

数学建模

论文写作基本流程

题目
摘要（含关键词）
问题重述
问题假设
问题分析
符号说明
模型建立
模型求解
模型优缺点评价
参考文献

六大步骤

- 1.模型准备
- 2.模型假设
- 3.模型建立
- 4.模型求解
- 5.模型分析
- 6.模型检验

赛题类型

- 1.预测类
- 2.评价类
- 3.机理分析类赛题
- 4.优化类赛题

模型

- 1.线性规划（LP）模型
 - 涉及内容
 - 典型赛题
 - 实现及讲解
 - 一些情况下的代码如何写
 - 注意事项
 - 代码示例
- 2.非线性规划
 - 典型赛题
 - 实现及讲解
 - 代码示例
- 3.最短路径
 - 代码实现及讲解
 - 导出图形及显示路径

数学建模

论文写作基本流程

题目

模版：基于 模型的 研究与分析；（简短，精炼）

摘要（含关键词）

1. 评审获奖的关键
2. 不超过一页的四分之三
3. 看优秀论文（只看摘要和问题分析和建模过程）

问题重述

用你的理解将问题转换成数学语言

问题假设

对模型或全文做出合理的假设，可最大程度提高模型的准确率

问题分析

（用哪些模型，具体思路）

思路：

1. 分析问题
 2. 拆解问题
 3. 匹配模型
- 最好做一个流程图

符号说明

对论文中频繁出现或与寻常符号有不同意义的符号写在符号说明里

模型建立

通过数学符号和数学公式建立能够实现某个功能的数学模型

模型求解

考察编程能力

模型优缺点评价

参考文献

六大步骤

1. 模型准备

了解问题的实际背景，以数学思路贯穿问题的全过程，进而用数学语言来描述问题

2.模型假设

根据对象的特征和建模的目的对问题进行**简化**，提出一些恰当的**假设**

3.模型建立

在假设的基础上，规划各常量变量之间的关系，即建立数学结构

4.模型求解

利用获取的数据资料，对模型的所有参数做出计算（或近似计算）运用数学软件，通过限制条件对模型进行求解

5.模型分析

对建立模型的思路进行分析，对结果进行误差分析，数据稳定性分析等

6.模型检验

将结果与实际情形作比较

- （1）较吻合，对计算结果给出实际含义并解释
- （2）吻合性差，则修改假设，再次重复建模过程

赛题类型

1.预测类

2.评价类

3.机理分析类赛题

一般在国赛和美赛中有

4.优化类赛题

模型

1.线性规划（LP）模型

（在一组**线性**¹约束条件限制下，求**线性目标函数**最大或最小值）

涉及内容

- 1.决策变量：参数
- 2.目标函数
- 3.约束条件

典型赛题

- 题目中提到“怎样安排/分配，尽量（可能）多（少），最多（少），利润最大，最合理”等
- 生产安排
- 投资收益
- 销售运输
- 车辆安排

(整数规划，0-1规划默认为线性规划的特例)

实现及讲解

$$\begin{aligned} \max \quad & y = 40x_1 + 30x_2 \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \leq 6 \\ x_1 \geq 1, x_2 \geq 1 \\ 240x_1 + 120x_2 \leq 1200 \end{cases} \end{aligned}$$

$$\begin{aligned} \min \quad & w = \begin{bmatrix} -40 \\ -30 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{s.t.} \quad & \begin{cases} \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ 240 & 120 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 6 \\ -1 \\ -1 \\ 1200 \end{bmatrix} \\ [1,1] \cdot [x_1, x_2]^T \geq [0,0] \end{cases} \end{aligned}$$

- `[x, fval] = linprog2(f,A, b, Aeq, beq,lb,ub)`
 - x返回最优解的变量取值，fval返回目标函数的最优值

| | |
|---------|---------------------------|
| f | 目标函数的系数列向量 |
| A,b | 不等式约束条件的变量系数矩阵和常数项矩阵A.b |
| Aeq,beq | 等式约束条件的系数矩阵和常数项矩阵Aeq, beq |
| lb,ub | 决策变量的最小取值和最大取值 |

注：s.t.在论文中代表约束条件

一些情况下的代码如何写

- 若不存在不等式约束，用“[]”代替A和b: $[x, fval] = \text{linprog}(f, [], [], Aeq, beq, lb, ub)$
- 若不存在等式约束，用“[]”代替Aeq和beq: $[x, val] = \text{linprog}(f, A, b, [], [], lb, ub)$
- 没有等式约束和最小、最大取值的约束时，可以不写Aeq, beq 和lb, ub: $[x, val] = \text{linprog}(f, A, b)$

注意事项

matlab标准型中：目标函数只能求**最小值**，**约束条件只有“ ≤ 0 ”或“ $=$ ”**；故：

- 若约束条件中有“ \geq ”，则在两边添加负号变成“ \leq ”
- 若题目求最大值：目标函数等号两端加负号转为求最小值；求解后目标值再取负

代码示例

$$\begin{aligned} \max \quad & y = 40x_1 + 30x_2 \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \leq 6 \\ x_1 \geq 1, x_2 \geq 1 \\ 240x_1 + 120x_2 \leq 1200 \end{cases} \end{aligned}$$

$$\begin{aligned} \min \quad & w = \begin{bmatrix} -40 \\ -30 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{s.t.} \quad & \begin{cases} \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ 240 & 120 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 6 \\ -1 \\ -1 \\ 1200 \end{bmatrix} \\ [1, 1] \cdot [x_1, x_2]^T \geq [0, 0] \end{cases} \end{aligned}$$

```
f = [-40; -30]; %目标函数中变量的系数矩阵
a = [1, 1; -1, 0; 0, -1; 240, 120]; %小于等于的约束条
b = [6; -1; -1; 1200]; %小于等于的约束条
```

```
%标准形式: [x, fval] = linprog(f, A, b, aeq, beq, lb,
%从某个变量开始后面都没有？全都不写
%没有等式约束条件、但有上下界约束时，用 [] 在函数里占
[x, y] = linprog(f, a, b)
y = -y
```

```
% zeros(m,n)是元素全为0的m行n列的矩阵
```

Optimal solution found.

```
x = 2x1
    1
    4.0000
    2.0000
```

y = -220

y = 220

2.非线性规划

内容与线性规划一样，除了变量，**至少有一个变量不是一次方**

典型赛题

- 题目中提到“怎样安排/分配，尽量（可能）多（少），最多（少），利润最大，最合理”等，**但至少有一个变量不是一次方**
- 投资规划：收益率，最佳投资方案（也是求收益率）；
- 角度调整：eg:飞机管理避免相撞，影院最佳视角等与角度有关的，涉及三角函数的
- 生产安排：目标函数或者约束条件有非线性变量

实现及讲解

eg:

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 + x_3^2 + 8 \\ \begin{cases} x_1^2 - x_2 + x_3^2 \geq 0 \\ x_1 + x_2^2 + x_3^3 \leq 20 \\ -x_1 - x_2^2 + 2 = 0 \\ x_2 + 2x_3^2 = 3 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$

- `[x,fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)`
- `x`得到决策变量取值, `fval`得到最优解取值
- 本题:设好后`fun`和`nonlcon`函数文件以及其他参数后, 执行:
- `[x,y] = fmincon3 ('fun1',[0;0;0],[],[],[],[],[0;0;0],[],'fun2')`
 - 约束条件中无线性约束, 故`A,b,Aeq,beq`用`[]`代替

| | |
|-----------------------|----------------------------------|
| <code>fun</code> | 单独脚本文件里定义的目标函数 |
| <code>x0</code> | 决策变量的初始值。不知道的话随便写个数 |
| <code>A, b</code> | 线性约束的不等式变量系数矩阵和常数项矩阵(都是 \leq) |
| <code>Aeq, beq</code> | 线性约束的等式变量系数矩阵和常数项矩阵 |
| <code>lb, ub</code> | 决策变量的最小取值和最大取值 |
| <code>nonlcon</code> | 非线性约束, 包括不等式和等式 |

代码示例

```
function f = fun1(x)
    f = sum(x.^2)+8;
```

```
function [g,h]=fun2(x)
    g=[-x(1)^2+x(2)-x(3)^2          % g是非线性不等式
        x(1)+x(2)^2+x(3)^3-20]
    h=[-x(1)-x(2)^2+2                % h是线性不等式
        x(2)+2*x(3)^2-3]
```

```
[x,y] = fmincon('fun1',[1000;8;500],[],[],[],[],[0;0;0],[],'fun2')
```

```
x
y
[0.5522;1.2033;0.9...
10.6511]
```

3.最短路径

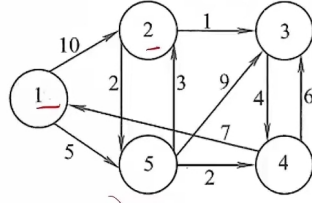
从图中的某个顶点出发, 到达另外一个顶点的所经过的边的权重之和最小的一条路径

- 图:边和节点组成的结构, 在数学建模中例如题中道路和城市。

- 边带有方向的是有向图，否则为无向图
- 权重:每条边都有与之对应的值，eg:题中边道路，边的权重就是道路长度，当然是越小越好.

代码实现及讲解

- 在下图中，用点表示城市，现有 v_1, v_2, \dots, v_5 共5个城市。点与点之间的连线表示城市间有道路相连。连线旁的数字表示道路的长度。现计划从城市 v_1 到城市 v_3 铺设一条天然气管道，请设计出最小长度管道铺设方案。



```

1 % sparse生成稀疏矩阵，也就是除了注明的几个元素外，其余都是0
2 % sparse里第一个和第二个矩阵相同位置的元素值就是非零元素的索引
3 % 非零元素的值
4 % w是每条边的权重
5 w = [10,5,2,1,4,6,7,3,9,2]
6 DG = sparse([1,1,2,2,3,4,4,5,5,5], ...
7             [2,5,5,3,4,3,1,2,3,4],w)
8 % dist是最短路径的值。
9 % path是最短路径的节点顺序
10 % pred是到每一个节点的最短路径的终点前一个节点
11 [dist,path,pred] = graphshortestpath(DG,1,3)

```

DG:

$$\begin{bmatrix} 0 & 10 & 0 & 0 & 5 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 4 & 0 \\ 7 & 0 & 6 & 0 & 0 \\ 0 & 3 & 9 & 2 & 0 \end{bmatrix}$$

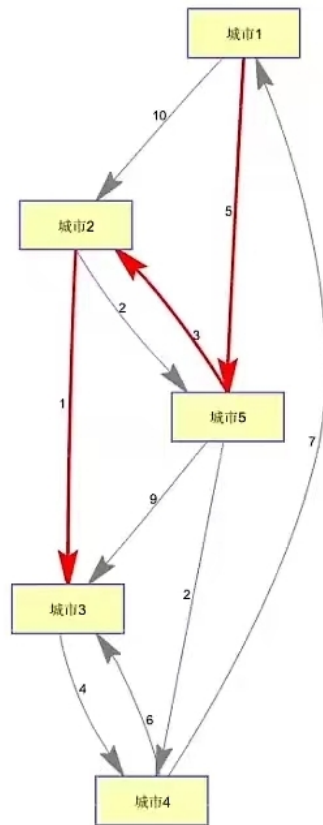
(1,2)=10; (1, 5) =5.....;

导出图形及显示路径

```

1 % 将最短路径的节点和边缘标记为红色并增加线宽
2 % getedgesbynodeid得到图h的指定边的句柄% 第一个参数是图，第二个是边的出点，第三个是边的入点
3 % 句柄确保能找到对应的东西% get查询图的属性，h.Nodes(path)，ID 得到图h中最短路径的% set函数设置图形属性
4 edges = getedgesbynodeid(h,get(h.Nodes(path), ID));
5 set(edges,'LineColor',[1 @])% RGB数值，红绿蓝
6 set(edges,'Linewidth',2)

```



1. 所有变量都是一次方 [↗](#)
2. `linprog`函数使用matlab标准型 [↗](#)
3. `fmincon`函数使用matlab标准型 [↗](#)