从校园问题到数学模型

论文结构与问题拆解 · Axiom · gh & wjs

Contents

课程目录

从问题理解到模型建立，再到数值求解与论文表达的系统性学习路径。

01 理解问题：交付清单与动词解码

02 建立模型：校园场景数学化

03 数值方法：算不出来怎么办

04 检验与表达：模型验证与论文写作

CHAPTER 01

理解问题

交付清单与动词解码

数学建模的第一步不是理解背景故事，而是识别题目到底让你交什么

PROBLEM DECOMPOSITION

第一件事：找交付清单，不读背景故事

数学建模的第一步不是理解背景故事，而是识别题目要求的交付物和约束条件。通过提取 Background、Requirements、Deliverables 和 Constraints 四个要素，将模糊描述转化为明确的建模目标，避免在无关信息上浪费时间。

01 · Background

情境与任务

回答"现实世界发生了什么"，仅提供情境支持，不直接决定建模方向，如"学校早晨拥挤"只是背景而非问题本身

Context

02 · Requirements

情境与任务

明确"需要交付什么"，是建模的核心驱动力，如"计算最优入口数量""比较不同分流方案""预测等待时间分布"

Core Driver

03 · Deliverables

产出与边界

指出除论文外是否需提交代码、数据表或可视化图表，直接影响模型的输出形式和技术实现路径

Output Form

04 · Constraints

产出与边界

界定模型的自由程度，如预算限制、物理空间限制、时间窗口限制，这些边界条件直接决定决策变量的可行域

Feasible Region

VERB TAXONOMY

题目动词决定答案的"形状"

竞赛题中的动词直接定义了最终答案的形式：calculate 要求具体数值，compare 要求多方案对比，optimize 要求目标函数与最优解，predict 要求未来情景外推。识别动词是选择数学工具的前提，先判断答案形状再决定建模方法。

01 · 计算与对比

Calculate：要求精确数值结果，如"计算平均等待时间""求解最优窗口数"，输出通常是一个数或数值区间

Compare：要求建立多方案评价体系，明确比较维度和权重，输出通常是排序、排名或权衡分析图表

数值 · 排名

02 · 优化与预测

Optimize：需要构建目标函数（如最小化等待时间）和约束条件（如预算上限），寻找全局或局部最优解

Predict：涉及时间序列外推或情景模拟，必须包含不确定性分析和置信区间，而非单一数值预测

最优解 · 置信区间

03 · 讨论与验证

Discuss / Recommend：比较不同可能性并说明取舍，recommend 必须明确给出"我们推荐 X 因为 A、B、C"的结论

Validate：不是重新算一次，而是拿模型预测结果与已知数据、实验结果或现实情况进行对照验证

取舍 · 对照

Chapter 02

建立模型：校园场景数学化

把食堂排队、教室灯光、跑步路线这些日常问题翻译成数学语言

CAMPUS MODELING

"身边问题"其实都是建模题

校园日常场景如食堂排队、教室灯光、跑步路线等，本质上都是可以用数学建模解决的优化问题。

食堂与教室

研究到达率、服务率与等待时间，用排队论分析"增加窗口是否值得"

平衡自然光与能耗，用函数拟合求解最佳开灯策略

排队论

路线与通勤

加权图中寻找多重约束的最优跑步路径，属于图论与多目标优化

500 名学生同时出发，用离散模型与仿真求解调度方案

图论

运动与表现

用回归与动力学模型揭示训练量、恢复时间与运动表现的关系，为科学训练提供量化依据

通过历史数据拟合预测最佳参赛状态窗口，优化备赛周期安排

回归分析

模型的语法：四类变量

数学模型由四类不同角色的变量构成：决策变量（可控输入）、状态变量（系统状态）、参数（外生估计量）和常数（固定值）。分清它们的角色是建模的基本功，不是所有出现在公式里的数字都属于同一种东西。

Decision Variables

决策变量

我们可以主动选择的量，如食堂窗口数 **k**、发车间隔 **T**，其可行域由约束条件界定，最优解就在这些变量中寻找。决策变量是优化问题的核心求解对象。

可控输入

State Variables

状态变量

描述系统在时刻 **t** 的内部状况，如排队人数 **N(t)**、剩余座位数 **S(t)**，随时间演化并受决策变量影响。状态变量反映系统的动态特征。

系统状态

Parameters

参数

需标定但不可控的外生变量，如学生到达率 **λ**、服务率 **μ**、预算上限 **B**，通过数据统计或文献调研获得。参数决定了模型的具体情境。

外生估计量

Constants

常数

不随情形改变的固定数值，如一天 24 小时、光速 **c**、**π** 值，在模型中作为已知常量处理。常数为计算提供基准参照。

固定值

MODELING FRAMEWORK

一个模型至少回答三个问题

> *任何数学模型都需要回答三个核心问题：What can I change（决策空间）、What can I measure（可观测数据）、What am I trying to optimize/predict（目标函数）。三者共同构成模型的骨架。*

DECISION SPACE

Q1: 我能控制什么?

界定决策空间：明确可控的调节手段与约束边界，如食堂窗口数 k 的取值范围为 1 到 10，这是模型中唯一能主动改变的量。

$k \in [1, 10]$

OBSERVABLE DATA

Q2: 我能观察什么?

确定数据来源：识别可观测的状态变量与参数，如每分钟到达人数 λ 可通过实地计数获得，排队长度 N(t) 可通过定时记录追踪。

$\lambda \cdot N(t)$

OBJECTIVE FUNCTION

Q3: 我想得到什么?

定义目标函数：量化优化方向与评价标准，如"最小化平均等待时间"或"在成本不超过预算的前提下最大化服务效率"。

$\text{Min} \cdot \text{Max}$

Chapter 03

数值方法：算不出来怎么办

插值、迭代与离散化——让高一学生第一次接触真正的计算数学

插值：两次测量之间怎么办？

> *插值是数值分析最基础的思想之一：根据已知的离散测量点，估计中间未知位置的值。精度随方法复杂度提升，核心逻辑始终是"用已知推断未知"。*

01 线性插值

已知 8:00 教室温度 20°C、10:00 温度 24°C，假设温度线性变化，则 9:00 温度可通过线性公式计算得 22°C

$$T(t) = T_0 + (T_1 - T_0) \cdot (t - t_0)/(t_1 - t_0)$$

核心假设是两点之间为直线，适用于变化平稳的场景

直线近似

02 从线性到非线性

当实际温度变化呈曲线（如中午出现峰值），线性插值误差增大，需引入多项式拟合、样条插值或回归分析等高级方法

方法选择的本质是精度与复杂度的权衡：采样点越多、模型越精细，估计越准确，但计算成本也随之增加

精度 × 成本

迭代：算不出来就"越来越靠近答案"

迭代是数值分析最值得掌握的思想之一：从一个初始猜测出发，通过反复修正逐步逼近真实答案。



初始猜测

从一个初始值 x_0 出发，作为逼近真实答案的起点

x_0



迭代修正

利用公式 $x_{n+1} = x_n - f(x_n)/f'(x_n)$ 不断更新，每次使解更精确

$x_{n+1} = x_n - f/f'$



生活迭代

试 3 小时学习→效率不够→调 4 小时→注意力降→收敛到 3.5h + 休息

3.5h



普适逻辑

不知道最优解但能判断好坏的问题，都可通过猜测→评估→修正逼近答案

Try · Evaluate · Adjust

离散化：让连续世界变成计算机能理解的东西

> *离散化是连接连续数学与计算机计算的关键桥梁：将连续问题切割成有限个离散小块，用有限差分近似导数，使原本无法求解的微分方程转化为可数值计算的代数方程组。*

01 教室温度变化是连续系统，每一秒都有温度值，但计算机无法处理"无限多个点"，必须将连续问题**离散为每小时一个数据点。**

连续→离散

02 有限差分近似导数：**dT/dt ≈ [T(t+h) − T(t)] / h**，其中 h 为
时间步长，步长越小精度越高但计算量越大。

有限差分

03 离散化使连续微分方程转化为代数方程组，计算机可以逐步求解，这是**数值分析连接现实系统与计算结果**的核心桥梁。

代数方程组

04 完整流程：**现实系统 → 数学模型 → 离散化 → 数值计算 → 结果验证**，这条路径适用于绝大多数无法解析求解的建模问题。

完整流程

Chapter 04

检验与表达

模型验证与论文写作

三重验证确保模型可靠，敏感性分析测试结论稳健，标准论文结构让评委一目了然

Validation Framework

模型三重验证：算完不等于做完

> *可靠的数学模型必须通过三重验证：量纲一致性检查确保公式逻辑正确，极端情况检验模型在边界条件下的行为合理性，真实数据验证评估预测精度。三重验证层层递进，是从"会算"到"真懂"的关键跨越。*

01 · 逻辑验证

量纲检查：验证方程两边单位一致性，如等待时间公式结果必须为分钟，若出现面积或速度单位则模型存在根本错误

极端情况检验：测试参数趋向边界时的模型行为，如学生数→0 时等待时间应→0，窗口数→∞ 时应趋近单个服务时间

02 · 实证验证

真实数据验证：将模型预测值（如平均等待 6.2 分钟）与实地测量值（6.5 分钟）对比，计算相对误差约 4.8%，在置信区间内则通过

Validation 是建模从"纸面推导"走向"解决现实问题"的分水岭，HiMCM 评审明确要求讨论模型的 **strengths** 与 **weaknesses**

SENSITIVITY ANALYSIS

敏感性分析：我的答案到底有多脆弱?

一个好的模型不仅要"算得准"，更要在参数扰动下保持结论稳健。

最优解验证：食堂最优窗口数 k=7，但当学生数量 +10%、服务时间 +5%、高峰提前 5 分钟后，最优解是否仍为 7？若变为 9 则模型敏感。

边际递减：等待时间 T(k) 随窗口数增加而递减，但增速放缓，第 7 个窗口的改善远大于第 10 个窗口。

鲁棒性标准：参数扰动 ±10% 后最优解不变则模型可信，剧烈波动则需重新评估假设或增加弹性设计。



Paper Structure

HiMCM 标准论文结构

数学建模竞赛论文遵循标准化结构，与 COMAP 评审标准完全对齐。论文不是模板填空，而是建模思维的自然展开——从摘要到结论，每个部分都对应建模闭环中的一个关键环节，确保逻辑严谨、表达清晰。

基础框架

01

Summary 四句话结构：明确问题、方法、关键数值结果、最终建议，避免写成 Introduction 缩写或泛泛而谈

02

Assumptions & Justifications：不仅列出简化假设，还需论证其合理性，需配合实地数据支撑

03

Model Development 分层递进：先建简化模型抓住主要矛盾，再逐步增加复杂度，最后说明数值求解方法

04

Notation 符号表：统一全文记号，每个符号标注含义、量纲和说明，放在建模页之前避免后文重复解释

验证与结论

05

Validation & Sensitivity：展示模型预测与真实数据对比，用图表呈现参数变化对结果的影响程度，证明模型稳健性

06

Strengths & Weaknesses：诚实讨论模型的优势与局限，展现批判性思维

07

Conclusion / Recommendation：给出明确可执行的建议，并注明结论的适用条件和风险

摘要四句话与建模闭环

摘要不是 Introduction 的缩写，而是一张"论文成绩单"，用四句话概括问题、方法、结果和建议。数学建模本质上是一个从现实出发最终回归现实的闭环——如果数学结果不能回答原始问题，模型就没有完成。模型不是终点，现实才是终点。

摘要四句话模板

01 **We investigated...** 明确研究对象与目标，如"高中食堂最优窗口数"

02 **We developed...** 简述核心模型与方法，如"排队论模型结合敏感性分析"

03 **The model predicts...** 给出最关键的具体数值，如"7 个窗口降至 6.3 分钟"

建模闭环流程

01 从现实问题出发 → 任务拆解 → 变量与假设 → 数学模型 → 数值计算 → 结果 → Validation → Sensitivity → 现实结论 → Recommendation

02 **闭环检验标准**：最终数学结果能直接回答最初的现实问题，且通过验证与敏感性测试，则建模闭环完成；否则需返回修正

Case Study

校园 Mini Case：食堂窗口优化

一个 5 分钟的小案例串起全课所有知识点：从决策变量识别到敏感性分析，证明数学建模就在我们身边。



问题与参数

决策变量 k=窗口数（最多 10 个），到达率 λ=30 人/分钟，单窗口服务率 μ=5 人/分钟。目标为最小化总成本（运营+等待成本），典型 optimize 问题。通过建立排队模型，将实际运营问题转化为可求解的数学优化问题。

λ=30 · μ=5 · k≤10



求解与优化

排队稳定性要求 5k>30，即 k>6，最少 7 个窗口。k=10 虽最小化等待但人力成本高，总成本函数 $C(k)=c_1 \cdot k+c_2 \cdot L(k)$ 解得最优 k 在 7-9 之间。权衡服务效率与运营成本，找到帕累托最优解区间。

最优 k ∈ [7, 9]



敏感性检验

到达率增 10% 至 33 人/分钟，最优解可能从 k=7 变为 k=8；服务率提升 20% 则可能允许 k=6。建议弹性调度：高峰 7 窗口、平峰 5 窗口。模型参数微小变化对决策的影响分析，体现鲁棒性设计思想。

弹性调度 · 动态响应

CLOSING

数学不是抽象的符号，而是理解现实的语言

从校园问题出发，用数学模型思考 · 课件与代码将发至社团群

THANK YOU