# 线性规划（Linear Programming）程序

17071107 李荣庆

2020/3/17

## 一、功能

本程序使用 M 法，根据用户的输入的目标函数以及约束条件，自动计算线性规划问题的最优解$X^*$以及最值。

## 二、程序输入

本程序需要输入目标函数(Objective Function)、约束的数量(k)，以及约束(Constraint)。其形式如下：

目标函数以$\max or \min\ (c_1, c_2, \dots, c_n)$的形式给出，资源向量$C \in \mathbb{R}^n$，例如：max (2,3,1) 代表$\max\ 2x_1 + 3x_2 + x3$。

约束以 $(a_{i1}, a_{i2}, \dots, a_{in}) \leq or \geq or = b_i$的形式给出其中约束向量$a_i \in \mathbb{R}^n$，$b_i \in \mathbb{R}^m$。最后写成增广矩阵的形式为：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix}$$

#输入实例

$$max\ (2,3)$$

3（表示有三个约束）

$$(1,2)\ <=\ 8$$

$$(4,0)\ <=\ 16$$

$$(0,4)\ <=\ 12$$

**其最终矩阵的形式为**

$$mat\_constraint = \begin{bmatrix} 1 & 2 & 8 \\ 4 & 0 & 6 \\ 0 & 4 & 12 \end{bmatrix}$$

# 三、化标准型

依照单纯形法的规则化标准型

1、若目标函数是 min，则将 z 取相反数，目标函数变为 max，此时最优解不变。

2、若$b_i < 0$则将该约束取相反数，相应的约束条件也要变号。≤ 变 ≥， ≥ 变 ≤

3、若约束条件为≤，则添加松弛变量；若约束条件为≥则减去松弛变量，再加上人工变量；若约束条件为=，则添加人工变量。

例如，

$$min\ (-3, 1, 1)$$

$$3$$

$$(1, -2, 1)\ <=\ 11$$

$$(-4, 1, 2)\ >=\ 3$$

$$(-2, 0, 1)\ =\ 1$$

化为标准型后，约束矩阵为：

$$mat\_constraint = \begin{bmatrix} 1 & -2 & 1 & 1 & 0 & 0 & 0 & 11 \\ -4 & 1 & 2 & 0 & -1 & 1 & 0 & 3 \\ -2 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

# 四、函数功能

## 1、lp_input

参数：无
功能：接受用户输入
输入：目标函数以及约束
返回值：标准化后的资源向量、约束矩阵、人工变量的位置、是否改变目标函数

## 2、normalization

参数：需要标准化的资源向量以及约束矩阵
功能：将资源向量及约束矩阵转化为标准型
返回值：标准化后的资源向量、约束矩阵、人工变量的位置、是否改变目标函数

## 3、simplex_method

参数：标准化后的资源向量、约束矩阵、人工变量的位置、是否改变目标函数
功能：通过 M 法，计算线性规划问题的解
返回值：最优解$X^*$，最优值，迭代次数，计算时间

### 4、find_base_index

参数：v_base，约束矩阵
功能：寻找当前约束矩阵中的基
返回值：基向量所在的列号
解释：假设 m=4，则总共需要调用四次 find_base_index，每一次传入的 v_base 分别是(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)

### 5、print_solution

参数：最佳解，最佳值，迭代次数，执行时间
功能：打印解有关的信息
返回值：无

### 6、solve_lp

参数：无
功能：通过用户控制台的输入求解线性规划问题
返回值：无

### 7、solve_lp_manual_input

参数：目标函数类型，资源向量，约束矩阵，约束条件
功能：通过用户在程序中手动设置输入，以求解线性规划问题
解释：函数类型为字符串类型，为'max'或'min'；约束条件为列表，其长度为约束的个数，如['<=', '>=', '=]
返回值：无

## 五、关于解的说明

该程序可正确求解存在最优解的线性规划问题，对于无界解，程序会输出"LP problem has UNBOUNDED SOLUTION!"，对于可行域为空的解，也就是不存在解，程序会输出"Artificial variables are in the base, No SOLUTION!"

## 六、测试

### 1、测试样例

#测试样例一（正确答案：14）

max (2, 3)
3
(1, 2) <= 8
(4, 0) <= 16
(0, 4) <= 12

#测试样例二（正确答案：-2）

min (-3, 1, 1)
3
(1, -2, 1) <= 11
(-4, 1, 2) >= 3
(-2, 0, 1) = 1

#测试样例三（无界解）

max (2, 2)
2
(1, -1) >= -1
(-0.5, 1) <= 2

#测试样例四（无解）

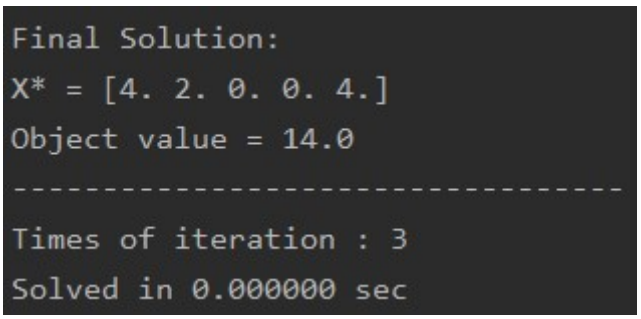
max (1, 1)
2
(1, -1 ) >= 0
(3, -1) <= -3

#测试样例五（正确答案：2.25）

min (1, 1.5)
2
(1, 3) >= 3
(1, 1) >= 2

## 2、测试截图

测试一：



测试二：

```
Final Solution:
X* = [4. 1. 9. 0. 0. 0. 0.]
Object value = -2.0
---------------------------------
Times of iteration : 3
Solved in 0.000998 sec
```

测试三：

```
max (2, 2)
2
(1, -1) >= -1
(-0.5, 1) <= 2
Please input number of constraints A
Please input the next constraint
           format: (ai1, ai2, ..., ain) (<= or >= or =) bi

Please input the next constraint
           format: (ai1, ai2, ..., ain) (<= or >= or =) bi

[Hint] lp problem has UNBOUNDED SOLUTION!
```

测试四：

```
max (1, 1)
2
(1, -1 ) >= 0
(3, -1) <= -3
Please input number of constraints A
Please input the next constraint
           format: (ai1, ai2, ..., ain) (<= or >= or =) bi

Please input the next constraint
           format: (ai1, ai2, ..., ain) (<= or >= or =) bi

[Hint] Your artificial variable in the base, NO SOLUTION!
```

测试五：

```
Final Solution:
X* = [1.5 0.5 0.  0.  0.  0. ]
Object value = 2.25
---------------------------------
Times of iteration : 2
Solved in 0.000000 sec
```

## 六、运行环境

Python3，安装 numpy 包即可运行。程序为 LP.py，LP_debug.py 是带有中间输出的 debug 版本，只具有从控制台输入的功能。

## 七、总结

由于第一次写 python 代码，一边写，一边在查 python 语法的用法、numpy 包的用法以及正则表达式的用法，前前后后写此次程序花费了一天半的时间，用的时间比较久。最终完成的求解线性规划问题的程序可以选择从控制台输入，但是其缺点就是不能输入分数。因此程序提供了另一个接口，也就是从代码中手动设置输入，来求解线性规划问题