



中国研究生创新实践系列大赛
“华为杯”第二十二届中国研究生
数学建模竞赛

学 校

学校名称填写

参赛队号

00000000001

队员姓名

1.

成员 A

2.

成员 B

3.

成员 C

中国研究生创新实践系列大赛
“华为杯”第二十二届中国研究生
数学建模竞赛

题 目： 全国研究生数学建模竞赛论文标题

摘 要：

本模板是为全国研究生数学建模竞赛编写的 \LaTeX 模板, 旨在让大家专注于论文的内容写作, 而不用花费过多精力在格式的定制和调整上. 本手册是相应的参考, 其中提供了一些环境和命令可以让模板的使用更为方便. 同时需要注意, 使用者需要有一定的 \LaTeX 的使用经验, 至少要会使用 `ctex` 宏包的一些功能, 比如调节字距或修改字体大小等等.

2025 年格式变化说明

今年的格式变化如下：

1. 论文前两页 Logo 和 Title 替换；

这是研究生报名官方网站，点击[这里](#)进入。

关键词： 食堂打饭 明德中学 非线性优化模型 多属性决策

目录

1	问题重述	3
1.1	引言	3
1.2	问题的提出	3
2	模型的假设	4
3	符号说明	4
4	模型的建立	5
4.1	问题一分析	5
4.2	算法示例	5
5	表格和图形	6
5.1	表格	6
5.2	图形	8
6	模型评价	10
	参考文献	10
	附录 A MATLAB 源程序	11
A.1	第 1 问程序	11
	附录 B Python 源程序	12
B.1	第 2 问程序	12

1 问题重述

1.1 引言

午休铃刚响，明德中学的食堂就炸开了锅。初三（2）班的林晓宇抱着饭盒冲进人群，看着每条队伍都像蜿蜒的长蛇，不禁叹气：“每天都要花 20 分钟排队，作业都要写不完了！”同桌陈默推了推眼镜：“说不定，我们能用数学建模解决这个问题？”

1.2 问题的提出

明德中学食堂每日午休时段（12:00-12:30）、晚餐时段（18:00-18:30）人流高度集中，6 个打饭窗口前均出现长队。经初步统计，学生平均排队等待时间达 20 分钟，部分学生因排队错过午休，或因担心迟到放弃堂食。

问题一：学生选择队伍的盲目性（仅依据队伍长度，忽略窗口打饭速度差异）。在明德中学食堂高峰时段，学生选择打饭队伍时，普遍存在“以长度论快慢”的盲目决策现象。比如午休 12:05 左右，食堂 6 个窗口中， W_5 （汤类窗口）前排队人数为 8 人， W_1 （主食窗口）前排队人数为 10 人，多数学生会优先选择 W_5 队伍——仅因直观上“少 2 人”，却忽略了两个窗口的打饭速度差异： W_1 平均 30 秒就能为 1 名学生打饭（主食只需快速舀取米饭/馒头），而 W_5 因需逐个盛汤、加配料，平均 1 分 30 秒才能服务 1 人。

问题二：窗口功能分配不合理（慢窗口集中导致局部拥堵）食堂现有窗口布局为“横向连续排列”，且功能分配存在明显的“快慢扎堆”问题：6 个窗口从左到右依次为 W_1 （主食，快）、 W_2 （菜品，中慢）、 W_3 （菜品，中慢）、 W_4 （主食，快）、 W_5 （汤类，慢）、 W_6 （特色菜，慢）。其中， W_2, W_3 （中慢窗口）集中在中间区域， W_5, W_6 （慢窗口）集中在右侧区域，形成了两个明显的“拥堵带”。

问题三：缺乏对人流规律与等待时间的量化分析，无法针对性优化。食堂仅知道“午休、晚餐人多”，却不清楚具体高峰时段的人流峰值（如午休 12:08-12:15 为人流最高峰，到达率达 0.4 人/秒，远超其他时段的 0.2 人/秒）、不同年级学生的就餐偏好（初三学生因作业多，多在 12:00-12:10 集中就餐，初一学生则分散在 12:10-12:25），无法根据人流波动调整窗口开放数量（如高峰时开 6 个窗口，平峰时开 3 个）。

2 模型的假设

为简化问题、聚焦核心矛盾，建立以下假设：

1. 人流分布假设：学生到达食堂的时间服从泊松分布，即任意时间段内到达的学生数量相互独立，无集中“批量到达”（如整班集体前往）；
2. 窗口效率假设：同一窗口的打饭速度稳定（如主食窗口平均 30 秒 / 人，菜品窗口平均 1 分钟 / 人），无突发故障（如餐具短缺导致速度骤降）；
3. 学生选择假设：学生仅依据“当前队伍长度”选择窗口，不了解历史等待数据，且选择后不更换队伍；
4. 时间范围假设：仅研究人流高峰时段（午休 12:00-12:30、晚餐 18:00-18:30），非高峰时段（如 12:30 后）人流稀疏，无需优化。

3 符号说明

符号	意义
a	符号 1 的意义
b	符号 2 的意义
c	符号 3 的意义符号 3 的意义
d	符号 4 的意义
e	符号 5 的意义
f	符号 6 的意义符号 6 的意义
g	符号 7 的意义
h	符号 8 的意义
i	符号 9 的意义符号 9 的意义
k	符号 10 的意义
l	符号 11 的意义
m	符号 12 的意义
n	符号 13 的意义
p	符号 14 的意义
q	符号 15 的意义

4 模型的建立

4.1 问题一分析

题目要求建立模型描述折叠桌的动态变化图，由于在折叠时用力大小的不同，我们不能描述在某一时刻折叠桌的具体形态，但我们可以用每根木条的角度变化来描述折叠桌的动态变化。首先，我们知道折叠桌前后左右对称，我们可以运用几何知识求出四分之一木条的角度变化。最后，根据初始时刻和最终形态两种状态求出桌腿木条开槽的长度

4.2 算法示例

数学建模求解算法示例：

算法 1 算法的名字

输入: input parameters A, B, C

输出: output result

1: some description 算法介绍

2: **for** condition **do**

3: ...

4: **if** condition **then**

5: ...

6: **else**

7: ...

8: **while** condition **do**

9: ...

10: **return** result

多属性决策算法通过对每个决策选项（即各打饭窗口）的多个属性（队伍长度、服务效率）进行加权计算，得到每个选项的综合得分，得分最高的选项即为最优选择。在食堂排队场景中，需先定义属性权重（根据学生对“等待时间”的敏感度设定），再通过标准化处理消除属性单位差异，最终计算综合得分。

步骤 1：计算各窗口预估等待时间

预估等待时间 T_i 反映该窗口的实际等待成本，计算公式为：

$$T_i = \frac{L_i}{v_i}$$

示例： $T_1 = \frac{10}{2} = 5$ 分钟（主食窗口）， $T_5 = \frac{8}{0.67} \approx 12$ 分钟（汤类窗口）。

步骤 2：属性标准化处理

因“队伍长度”“服务效率”单位不同，需标准化为 $[0, 1]$ 区间的得分（得分越高，属性越

优): 1. 队伍长度标准化得分 (长度越短, 得分越高):

$$S_{L_i} = \frac{\max(L_i) - L_i}{\max(L_i) - \min(L_i)}$$

示例: 若当前窗口最大长度 $\max(L_i) = 10$, 最小 $\min(L_i) = 8$, 则 $S_{L_1} = \frac{10-10}{10-8} = 0$, $S_{L_5} = \frac{10-8}{10-8} = 1$ 。2. 服务效率标准化得分 (效率越高, 得分越高):

$$S_{v_i} = \frac{v_i}{\max(v_i)}$$

示例: $\max(v_i) = 2$ (主食窗口), 则 $S_{v_1} = \frac{2}{2} = 1$, $S_{v_5} = \frac{0.67}{2} \approx 0.33$ 。

步骤 3: 计算窗口综合得分

综合得分 S_i 为各属性标准化得分的加权和, 得分越高, 窗口越优:

$$S_i = w_1 \times S_{L_i} + w_2 \times S_{v_i}$$

示例: $S_1 = 0.4 \times 0 + 0.6 \times 1 = 0.6$, $S_5 = 0.4 \times 1 + 0.6 \times 0.33 \approx 0.598$ 。

5 表格和图形

5.1 表格

使用 `tabular` 环境, 如下表格: 表 5.1. 通过 `autoref` 引用表格: Table 5.1.

这里可使用命令 `\tabcolsep` 和 `\arraystretch` 分别控制列间距和行间距。

表 5.1 学术活动安排样例

日期	地点	活动名称	备注
2024 年 8 月 1 日	上海	学术研讨会	主题: 人工智能
2024 年 8 月 15 日	北京	学术交流会	重点: 数学建模
2024 年 9 月 1 日	深圳	研究研讨会	主题: 数据科学
2024 年 10 月 15 日	广州	创新论坛	重点: 科技创新

三线表

表 5.2 某校学生升高体重样本

序号	年龄	身高	体重
001	15	156	42
002	16	158	45
003	14	162	48
004	15	163	50
平均	15	159.75	46.25

研究生数学建模 2019 年 F 题结果示例

表 5.3 问题 1 结果 1 (左) 与问题 2 结果 (右)

数据集 1	数据集 1	数据集 2	数据集 1	数据集 1	数据集 2
A 问题 1	A 问题 1	A 问题 1	A 问题 2	A 问题 2	A 问题 2
503	503	163	503	503	163
294	200	114	294	200	114
91	80	8	91	80	8
607	237	309	607	237	309
540	170	305	540	170	305
250	278	123	250	278	123
340	369	45	340	369	45
277	214	160	277	214	160
B	397	92	B	397	92
	B	93		B	93
		61			61
		292			292
		B			B
104861	103518	109342	104917	103563	109427

表 5.4 问题 3 结果

数据集 1	数据集 1	数据集 2 (无问题点)		数据集 2 (有问题点)	
A 问题 3	A 问题 3	A 问题 3		A 问题 3	
503	503	169	73	169	73
69	69	322	249	322	249
506	506	270	274	270	274
371	371	89	12	89	12
183	183	236	216	236	216
194	194	132	16	132	16
450	450	53	282	53	282
286	113	112	84	112	141
485	485	268	287	268	291
B (9D)	248	250	99	250	161
	B (10D)	243	B (21D)	243	B (21D)
104861m	103518m		168924m		161650m

5.2 图形

图形并列

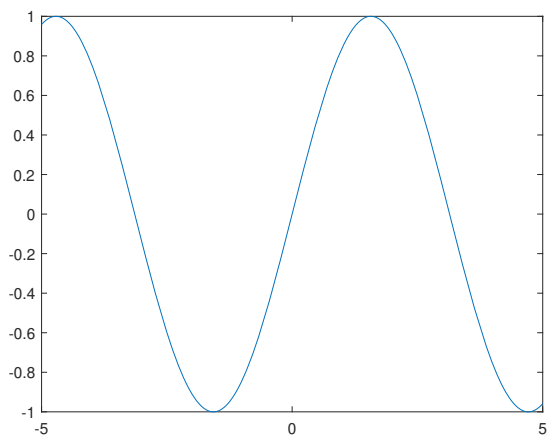


图 5.1 fig1

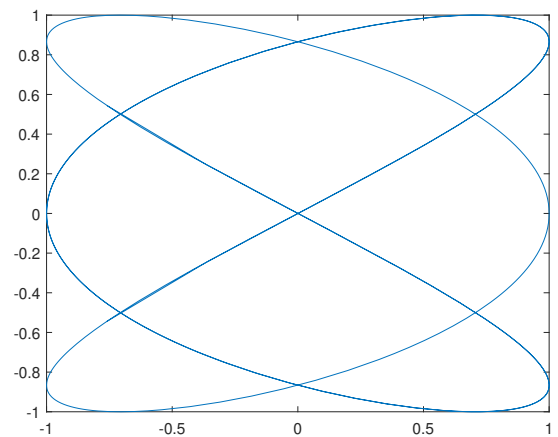


图 5.2 fig2

这是一个算法流程图

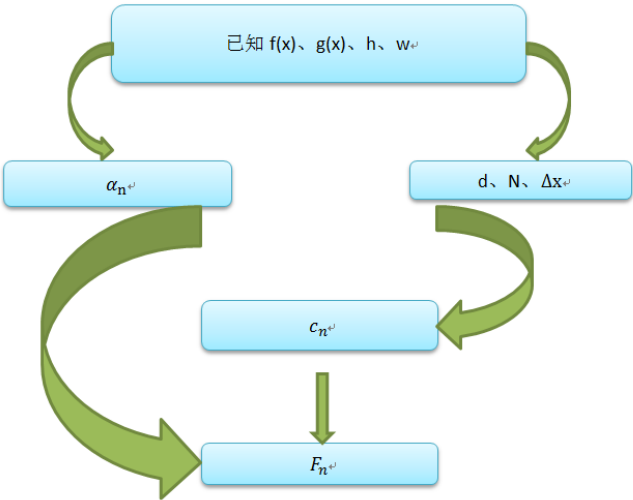


图 5.3 算法流程图

多图并排

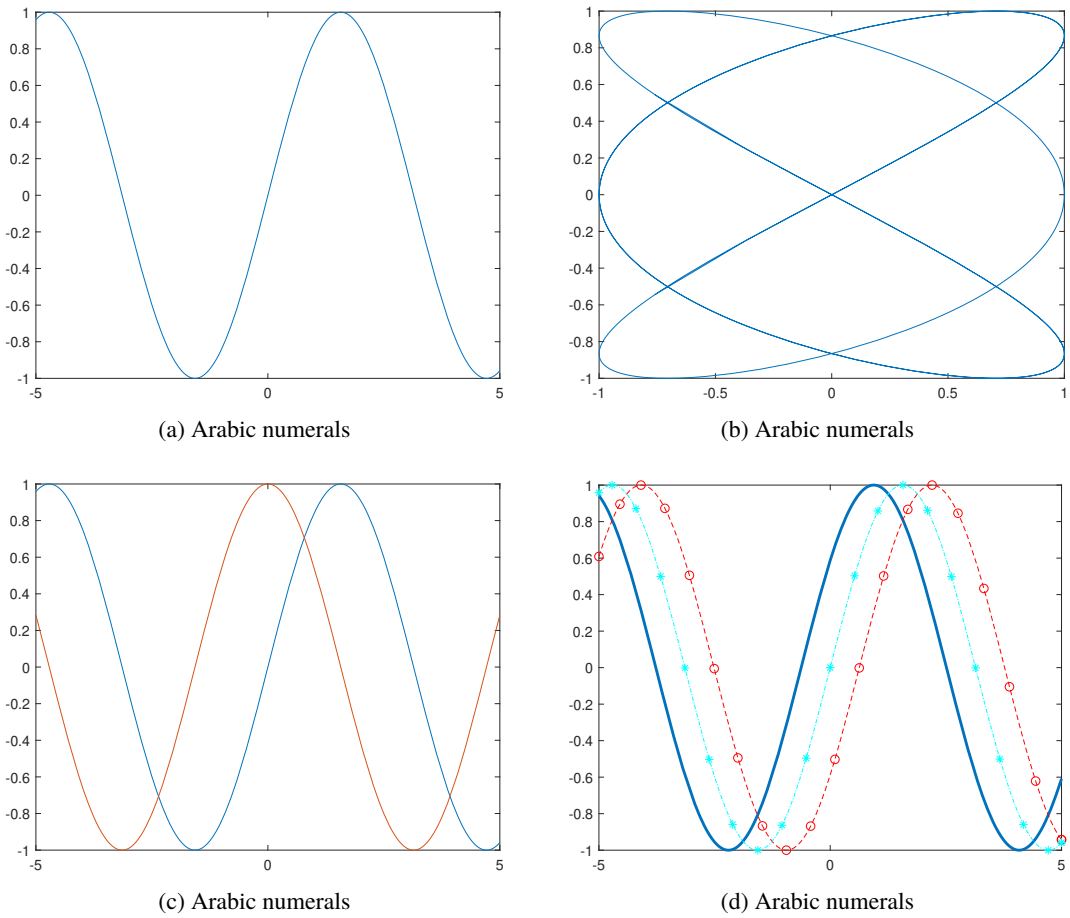


图 5.4 多图示例

6 模型评价

这里是模型评价

[1, 2, 3, 4]

参考文献

- [1] Mittelbach F, Goossens M, Braams J, et al., The \LaTeX Companion, 2nd ed., Reading, MA, USA: Addison-Wesley, 107-109, 2004.
- [2] Wright J, \LaTeX 3 programming: External perspective, TUGboat, 30(1):107-109, 2009.
- [3] Beeton B, Freytag A, Sargent III M, Unicode support for mathematics, <http://www.unicode.org/reports/tr25/>, 2018-07-21.
- [4] Vieth U, Experiences typesetting mathematical physics, Proceedings of EuroTeX, 13, 2009.

附录 A MATLAB 源程序

A.1 第 1 问程序

code.m

```
clear all
kk=2;
[mdd, ndd]=size(dd);
while ~isempty(V)
    [tmpd, j]=min(W(i, V));
    tmpj=V(j);
    for k=2:ndd
        [tmp1, jj]=min(dd(1, k)+W(dd(2, k), V));
        tmp2=V(jj);
        tt(k-1, :)= [tmp1, tmp2, jj];
    end
    tmp=[tmpd, tmpj, j; tt];
    [tmp3, tmp4]=min(tmp(:, 1));
    if tmp3==tmpd,
        ss(1:2, kk)= [i; tmp(tmp4, 2)];
    else
        tmp5=find(ss(:, tmp4)~=0);
        tmp6=length(tmp5);
        if dd(2, tmp4)==ss(tmp6, tmp4)
            ss(1:tmp6+1, kk)= [ss(tmp5, tmp4); tmp(tmp4, 2)];
        else, ss(1:3, kk)= [i; dd(2, tmp4); tmp(tmp4, 2)];
        end
    end
    end
    dd=[dd, [tmp3; tmp(tmp4, 2)]];
    V(tmp(tmp4, 3))=[];
    [mdd, ndd]=size(dd); kk=kk+1;
end;
S=ss; D=dd(1, :);
```

附录 B Python 源程序

B.1 第 2 问程序

mip1.py

```
# This example formulates and solves the following simple MIP
model:
# maximize
#      x +   y + 2 z
# subject to
#      x + 2 y + 3 z <= 4
#      x +   y       >= 1
#      x, y, z binary

# import gurobipy as gp
from gurobipy import * #GRB
try:
    # Create a new model
    m = Model("mip1")
    # Create variables
    x = m.addVar(vtype=GRB.BINARY, name="x")
    y = m.addVar(vtype=GRB.BINARY, name="y")
    z = m.addVar(vtype=GRB.BINARY, name="z")
    # Set objective
    m.setObjective(x + y + 2 * z, GRB.MAXIMIZE)
    # Add constraint: x + 2 y + 3 z <= 4
    m.addConstr(x + 2 * y + 3 * z <= 4, "c0")
    # Add constraint: x + y >= 1
    m.addConstr(x + y >= 1, "c1")
    # Optimize model
    m.optimize()
    for v in m.getVars():
        print('%s %g' % (v.varName, v.x))
    print('Obj: %g' % m.objVal)

except GurobiError as e:
    print('Error code ' + str(e.errno) + ': ' + str(e))

except AttributeError:
    print('Encountered an attribute error')
```