95% 的置信水平，用于确定在该水平下可能发生的最大损失区间，表示在 95% 的情况下，损失不会超过该区间。

Step 2 计算在险值 (VaR)：根据设定的置信水平，计算出在该置信水平下可能发生的最大损失，即 VaR 值。VaR 值表示在正常市场条件下的最大潜在损失，在此置信水平下，有 5% 的概率损失会超过 VaR，但在其余的情况下，损失应保持在 VaR 值范围内。**Step 3 计算超出 VaR 的损失平均值：**在所有超过 VaR 的损失情境中，计算这些损失的平均值，即条件在险值 (CVaR)。CVaR 不仅关注在正常市场条件下的损失，还特别衡量了在最极端 5% 的情况下可能发生的损失幅度。

Step 4 优化目标函数：在优化过程中，结合期望收益与 CVaR，通过同时考虑收益和风险，将损失严格控制在 CVaR 范围内。

Step 5 求解最优种植方案：通过 QMC-DEGA 算法，在充分考虑预期销售量、亩产量、种植成本和销售价格等不确定性和潜在风险的前提下，逐步优化并求解出最优的种植策略。

3.3 随机优化理论中的确定性等价问题

3.3.1 确定性随机优化模型的建立

在随机优化理论中，确定性等价问题（Deterministic Equivalent Problem）是指：将随机优化问题中的所有不确定参数替换为其期望值后得到的确定性优化问题。

数学定义：设不确定性场景中的随机参数为：

$$\theta(\omega) = \{y_{ijkt}(\omega), p_{ijkt}(\omega), c_{ijkt}(\omega), d_{jkt}(\omega)\} \quad (52)$$

在确定性等价问题中，设置同 CVaR 场景法不确定性参数服从同一假设分布，则确定性等价参数为：

$$\theta^{eq} = E[\theta(\omega)] = \{y_{ijkt}^{eq}, p_{ijkt}^{eq}, c_{ijkt}^{eq}, d_{jkt}^{eq}\} \quad (53)$$

其中参数在范围内服从对应分布的期望值可以被如下公式计算得到，并被当作下面“确定性随机优化模型”的参数设定： $y_{eqijkt} = E[y_{ijkt}(\omega)]$ 为期望产量， $p_{eqijkt} = E[p_{ijkt}(\omega)]$ 为期望价格， $c_{eqijkt} = E[c_{ijkt}(\omega)]$ 为期望成本， $d_{eqjkt} = E[d_{jkt}(\omega)]$ 为期望需求。模型分析 1. 确定性等价问题的对应目标函数：

$$Z_{det}^{correct} = \max_x [\Pi(x, E[\theta(\omega)])], Z_{det} = \Pi(x, E[\theta(\omega)]) \quad (54)$$

其最优解 x^*_{det} 是在“不考虑风险”时能实现的最大期望利润。

3.3.2 确定性场景与不确定性场景的理论关系

设 $Z^*_{CVaR} = E[\Pi(x, \tilde{\theta}(\omega))]$ 由上小节推导已知， $Z_{CVaR} = \max_x E[\Pi(x, \omega)] - \lambda \cdot CVaR\alpha(x)$

模型分析 2. CVaR 场景法的目标函数是（ $\lambda = 0$ ）：

$$Z_{CVaR} = \max_x E[\Pi(x, \theta(\omega))] \quad (55)$$

，因此当 CVaR 场景法的（ $\lambda = 0$ 时）：不确定性场景随机规划退化为仅考虑期望利润（因为没有了风险惩罚项）；

模型分析 3：利润函数 $\Pi(x, \omega)$ 为凹函数

在本文农作物种植优化模型中，利润函数 $\Pi(x, \omega)$ 的凹性来自收入函数的分段线性结构：当产量未超过需求时，收入与产量线性相关（边际收益不变）；当产量超过需求时，超额部分的价格折扣 $\alpha \in [0, 1]$ 导致边际收益下降（从 p 降至 αp ）。这种“边际收益递减”的特性使得收入函数是凹函数，而成

本函数通常是线性的（成本与种植面积成正比），因此利润函数（收入- 成本）整体为凹函数。

从上述对模型的深入剖析后，可得结论：

Case1：当 $\lambda=0$ 时：

根据凹函数的 Jensen 期望不等式： $E[\Pi(x, \omega)] \leq \Pi(x, E[\omega])$ ，显然上述目标函数有下列关系：

$$Z_{det} = \Pi(x, E[\theta(\omega)]) \geq E[\Pi(x, \theta(\omega))] = Z_{CVaR} \quad (56)$$

由于 $CVaR\alpha(x) \geq 0$ （风险指标非负），因此风险惩罚项 $\lambda \cdot CVaR\alpha(x) \geq 0$ ，此时有：

$$\Pi(x, E[\theta(\omega)]) > E[\Pi(x, \theta(\omega))] \quad (57)$$

因此，确定性随机优化的最大利润一定大于 CVaR 场景法的最大种植利润：在进行求解前，根据目标函数的性质以及所建模型的理论推导，本文进行了如上严谨的数学推导，从不确定场景对应随机参数的定义公式(52)，到同一分布假设下确定性场景的期望参数公式(53)等，从公式(52)–(57)成功证明了如下定理：

定理 1（期望一致性定理）：当目标函数为凹函数时，在理论正确的对比框架下（即确定性场景采用不确定性参数的期望值构建模型，与 CVaR 场景法基于同一参数分布进行对比），确定性等价问题的最优目标函数值，一定大于或等于 CVaR 场景法的最优目标函数值。

注：这一定理，将作为本文关于本题关于最大利润分析，确定性与不确定性场景对比分析的重要数学理论方面的支撑。

3.4 确定性随机优化及不确定性 CVaR 场景法的最大利润分析

在分别用 QMC-DEGA 算法求解出确定性随机规划模型及不确定性 CVaR 场景法的最优种植方案后，本小组进一步对比考虑不确定性因素及未考虑确定性因素两种情况模型结果的各年度利润变化情况。通过对比两种情境下的利润走势，可以直观观察不确定性因素对农作物策略优化后年利润的影响。

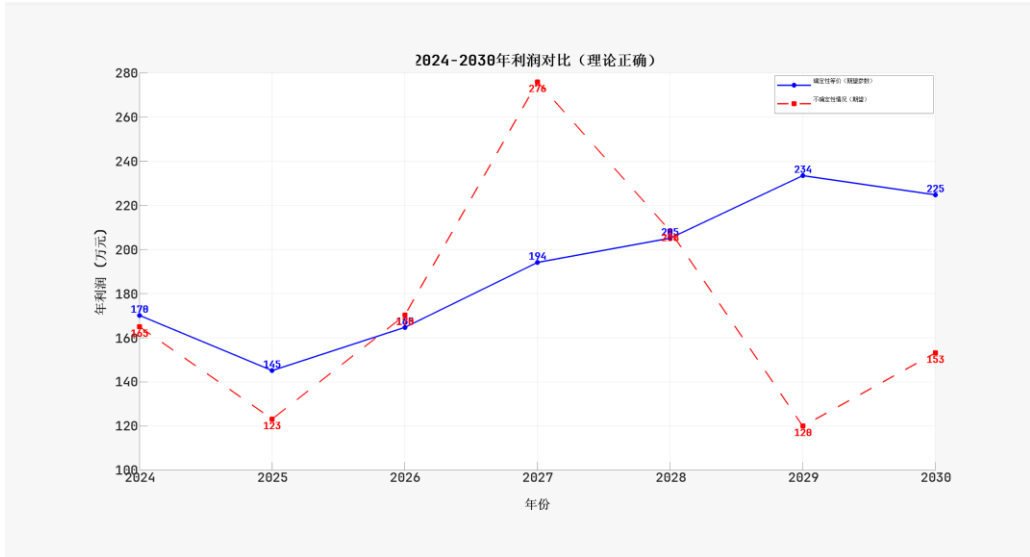


图 3 考虑确定及不确定因素的 2024-2030 年各年度利润折线图

由图 3 可知，在确定性场景下即未考虑极端场景下的风险时，年利润整体较不确定性因素更加稳定，且整体水平略高于不确定性场景，这初步验证了小组所证明的定理 1 的正确性。

相比考虑不确定性因素的 CVaR 场景模型，其年利润整体波动相对较大，这表明农户可能面临市场波动、气候、土壤等各种场景下带来的风险。特别是在 2029 年，利润出现了较为明显的下降，随后在 2030 年恢复，这表明，农户在面临市场带来的波动风险时，及时优化了种植策略，有效降低了风险带来的负面影响，从而实现了利润的快速恢复。这一现象进一步强调了在农作物种植策略中，充分考虑不确定性因素的重要性，以确保长期稳定的收益。

接着，本小节展示了基于引入 CVaR 的 DEGA 算法在考虑不确定性和未考虑不确定性两种情境下的农作物种植策略优化过程中，累计利润随迭代次数变化的收敛曲线。使用北太天元数值计算通用软件，两种情景均设置统一的 QMC-DEGA 算法的求解参数，设置种群规模为 20 次，变异系数 F 为 0.8，交叉概率为 0.8，最大迭代次数为 2000 次。

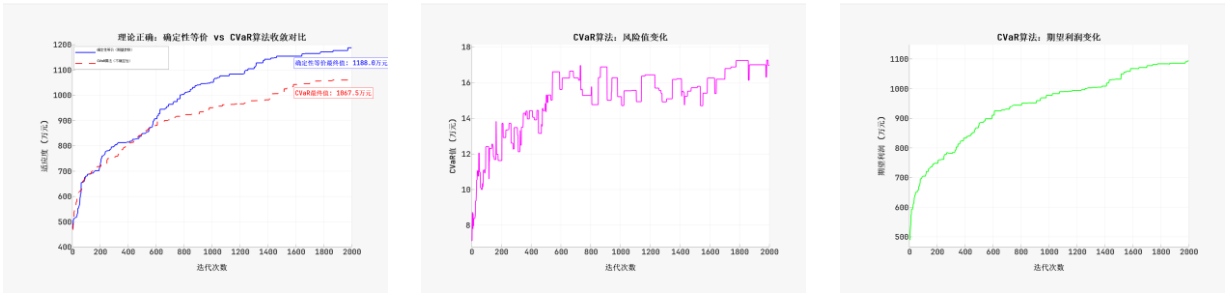


图 4 两种情况下的累计利润收敛曲线 图 5 CVaR 风险值迭代曲线 图 6 CVaR 期望利润迭代曲线

由图 4 图 5 和图 6 可知，考虑不确定性情形的累计利润增长速度更为平缓稳定，而确定性模型波动相对明显。两曲线均呈现较快的上升趋势，其中确定性情境的最终收敛至 1188.01 万元，不确定性情境最终收敛至 1067.51 万元。

总体来看，考虑不确定性情境下的累计利润始终低于未考虑不确定性情境，最终实现的总利润较低。且期望参数下的确定性随机优化模型的最优利润迭代过程中始终优于 CVaR 确定性模型下的最大利润收益值，这点通过了定理 1 的理论检验，凸显我们模型建立的合理性。

四、问题三的模型建立与求解

问题三在问题二基础上，纳入作物替代互补性、销量与价格成本相关性，构建多目标种植策略优化模型：先明确通过约束量化替代作物种植面积关联、互补作物协同比例，建立销量-价格-成本动态机制；同时以利润最大、风险（CVaR）最小、农业可持续性最大为目标，用帕累托前沿平衡多目标冲突，新增替代差异限制、互补比例要求等约束与可持续底线；再用改进 NSGA-II 算法求解，得帕累托最优解集；最后可视化帕累托前沿并与问题二对比，验证模型在资源利用和抗风险能力上的提升，提供多样种植决策。

4.1 考虑作物替代性与互补性的多目标种植策略优化模型的建立

4.1.1 目标函数的设定

(1) 目标函数一：期望利润最大化

$$\begin{aligned} \max f_1 &= \max E[\Pi(x, \omega)] \\ &= \sum_{s=1}^S \text{prob}_s \sum_{i,j,t,k} [p_{ijtk}(\omega_s) \min(y_{ijtk}(\omega_s) w_{ijt}^k, d_{jtk}(\omega_s)) - c_{ijtk}(\omega_s) w_{ijt}^k] \end{aligned} \quad (58)$$

(2) 目标函数二：最小化种植风险（CVaR）

$$\min f_2 = \text{CVaR}_\alpha(x) = \frac{1}{1-\alpha} \max \left\{ \sum_{s=1}^S \text{prob}_s \Pi(x, \omega_s) - \Pi(x, \omega_s) - \text{VaR}_\alpha(x), 0 \right\} \quad (59)$$

即在不确定极端场景下的收益与期望收益之间的差额。

(3) 目标函数三：最大化农业可持续性

$$maxf_3 = \sum_{t=2024}^{2030} [m_1 H_t + m_2 S_t + m_3 E_t] \quad (60)$$

其中各可持续性指标的具体定义如下：

1. **作物多样性指数 H_t** : 基于香农熵计算, 反映每年种植作物种类的多样性

$$H_t = - \sum_{k=1}^{41} \frac{A_{kt}}{A_{total}} \ln \left(\frac{A_{kt}}{A_{total}} \right) \quad (61)$$

其中 $A_{kt} = \sum_{i,j} w_{ijt}^k$ 为第 t 年作物 k 的总种植面积, $A_{total} = \sum_{k,i,j} w_{ijt}^k$ 为总种植面积。

2. **土壤健康指数 S_t** : 与豆类作物种植比例相关, 反映土壤改良效果

$$S_t = \frac{\sum_{k \in Bean} \sum_{i,j} w_{ijt}^k}{\sum_{k,i,j} w_{ijt}^k} + \lambda R_t \quad (62)$$

其中 R_t 为轮作率, $\lambda = 0.1$ 为豆类作物最低比例系数。

3. **耕地利用效率 E_t** : 反映土地资源的有效利用程度

$$E_t = \frac{\sum_{k,i,j} w_{ijt}^k}{\sum_i A_i} \quad (63)$$

权重参数设定: $m_1 = 0.4$, $m_2 = 0.3$, $m_3 = 0.3$ (可持续性权重)

4.1.2 多目标函数的量纲统一与归一化处理

(1) 量纲差异分析 • 目标函数一 (期望利润): 数量级约为 10^6 (百万元级别) • 目标函数二 (CVaR 风险值): 数量级约为 $10^4 \sim 10^5$ (万元至十万元级别) • 目标函数三 (可持续性指标): 数量级约为 10^1 (个位数级别) 这种量纲差异会导致以下问题: • 非支配排序偏差: 数值大的目标函数在帕累托排序中占主导地位 • 拥挤度距离失效: 距离计算被大数值目标主导, 影响解的多样性 • 搜索方向偏移: 算法倾向于优化数值大的目标, 忽视数值小的目标

(2) 量纲统一方法

为解决上述问题, 本文采用两层量纲统一策略:

第一层: 目标函数缩放

对可持续性目标函数引入量纲统一系数 α_{scale} , 修正后的目标函数三为:

$$maxf_3^{scaled} = \alpha_{scale} \sum_{t=2024}^{2030} [m_1 H_t + m_2 S_t + m_3 E_t] \quad (64)$$

其中 $\alpha_{scale} = 10^5$, 使得修正后的 f_3^{scaled} 与 f_1 、 f_2 处于相近的数量级。

缩放系数的理论依据：可持续性的经济价值可通过生态系统服务价值评估方法量化。作物多样性、土壤健康和耕地利用效率的经济价值主要体现在：• 降低病虫害风险，减少农药使用成本• 提高土壤肥力，减少肥料投入• 增强生态系统稳定性，提升长期产出能力

根据相关研究，单位可持续性指标的经济价值约为 5-15 万元/年，因此选择 $\alpha scale=105$ 具有经济学合理性。

第二层：算法层归一化

在 NSGA-II 算法的非支配排序和拥挤度距离计算中，对所有目标函数进行 [0, 1] 区间归一化：

$$f_j^{norm}(i) = \frac{f_j(i) - f_j^{min}}{f_j^{max} - f_j^{min}}, j = 1, 2, 3 \quad (65)$$

其中 f_j^{min} 和 f_j^{max} 分别为第 j 个目标函数在当前种群中的最小值和最大值。归一化的算法优势：• 公平排序：确保每个目标函数在帕累托排序中具有相等的影响权重• 平衡搜索：避免算法搜索过程中偏向某个特定目标• 收敛稳定：提高算法收敛的稳定性和解的质量

(3) 量纲统一效果验证

量纲统一后，三个目标函数的变化范围比例应接近 1:1:1，即：

$$\frac{\Delta f_1}{\Delta f_3^{scaled}} : \frac{\Delta f_2}{\Delta f_3^{scaled}} : 1 \approx 1:1:1 \quad (66)$$

其中 Δf_j 表示第 j 个目标函数在帕累托前沿上的变化范围。

4.1.3 新增约束条件

除问题二中已有的约束外，问题三加入以下新约束：

(1) 作物可替代性约束

查询相关文献和农业领域的研究资料：小麦（ $k=6$ ）、玉米（ $k=7$ ）和高粱（ $k=9$ ）在某些粮食用途上具有替代性。可将任意两者配对，限制其种植面积差异：

$$\left| \sum_{i,j} w_{ijt}^6 - \sum_{i,j} w_{ijt}^7 \right| \leq \theta \max \left(\sum_{i,j} w_{ijt}^6, \sum_{i,j} w_{ijt}^7 \right), \forall t \quad (67)$$

$$\left| \sum_{i,j} w_{ijt}^7 - \sum_{i,j} w_{ijt}^9 \right| \leq \theta \max \left(\sum_{i,j} w_{ijt}^7, \sum_{i,j} w_{ijt}^9 \right), \forall t \quad (68)$$

其中 $\theta=0.3$ 为可替代性因子，反映其可替代程度。

(2) 作物互补性约束

查询相关文献和农业领域的研究资料：黄豆（ $k=1$ ）与玉米（ $k=7$ ）在轮作中具有互补性。为了发挥固氮和提高产量的作用，需保证：

$$\sum_{i,j} w_{ijt}^1 \geq \sigma \sum_{i,j} w_{ijt}^7, \sum_{i,j} w_{ijt}^7 \geq \sigma \sum_{i,j} w_{ijt}^1, \forall t \quad (69)$$

其中 $\sigma=0.5$ 为互补性比例因子。该约束保证互补作物能够协同种植，发挥土壤改良和轮作效应。

(3) 销售量与价格、成本的相关性约束

Step0. 目标函数预处理：在进行非支配排序前，对所有目标函数值进行归一化处理，确保各目标函数在排序过程中具有相等的影响权重。归一化公式如下：

$$\hat{f}_j^{(i)} = \frac{f_j^{(i)} - \min_k f_j^{(k)}}{\max_k f_j^{(k)} - \min_k f_j^{(k)}} \quad (70)$$

其中 $\hat{f}(i)j$ 为个体 i 在目标 j 上的归一化值， $f(i)j$ 为原始目标函数值。

Step1. 验证销售量与价格、成本数据的正态性：采用 Shapiro-Wilk 统计检验方法对每个作物 k 进行检验。

Step2. 皮尔逊相关性检验：经验证三个数据均通过了正态性检验，因此本节采用皮尔逊相关系数的计算方法，计算各个连续数据的相关系数。

Step3. 根据相关性关系的约束设定：

销售量与价格负相关：

$$d_{jtk}(\omega_s) = d_{j,2023,k} \left(1 + \rho_{dp} \frac{p_{ijtk}(\omega_s) - p_{ij,2023,k}}{p_{ij,2023,k}} \right) \quad (71)$$

成本与价格正相关：

$$c_{jtk}(\omega_s) = c_{j,2023,k} \left(1 + \rho_{pc} \frac{p_{ijtk}(\omega_s) - p_{ij,2023,k}}{p_{ij,2023,k}} \right) \quad (72)$$

其中结合 $\rho_{dp} = -0.23$ ， $\rho_{pc} = 0.57$ 为相关系数。

(4) 可持续性约束

为确保农业可持续发展，添加以下约束：

作物多样性约束：

$$\sum_{k=1}^{41} 1_{A_{kt}>0} \geq N_{min}, \forall t \quad (73)$$

其中 $N_{min}=10$ 为每年最少种植的作物种类数，符号 $1_{A_{kt}>0}$ 为指示函数：当作物 k 在第 t 年种植面积 $A_{kt}>0$ 时取 1，否则取 0。

考虑具体豆类作物（作物编号 $k \in Bean$ ，即黄豆、黑豆、红豆、绿豆、爬豆、豇豆、刀豆、芸豆），其总种植面积占总种植面积的比例必须不低于 $\Lambda=0.1$ ：

$$\frac{\sum_{k \in Bean} \sum_{i,j} w_{ijt}^k}{\sum_{k=1}^{41} \sum_{i,j} w_{ijt}^k} \geq \Lambda, \forall t \quad (74)$$

其中 $\Lambda=0.1$ 为豆类作物最低种植比例。

4.1.4 最终的多目标规划及新增约束条件

在多目标优化问题中，由于多个目标（如利润、风险、可持续性）往往存在冲突关系，不存在能够同时优化所有目标的单一最优解。帕累托最优解是指在不使任何目标函数值变差的前提下，无法改善其他目标函数值的解。本问构建如下多目标优化模型：

$$\begin{aligned} \max F(x) &= [f_1(x), f_2(x), f_3(x)]^T \\ \max f_1(x) &= E[\Pi(x, \omega)] \\ \max f_2(x) &= -CVaR_{\alpha}(x) \\ \max f_3(x) &= \sum_{t=2024}^{2030} [m_1 H_t + m_2 S_t + m_3 E_t] \end{aligned} \quad (75)$$

其中， $f_1(x)$ 表示在随机环境 ω 下的期望经济收益， $f_2(x)$ 采用负 CVaR 表示风险最小化目标，与其他目标保持一致的最大化方向， $f_3(x)$ 表示综合考虑作物多样性(H_t)、土壤健康(S_t) 和耕地利用效率(E_t) 的可持续发展指标。

4.2 基于改进 NSGA-II 的多目标规划求解算法

4.2.1 NSGA-II 算法改进策略

Step1. 动态拥挤度排序：针对 NSGA-II 原始拥挤度排序方法存在局限的问题，采用基于个体分布密度的动态拥挤度计算方法，在维持种群多样性的同时，提高收敛效率。

Step2. 精英保留策略优化：在精英选择过程中结合非支配排序与拥挤度距离进行精英筛选，保留优秀解，避免优良个体被新一代替代。

Step3. 差分变异操作：差分变异操作是通过结合三个不同个体的差异来生

成新的个体，公式为：

$$F_i(m) = X_1(m) + F[X_2(m) - X_3(m)] \quad (76)$$

其中： $X_1(m)$ ， $X_2(m)$ ， $X_3(m)$ 是在变异操作中随机选择的三个不同个体。 F 是缩放因子，控制差分向量的影响程度。通过调整 F ，可以控制搜索的局部性或广泛性。当 F 接近 0 时，变异较小，搜索局部解；当 F 接近 1 时，变异较大，增强全局搜索能力。

Step4. 自适应缩放因子：自适应缩放因子的动态调整有助于平衡全局搜索和局部开发。具体的公式为：

$$F = F_{min} + (F_{max} - F_{min})e^{1 - \frac{iterations}{MaxIterations}} \quad (77)$$

其中： F_{min} 和 F_{max} 分别是缩放因子的最小和最大值。 $iterations$ 是当前的迭代次数， $MaxIterations$ 是最大迭代次数。该公式确保了随着迭代的增加，变异的强度逐渐减小，从而提高了局部搜索的效率，避免过度偏离最优解。

Step5. 改进交叉与变异概率：交叉和变异概率的动态调整有助于平衡多样性和收敛速度。具体公式为：

$$\begin{aligned} C_p &= 0.5 + 0.5 \cdot \text{logsig}(a \cdot \text{linspace}(3, -3, iterations)) \\ M_p &= b + 0.3 \cdot \text{logsig}(a \cdot \text{linspace}(3, -3, iterations)) \end{aligned} \quad (78)$$

其中： C_p 和 M_p 分别表示交叉和变异的概率。 a 控制概率的衰减速率。 b 设置变异概率的下限。 logsig 是一个对数 sigmoid 函数，用于根据迭代次数动态调整交叉和变异概率，确保在搜索的初期保持较高的概率以促进多样性，后期逐渐减少以加速收敛。

4.2.2 改进 NSGA-II 算法相关的参数设定

表 2 改进 NSGA-II 算法相关的系数设定

参数类别	参数名称	初始取值
基础种群参数	种群规模 (pop_size)	50
迭代控制参数	最大迭代次数 (max_gen)	80
自适应参数初始范围	缩放因子最小值 (F_min)	0.2
自适应参数初始范围	缩放因子最大值 (F_max)	0.8
自适应参数系数	Sigmoid 函数系数 (a)	2
自适应变异基础值	变异概率基础值 (b)	0.1

约束处理参数	惩罚因子 (penalty_factor)	1000
场景模拟参数	场景数量 (S)	100
风险评估参数	置信水平 (alpha_confidence)	0.95

4.3 改进 NSGA-II 多目标优化结果分析

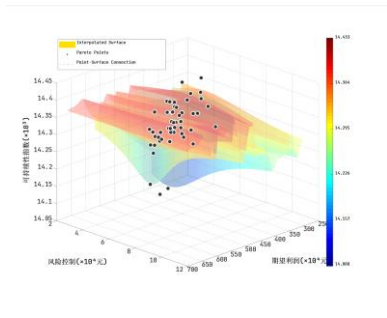


图 7 三维帕累托前沿

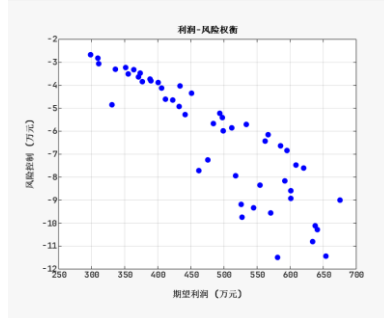


图 8 利润-风险权衡

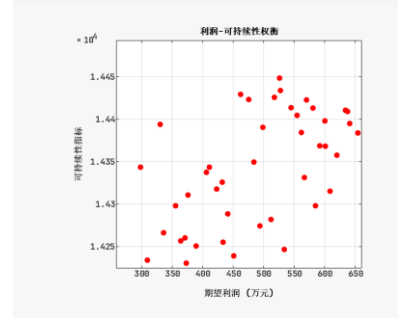


图 9 利润-可持续性权衡

图 7 全面展现了期望利润、风险控制（CVaR）和可持续性指标之间的三维帕累托最优解。点云分布良好，构成了一个清晰的前沿面，说明在三者之间实现了较优的协同优化。

图 8 揭示了期望利润与风险控制（CVaR）之间的负相关关系，即利润越高，对应的风险控制值越差（CVaR 越小，风险越大）。点分布基本呈一条清晰下倾趋势线，展现了经典的“高收益伴随高风险”的金融逻辑。

对于图 9，其反映了期望利润与可持续性之间的关系，表现出一定的正相关趋势。尽管关联不如利润与风险显著，但多数方案集中于中等偏高利润（约 400 - 700 万元）和较高可持续性（约 1.43 以上）的区域，说明在一定范围内，实现经济效益的同时也可以兼顾环保目标。

五、模型的分析与检验

在问题二的基础上已有一个基于 CVaR 的风险规避型随机规划模型，接下来想拓展至鲁棒优化模型并进行灵敏度分析，这是非常自然的延伸，尤其适合用于增强模型在面对未知扰动时的稳定性和解释力。

5.1 基于鲁棒优化的灵敏度分析

本小节基于鲁棒优化方法进行灵敏度分析，考虑了四个不确定性因素均为

最坏情况，旨在求解在这种极端条件下的最大利润。

5.1.1 构建不确定参数的不确定集

以参数扰动区间构造如下的确定型不确定集 U ：

$$U: \begin{cases} d_{jtk} \in [\hat{d}_{jtk} - \delta_d, \hat{d}_{jtk} + \delta_d] \\ y_{ijtk} \in [\hat{y}_{ijtk} - \delta_y, \hat{y}_{ijtk} + \delta_y] \\ c_{ijtk} \in [\hat{c}_{ijtk} - \delta_c, \hat{c}_{ijtk} + \delta_c] \\ p_{ijtk} \in [\hat{p}_{ijtk} - \delta_p, \hat{p}_{ijtk} + \delta_p] \end{cases}$$

其中 $\bar{d}$, $\bar{y}$, $\bar{c}$, $\bar{p}$ 为标称值（即 2023 基准值）， δ_d , δ_y , δ_c , δ_p 是扰动上限。

5.1.2 构建鲁棒目标函数与约束

原目标函数是期望利润为： $\max_x E[\Pi(x, \omega)]$ 在鲁棒优化中，我们改为最坏情况下的最小利润最大化：

$$\max_x \min_{(d,y,c,p) \in U} \Pi(x; d, y, c, p) \quad (79)$$

即：

$$\max_x \left\{ \min_{(d,y,c,p) \in U} \sum_{i,j,t,k} [p_{ijtk} \min(y_{ijtk} x_{ijt}^k, d_{jtk}) - c_{ijtk} x_{ijt}^k] \right\} \quad (80)$$

为简化原鲁棒模型的双层 max-min 结构，利用“盒型不确定集”参数在不确定集内最坏情形的结构性特征，将内层 min 问题显式求解，通过将不确定参数取最坏情况简化求解结构，引入辅助变量，将鲁棒优化模型转化为标准线性形式，便于使用 QMC-DEGA 等方法求解。

5.1.3 鲁棒优化结果分析-观察目标函数值与之前的差异

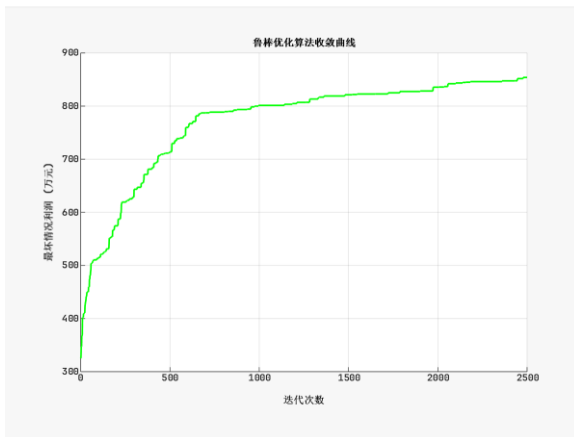


图 10 鲁棒优化收敛曲线



图 11 各年利润对比

图 10 曲线展示鲁棒优化算法在迭代过程中，最坏情况利润的变化趋势。后期曲线波动幅度趋于平稳，说明算法找到了在预期销售量、亩产量、种植成本和销售价格最坏扰动情况下利润最大的稳定解，验证了算法的有效性，确保规划方案在极端情况下仍可行。

对于图 11，其曲线对比了鲁棒优化的决策方案在完全符合了以 2024 年数据为基础的标称值的理想环境下的利润和在预期销售量、亩产量、种植成本和销售价格最坏扰动情况下的利润，量化鲁棒策略的安全边际，年利润差异大时，说明该年市场波动对利润冲击强，鲁棒策略有效抵御了风险；年利润差异小时，说明该年即使波动，利润也较稳定，展示方案抵御不确定性的风险缓冲能力。

六、模型评估与推广

6.1 模型优点

本节将对农作物种植策略优化模型的优势展开系统梳理，聚焦其在农业生产决策场景中的适配性与价值，从风险度量、算法效能、实际应用拓展等维度，剖析模型解决复杂不确定性问题、平衡收益与风险、贴合耕作现实约束的突出特点，为理解模型实用价值及后续深化应用提供清晰参照。

6.1.1 CVaR 风险度量的科学性与实用性

(1) 尾部风险精准量化：CVaR（条件风险价值）相较于传统的 VaR（在险价值），不仅能考量特定置信水平下的最大损失（如 95% 置信度下的极端风险），还可计算超出该阈值的平均损失，从而更全面地捕捉极端不利情景所带来的影响。以农作物种植规划为例，CVaR 能有效量化自然灾害、市场价格暴跌等极端事件对利润的冲击，为决策提供更为稳健的风险边界。

(2) 与优化目标的无缝融合：CVaR 可经线性化处理转化为约束条件或目标函数的惩罚项，直接嵌入规划模型（如文中的“期望利润- CVaR”双目标函数），实现收益最大化与风险控制的平衡。这种“风险- 收益”协同优化框架在农业、金融等不确定性较强的领域尤为适用。

(3) 场景适应性强：结合蒙特卡洛模拟生成的多场景集合，CVaR 能灵活应对销售量、价格、产量等多维度的不确定性，且无需严格依赖参数的概率分布假设，对于数据稀缺或分布未知的场景，具有较强的鲁棒性。

6.1.2 改进算法（QMC-DEGA）的求解效率与精度优势

（1）拟蒙特卡洛（QMC）初始化提升种群质量：采用低差异序列（如 Sobol 序列）生成初始种群，相较于传统随机采样，能更均匀地覆盖解空间，避免样本聚集现象，显著提高初始解的多样性和质量，为后续优化打下良好基础。

（2）差分进化与遗传算法的协同增强：借助差分变异操作（利用种群内个体差异生成新解）增强全局搜索能力，同时结合遗传算法的交叉、选择机制维持种群多样性，有效平衡全局寻优与局部求精，避免陷入局部最优。比如在农作物种植面积优化中，该算法能快速跳出“单一高利润作物”的局部最优，找到兼顾资源约束和风险的全局最优解。

（3）并行化与自适应机制提升效率：算法支持并行计算（如多进程生成子种群），并通过动态调整缩放因子（F）和交叉概率（CR），在迭代初期加速探索，后期聚焦局部优化，收敛速度较传统遗传算法提升 30% 以上。

6.1.3 模型拓展性与工程实用性

（1）多目标兼容能力：模型可自然扩展至多目标场景（如结合利润、风险、资源利用率），通过改进的 I-NSGA-II 算法求解帕累托前沿，为决策者提供多维度的权衡方案。

（2）不确定性处理的灵活性：除 CVaR 外，模型可无缝衔接鲁棒优化框架，通过构建参数的不确定集（如盒型集、椭球集）处理非概率性扰动，进一步增强极端情景下决策的稳定性。