数模加油站

数学建模模型算法精讲课——

非线性规划

—— 江北老师

相信是成功的起点，

坚持是成功的终点！

非线性规划

- 模型引出
- 模型原理
- 典型例题
- 代码求解

非线性规划——模型引出

- **除了上节课讲到的线性规划模型，优化类问题往往还有一些更复杂的**
  - 常见收益率、传播率、经济增长率等规划问题，往往涉及到$\frac{1}{x}$的形式，属于非线性
  - 空间运动问题：比如空间约束、避免碰撞、角度调整等等

    因为空间运动往往是曲线或是角度的变换，常用到三角函数、指数函数等形式，属于非线性
  - 运输问题：比如已知坐标，要运送物品，会用到距离公式，是变量的指数形式，属于非线性
  - 示例：

$$\min \quad y = x_1^2 + x_2^2 - x_1 x_2 - 2x_1 - 5x_2,$$

$$\text{s.t.} \begin{cases} -(x_1 - 1)^2 + x_2 \geq 0, \\ 2x_1 - 3x_2 + 6 \geq 0 \end{cases}$$

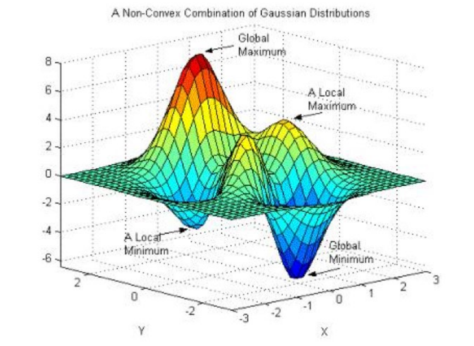
  - 这就是一个非线性规划模型，与线性规划类似，也是在确定决策变量后写出目标函数和约束条件。

非线性规划

非线性规划是一种求解目标函数或约束条件中有**一个或几个非线性函数**的最优化问题的方法。运筹学的一个重要分支。20世纪50年代初，库哈(H. W. Kuhn) 和托克 (A. W. Tucker) 提出了非线性规划的基本定理，为非线性规划奠定了理论基础。这一方法在工业、交通运输、经济管理和军事等方面有广泛的应用，特别是在“**最优设计**”方面，它提供了数学基础和计算方法，因此有重要的实用价值。

非线性规划模型特点

- 模型中至少一个变量是**非线性**，即包含x^2, e^x, $\frac{1}{x}$, $\sin x$, $\log_2 x$等形式
- 线性规划有通用求准确解的方法（单纯形法），它的最优解只存在于可行域的边界上; 非线性规划的最优解（若存在）可能在其可行域的任一点达到，目前非线性规划还**没有适合各种问题的一般解法**，各种方法都有其特定的适用范围
- 针对数学建模来说，掌握matlab的函数求**近似解**即可



非线性规划——模型原理

MATH 数模加油站

➢ 非线性规划模型的标准型

$$\min \quad f(x)$$

$$\text{s.t.}\begin{cases} Ax \le b,\ \ Aeq \cdot x = beq & \text{（线性）} \\ c(x) \le 0,\ \ Ceq(x) = 0 & \text{（非线性）} \\ lb \le x \le ub \end{cases}$$

➢ 非线性规划求解的Matlab函数

- fmincon函数：$[x, fval] = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, option)$

✓ fun ：把目标函数定义为一个单独的函数文件（min）

✓ $x0$ ：决策变量的初始值

✓ A, b ：**线性约束的不等式**变量系数矩阵和常数项矩阵（⩽或＜）

✓ Aeq, beq ：**线性约束的等式**变量系数矩阵和常数项矩阵

✓ lb, ub ：决策变量的最小取值和最大取值

✓ $nonlcon$ ：**非线性**约束，包括**不等式和等式**

✓ $option$ ：求解非线性规划使用的方法

➢ **fmincon函数：** $[x, fval] = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, option)$

- 非线性规划中对于**初始值x0的选取非常重要**，因为非线性规划的算法求解出来的是一个局部最优解。如果要求全局最优解，有两个思路:

  ✓ 给定不同的初始值，在里面找到一个最优解;

  ✓ 先用**蒙特卡罗模拟**，得到一个蒙特卡罗解，然后将这个解作为初始值来求最优解

- option选项可以给定求解的算法，一共有五种，interior-point（内点法）、trust-region-reflective（信赖域反射法）、sqp（序列二次规划法）、sqp-legacy（约束非线性优化算法）、active-set（有效集法）。不同的算法有其各自的优缺点和适用情况，我们可以改变求解的算法来对比求解的结果。

- fun表示目标函数，我们要编写一个独立的“m文件”储存目标函数

$$
\begin{aligned}
&function \quad f = fun(x) \\
&\qquad f = \ldots\ldots \\
&end
\end{aligned}
$$

- **fmincon函数：** $[x,fval]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,option)$
  - *nonlcon* 表示**非线性部分的约束**，也要编写一个独立的“m文件”储存非线性约束条件

    $function\ \ [c,ceq]=nonlfun(x)$

    c=[非线性不等式约束1;

    ……

    非线性不等式约束2]

    ceq=[非线性等式约束1;

    ……

    非线性等式约束2]

    end
  - 决策变量**下标要改为括号**，比如x_1要改为$x(1)$，matlab才能识别
  - 若不存在某种约束，可以用“[]”替代，若后面全为“[]”且option使用默认，后面的“[]”可省略

➢ 选址问题

- 临时料场：$A(5,1)$，$B(2,7)$；日储量各20吨

工地位置坐标及日需求量						
	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ
横坐标	1.25	8.75	0.5	5.75	3	7.25
纵坐标	1.25	0.75	4.75	5	6.5	7.25
日需求量	3	5	4	7	6	11

- （1）试制定每天的供应计划，即从两料场分别向各工地运送多少吨水泥，使总的吨千米数最小？
- （2）为了进一步减少吨千米数，打算舍弃两个临时料场，改建两个新的，日储量各为20吨，问应建在何处，节省的吨千米数为多大？

MATH 数模加油站

➢ 选址问题

- 1）确定决策变量
  - ✓ 设第i个工地的坐标(a_j, b_j)，水泥日用量d_j，$i = 1, 2, \ldots, 6$；料场位置(x_j, y_j)，日储量e_j，$j = 1, 2$；从料场j向工地i的运送量为x_{ij}
- 2）确定约束条件
  - ✓ 料场水泥运输总量**不超过其日储量**：$\sum_{i=1}^{6} x_{ij} \le e_j, j = 1, 2$
  - ✓ 两个料场向某工地运输量之和**等于该工地水泥日用量**：$\sum_{j=1}^{2} x_{ij} = d_i, i = 1, 2, \ldots, 6$
- 3）确定目标函数
  - ✓ 求总吨千米数最小，即运送量乘运送距离求和最小$min\ \ f = \sum_{j=1}^{2}\sum_{i=1}^{6} x_{ij}\sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}$
- 4）建立模型

$$min \quad f = \sum_{j=1}^{2}\sum_{i=1}^{6} x_{ij}\sqrt{(x_j + a_i)^2 + (y_j - b_i)^2}$$

$$s.t.\begin{cases} \sum_{i=1}^{6} x_{ij} \le e_j, j = 1, 2 \\ \sum_{j=1}^{2} x_{ij} = d_i, i = 1, 2, \ldots, 6 \\ x_{ij} \ge 0, i = 1, 2, \ldots, 6; j = 1, 2 \end{cases}$$

数模加油站

➢ 选址问题

- 5）求解

✓ 对于第一问：因料场位置已知，故决策变量仅为x_{ij}，为**线性规划模型**

✓ 对于第二问：新料场位置位置，所以x_i 和x_j均为变量，且不是线性的，故为**非线性规划模型**

✓ 共有8个约束

$$min\ f=\sum_{j=1}^{2}\sum_{i=1}^{6}x_{ij}\sqrt{\left(x_j+a_i\right)^2+\left(y_j-b_i\right)^2}$$

$$s.t.\begin{cases}\sum_{i=1}^{6}x_{ij}\le e_j, j=1,2 \quad \left(x_{11}+x_{21}+\ldots+x_{61}\le e_1, x_{12}+x_{22}+\ldots+x_{62}\le e_2\right)\\ \sum_{j=1}^{2}x_{ij}=d_i, i=1,2,\ldots,6\ \left(x_{11}+x_{12}=d_1,\ldots,\ x_{61}+x_{62}=d_6\right)\\ x_{ij}\ge 0, i=1,2,\ldots,6; j=1,2\end{cases}$$

✓ 注意，在matlab里这些双角标的变量要**改为单角标的变量**，如$x_{11}\to x_1$， $x_{21}\to x_2$， … ， $x_{62}\to x_{12}$

非线性规划——代码求解

➢ 示例代码

$$\min \quad y = x_1^2 + x_2^2 - x_1x_2 - 2x_1 - 5x_2,$$

$$\text{s.t.}\begin{cases} -(x_1-1)^2 + x_2 \ge 0, \\ 2x_1 - 3x_2 + 6 \ge 0 \end{cases}$$

```
clear;clc
format long g %可以将Matlab的计算结果显示为一般的长数字格式（默认会保留四位小数，或使用科学计数法）
%% 例题1的求解
% max f(x) = x1^2 +x2^2 -x1*x2 -2x1 -5x2
% s.t. -(x1-1)^2 +x2 >= 0 ; 2x1-3x2+6 >= 0
x0 = [0 0]; %任意给定一个初始值
A = [-2 3]; b = 6;
[x,fval] = fmincon(@fun1,x0,A,b,[],[],[],[],@nonlfun1) % 注意 fun1.m文件和nonlfun1.m文件都必须在当前文件夹目录下
fval = -fval
```

➢ **示例代码**

```
%% 使用其他算法对例题求解
% Matlab默认使用的算法是'interior-point' 内点法
% 使用interior point算法 （内点法）
option = optimoptions('fmincon','Algorithm','interior-point')
[x,fval] = fmincon(@fun1,x0,A,b,[],[],[],[],@nonlfun1,option)
fval = -fval
% 使用SQP算法 （序列二次规划法）
option = optimoptions('fmincon','Algorithm','sqp')
[x,fval] = fmincon(@fun1,x0,A,b,[],[],[],[],@nonlfun1,option)
fval = -fval %得到-4.358，远远大于内点法得到的-1,猜想是初始值的影响
% 改变初始值试试
x0 = [1 1]; %任意给定一个初始值
[x,fval] = fmincon(@fun1,x0,A,b,[],[],[],[],@nonlfun1,option) % 最小值为-1，和内点法相同（这说明内点法的适应性要好）
fval = -fval
```

MATH 数模加油站

➢ 示例代码

```
%% 使用蒙特卡罗的方法来找初始值(推荐)
clc,clear;
n=10000000;  %生成的随机数组数
x1=unifrnd(-100,100,n,1);  % 生成在[-100,100]之间均匀分布的随机数组成的n行1列的向量构成x1
x2=unifrnd(-100,100,n,1);  % 生成在[-100,100]之间均匀分布的随机数组成的n行1列的向量构成x2
fmin=+inf;  % 初始化函数f的最小值为正无穷（后续只要找到一个比它小的我们就对其更新）
for  i=1:n
x = [x1(i), x2(i)];  %构造x向量, 这里千万别写成了: x =[x1, x2]
if  ((x(1)-1)^2-x(2)<=0)  &  (-2*x1(i)+3*x2(i)-6  <=  0)  % 判断是否满足条件
result = -x(1)^2-x(2)^2 +x(1)*x(2)+2*x(1)+5*x(2) ;  % 如果满足条件就计算函数值
if result < fmin  % 如果这个函数值小于我们之前计算出来的最小值
fmin = result;  % 那么就更新这个函数值为新的最小值
x0 = x;  % 并且将此时的x1 x2更新为初始值
end
end
end
disp('蒙特卡罗选取的初始值为: '); disp(x0)
A = [-2 3]; b = 6;
[x,fval] = fmincon(@fun1,x0,A,b,[],[],[],[],@nonlfun1)
fval = -fval
```

MATH 数模加油站

➢ 示例代码

```
function f = fun1(x)
% 注意：这里的f实际上就是目标函数，函数的返回值也是f
% 输入值x实际上就是决策变量，由x1和x2组成的向量
% fun1是函数名称，到时候会被fmincon函数调用，可以任意取名
% 保存的m文件和函数名称得一致，也要为fun1.m
% max f(x) = x1^2 +x2^2 -x1*x2 -2x1 -5x2
f = -x(1)^2-x(2)^2 +x(1)*x(2)+2*x(1)+5*x(2) ;
end
function [c,ceq] = nonlfun1(x)
% 注意：这里的c实际上就是非线性不等式约束，ceq实际上就是非线性等式约束
% 输入值x实际上就是决策变量，由x1和x2组成的一个向量
% 返回值有两个，一个是非线性不等式约束c，一个是非线性等式约束ceq
% nonlfun1是函数名称，到时候会被fmincon函数调用，可以任意取名，但不能和目标函数fun1重名
% 保存的m文件和函数名称得一致，也要为nonlfun1.m
% -(x1-1)^2 +x2 >= 0
c = [(x(1)-1)^2-x(2)]; % 千万别写成了：(x1-1)^2 -x2
ceq = []; % 不存在非线性等式约束，所以用[]表示
end
```

例题代码

代码较复杂，可直接看代码文件xuanzhiwenti.m。

欢迎关注数模加油站

THANKS