线性规划

一. 线性规划的概念和理论

1.线性规划问题的概述

> 线性规划(Linear Programming)是运筹学的一个重要分支，数学上它用来确定多变量线性函数在变量满足线性约束条件下的最优值。线性规划模型通常由三个要素--**决策变量**、**目标函数**和**约束条件**构成。

一般模型：

$$max(min)z = sum_{j=1}^{n} c_j x_j$$

$$s.t. sum_{j=1}^{n} a_{ij} x_j \leq (\geq, =) b_i, i = 1,2,\ldots,m \text{ 且 } x_j \geq 0, j = 1,2,\ldots,n$$

矩阵形式：

$$max(min)z = c^T x \text{ ,}$$

$$s.t. Ax \leq (\geq, =) b, \text{且} x \geq 0$$

向量形式：

$$max(min)z = c^T x,$$

$$s.t. sum_{j=1}^{n} p_j x_j \leq (\geq, =) b, \text{且} x \geq 0$$

数学上，规定线性规划的标准型为 $maxz = c^T x, s.t. Ax = b \text{且} x \geq 0$

2.线性规划解的概念和性质

- 可行域：满足全部约束条件的决策向量$x \in R^n$构成的集合，且为*凸集
- 最优解：使得目标函数得到最优值（并且有界）的可行解
- 定理： **当有最优解时，一定可以在可行域的某个顶点上取到。当有唯一解时，最优解就是可行域的某个顶点。当有无穷多个最优解时，其中至少有一个解是可行域的一个顶点。**

3.可转化为线性规划的问题

如问题 $min|x_1| + |x_2| + \ldots + |x_n|, s.t. Ax \leq b$

只要取$u_i = \frac{|x_i| + x_i}{2}, v_i = \frac{|x_i| - x_i}{2}$就可以满足线性规划的条件

问题自然转化为$minsum_{i=1}^{n}(u_i + v_i), s.t. [A, -A]\begin{bmatrix} u \\ v \end{bmatrix} \leq b \text{且} u, v \geq 0$

二. 线性规划的Python求解

1.用scipy.optimize模块求解

scipy中线性规划的标准型：$minz = c^T x, s.t. Ax \leq b, Aeq \cdot x = beq, Lb \leq x \leq Ub$

```
from scipy.optimize import linprog
res=linprog(c,A,b,Aeq,beq,bounds,method='***')
res.fun 目标函数最小值（最优解）
res.x 最优解的解向量
```

2. 用cvxopt.solvers模块求解

cvxopt.solvers中线性规划的标准型：$minz = c^T x, s.t. Ax \leq b, Aeq \cdot x = beq$

```
import numpy as np
from cvxopt import matrix,solvers
sol=solvers.lp(c,A,b,Aeq,beq)
print(sol['x'])
print(sol['primal objective'])
```

在程序中虽然没有直接使用NumPy库中的函数，也必须加载，数据必须写成浮点型数据

3. 用cvxpy模块求解

属于可以计算导入数据的线性规划最优解的方法

```
import cvxpy as cp
import numpy as np
import pandas as pd
obj=cp.Minimize(cp.sum(cp,multiply(c,x)))#ΣX(i,j) C*X
con=[cp.sum(x,axis=1,keepdims=True)<=e,
    cp.sum(x,axis=0,keepdims=True)==d,x>=0]#1是按行相加，0是按列相加
prob=cp.Problem(obj,con)#obj是目标函数，con是约束条件
prob.solve(solver='GLPK_MI',verbose=True)
print(prob.value)
print(x.value)
```