

HW03-nonlinear programming

问题1

分析

- 决策变量

两种商品的促销水平: $x_i, i = 1, 2$

- 目标函数

- 总销售额:

$$\max z = \sum_{i=1}^2 f(x_i) = 3x_1 - (x_1 - 1)^2 + 3x_2 - (x_2 - 2)^2$$

- 约束条件

- $4x_1 + x_2 \leq 20$
- $x_1 + 4x_2 \leq 20$
- $x_i \geq 0, i = 1, 2$

代码

```
import numpy as np
from scipy.optimize import minimize

# 定义目标函数
def objective_function():
    def obj(x):
        # 目标函数:  $-(3*x[0] - (x[0]-1)**2 + 3*x[1] - (x[1]-2)**2)$ 
        return  $-(3*x[0] - (x[0]-1)**2 + 3*x[1] - (x[1]-2)**2)$ 
    return obj

# 定义第一个约束条件函数
def constraint1(x):
    # 约束条件1:  $4 * x[0] + x[1] - 20 \geq 0$ 
    return  $4 * x[0] + x[1] - 20$ 

# 定义第二个约束条件函数
def constraint2(x):
    # 约束条件2:  $4 * x[1] + x[0] - 20 \geq 0$ 
    return  $4 * x[1] + x[0] - 20$ 

# 定义约束条件集合
def constraints():
    cons = [{'type': 'ineq', 'fun': constraint1}, # 约束条件1
            {'type': 'ineq', 'fun': constraint2}] # 约束条件2
    return cons

if __name__ == "__main__":
    ## 初始迭代点
    x0 = np.array([1, 1]) # 初始猜测的解

    ## 获取约束条件
    cons = constraints()

    ## 定义变量的边界
    bnds = ((0, None), (0, None)) # 每个变量的边界 (0, 无上限)

    ## 使用 minimize 函数求解优化问题
    res = minimize(objective_function(), x0, bounds=bnds, constraints=cons)

    ## 打印优化结果
    print(res)
```

结果

x_1	x_2	销售额
4.0	4.0	10.99

问题2

分析

- 决策变量
第*i*轮生产时，投入生产A的量 y_i
- 目标函数
记第*i*轮生产的利润为 z_i
经过*n*轮生产后的总利润，
$$\max \quad z = \sum_{i=1}^n z_i = \sum_{i=1}^n [g(y_i) + h(x_i - y_i)]$$
- 约束条件
 - $x_i = ay_{i-1} + b(x_{i-1} - y_{i-1})$
 - $0 \leq y_i \leq ay_{i-1} + b(x_{i-1} - y_{i-1}), i = 2, 3, \cdots, n$

模型

$$\max \quad z = \sum_{j=1}^n z_j, \quad s.t. \quad 0 \leq y_j \leq x_j, \quad x_j = ay_{j-1} + b(x_{j-1} - y_{j-1}), z_j = g(y_j) + h(x_j - y_j)$$