Python在单纯形法教学中的应用

刘华煜

（洛阳师范学院数学科学学院，河南 洛阳 471934）

摘　要：单纯形法是求解一般线性规划问题的基本方法，其求解步骤较多，单纯形表的构造与迭代过程较为复杂，是运筹学教学中的重点和难点。针对传统教学中学生难以直观理解单纯形表变化过程的问题，利用Python的矩阵计算和符号计算功能，设计了一种单纯形法求解过程的程序化演示方法。该方法以Excel文件为线性规划问题及单纯形表的输入、输出载体，结合NumPy，SymPy和Pandas等Python库，实现初始单纯形表构建、初始基可行解确定、检验数计算、换入基与换出基确定以及行变换等过程的自动化，并将每次迭代形成的单纯形表及相关决策过程输出至Excel文件。通过该方法，学生可以直观查看单纯形法的完整求解过程，加深对各个计算步骤及其逻辑关系的理解，从而提高单纯形法的学习效果。

关键词：运筹学教学；计算机辅助教学；单纯形法教学；大学数学

中图分类号：TP311　　**文献标识码：**A

Application of Python in simplex method teaching

LIU Huayu

（Faculty of Mathematical Sciences，Luoyang Normal University，Luoyang，Henan 471934，China）

Abstract：The simplex method is regarded as a basic method for solving general linear programming problems. Its solving steps are numerous，and the construction and iterative process of the simplex tableau are relatively complex，so it is considered a key and difficult point in operations research teaching. To address the problem that the changing process of the simplex tableau is difficult for students to intuitively understand in traditional teaching，a programmed demonstration method for the solving process of the simplex method was designed by utilizing the matrix computation and symbolic computation functions of Python. In this method，Excel files are adopted as the input and output carriers of linear programming problems and simplex tableaus，and the automation of processes such as the construction of the initial simplex tableau，the determination of the initial basic feasible solution，the calculation of test numbers，the determination of entering and leaving basic variables，and row transformations is realized by combining Python libraries such as NumPy，SymPy，and Pandas. The simplex tableau formed in each iteration and the related decision-making process are output to Excel files. Through this method，the complete solving process of the simplex method can be intuitively viewed by students，the understanding of each calculation step and its logical relationships can be deepened，and thus the learning effect of the simplex method can be improved.

Key words：operations research teaching，computer-assisted instruction，simplex method teaching，college mathematics

1　引言

单纯形法是“线性规划”课程中的重要内容，其求解过程通常通过单纯形表进行展示。然而，单纯形表涉及的数据较多，且随着迭代过程不断变化，学生需要依次完成基可行解确定、检验数计算、换入基和换出基选择以及行变换等操作，步骤较多且计算过程较为复杂，容易出现理解困难和计算错误。因此，如果能够针对任意线性规划问题完整展示单纯形表的迭代过程及相关决策过程，将有助于学生理解各步骤之间的关系，并熟练掌握单纯形法的求解方法[1-2]。

单纯形法的求解过程中涉及较多矩阵、向量运算，Python提供的科学计算和符号计算库可以有效简化矩阵运算、符号运算及数据处理过程。

2　单纯形法求解过程

单纯形法的基本求解过程可以概括为以下步骤。(1)构造初始单纯形表。(2)确定初始基可行解。(3)计算

检验数并进行最优性检验。若所有检验数均小于等于0,则当前基可行解为最优解,求解结束;否则进入下一步。(4)根据检验数确定换入基变量。(5)根据最小比值原则确定换出基变量。(6)以主元素为基准进行行变换,得到新的单纯形表,并更新基变量,随后返回步骤(3)。

3 NumPy及Sympy简介

NumPy是Python中常用的开源科学计算库,其核心数据结构为多维数组ndarray,能够高效处理大规模数值数据,并提供丰富的数学运算、线性代数和矩阵运算等功能。与Python原生列表相比,NumPy在数组运算和科学计算方面具有较高的效率,是Python科学计算生态中的重要基础库。线性规划问题通常采用矩阵形式表示,而单纯形法的求解过程涉及大量矩阵和向量运算。利用NumPy直接完成数组和矩阵的相关计算,可以在减少程序代码量的同时,降低手动编写矩阵运算过程产生计算错误的可能性[3-4]。

SymPy是Python中用于符号数学计算的开源库,支持代数、微积分、数论、离散数学以及矩阵等数学运算。与仅进行数值计算的数学库不同,SymPy可以将数学量作为符号进行处理,并支持数学表达式的化简、求导、积分以及方程求解等操作。在单纯形法教学演示中,部分线性规划问题采用大M法求解时,单纯形表中可能出现符号M;迭代过程中经常出现分数形式的计算结果。若直接采用浮点数进行计算,容易产生精度误差,并不利于学生观察和理解实际计算过程。因此,本文采用SymPy进行符号运算,使程序能够较好地保留M及分数等数学表达形式。

4 单纯形法求解过程的Python实现

4.1 给出初始单纯形表

初始单纯形表的数据由使用者输入。需要注意的是,采用大M法处理相关线性规划问题时,单纯形表中可能包含符号M,因此需要使用Python的符号计算库SymPy进行处理。

本文将Excel文件作为初始单纯形表的数据输入载体,其结构如图1所示。

	A	B	C	D	E	F	G	H
1		-3	0	1	0	0	-M	-M
2	4	1	1	1	1	0	0	0
3	1	-2	1	-1	0	-1	1	0
4	9	0	3	1	0	0	0	1

图1 初始单纯形表数据

图1中,第一行对应各变量在目标函数中的系数向量$\boldsymbol{C}$,第一列对应约束条件右端项向量$\boldsymbol{b}$,其余数据对应约束变量系数矩阵$\boldsymbol{A}$。

程序首先利用Pandas的read_excel函数以字符串形式读取Excel中的数据,其次将数据转换为SymPy符号表达式,以便正确处理输入数据中可能出现的符号M。随后,将目标函数系数向量、右端项向量和约束变量系数矩阵分别赋值给变量$\boldsymbol{C}$,$\boldsymbol{b}$和$\boldsymbol{A}$,为后续计算做好准备。

4.2 确定初始基可行解

在构造初始单纯形表时,需要保证矩阵$\boldsymbol{A}$中包含相应的单位矩阵。因此,可以通过查找矩阵$\boldsymbol{A}$中与单位向量对应的列,确定初始基变量。具体而言,程序依次构造单位矩阵中的各个单位列向量,并在矩阵$\boldsymbol{A}$中查找与其完全相同的列向量,将对应的列号保存至列表JI中。其代码如下:

```
def getJI(A):
    JI=[]
    rows,cols=A.shape
    for i in range(rows):
        ei=[0]*len(A)
        ei[i]=1
        for j in range(cols):
            col=[row[j] for row in A]
            if ei==col:
                JI+=[j]
                break
    return JI
```

4.3 计算检验数

确定基变量后,首先提取各基变量在目标函数中的系数,形成基变量系数向量$\boldsymbol{xs}$。然后,将$\boldsymbol{xs}$与矩阵$\boldsymbol{A}$的各列分别进行向量运算,并利用目标函数系数与相应乘积之差计算检验数向量sigma。其代码如下:

```
xs=np.array([C[i] for i in JI])
sigma=[C[j]-xs.dot(A[:,j]) for j in range(len(A[0]))]
```

4.4 检验数检验与确定换入基

计算检验数后,需要首先进行最优性检验。若所有检验数均小于等于0,则当前基可行解已经达到最优,求解过程结束;若存在大于0的检验数,则选择其中最大的检验数所对应的变量作为换入基变量。

程序实现时,可以先计算检验数的最大值,再判断该最大值是否小于等于0。若最大值小于等于0,则说明所有检验数均满足最优性条件;若最大值大于0,则将最大检验数对应的列确定为换入基列。

需要注意的是,当检验数中包含符号M时,不能直接按照普通数值进行大小比较。程序需要首先根据设定规则对M进行数值化处理,再完成检验数的比较,以确定换入基变量。

4.5 确定换出基

确定换入基后,根据最小比值原则确定换出基变量。首先取矩阵$\boldsymbol{A}$中换入基对应的列向量$\boldsymbol{P}$,其次仅对其中大于0的元素计算$\boldsymbol{b}$与$\boldsymbol{P}$对应元素的比值,取其中的最小值,其所在行对应的基变量即为换出基变量。其代码如下:

```
m=10000000000
```

```
row=-1
for i in range(len(A)):
    if A[i][col]>0 and b[i]/A[i][col] <= m:
        m=b[i]/A[i][col]
        row=i
```

其中，col为换入基所在列；row为换出基所在行。

4.6　行变换与基变量更新

确定换入基和换出基后，需要将换入基列与换出基行交点处的元素作为主元素进行行变换。首先，将主元素所在行的所有元素除以主元素，使主元素变为1；其次，利用该行对其他各行进行消元，使换入基所在列的其他元素全部变为0。经过上述处理后，即可得到新的单纯形表。其代码如下：

```
maine=A[row][col]
A[row]=A[row]/maine
b[row]/=maine
for r in range(len(A)):
    if r==row:
        continue
    chengshu=A[r][col]
    A[r]=A[r]-A[row]*chengshu
    b[r]=b[r]-b[row]*chengshu
```

行变换完成后，还需要更新基变量列表JI。由于第row行的基变量已被换入基变量替代，因此将相应位置修改为换入基所在列：

```
JI[row]=col
```

至此完成一次单纯形表迭代，并形成新的基可行解。随后重新计算检验数并进行最优性检验，直至得到最优解。

5　求解过程演示

5.1　单纯形表演示

单纯形表是单纯形法求解过程的核心，因此程序演示的主要内容也是不同迭代阶段单纯形表的变化。由于单纯形表本身具有表格形式，本文将Excel文件作为演示结果的输出载体，以便学生直观查看每次迭代产生的单纯形表。

单纯形表在Excel文件中的结构如图2所示。

13			-3	0	1	0	0	-M	-M
14	0	x4	0	0	0	0	1	-1/2	1/2	-1/2
15	0	x2	3	0	1	1/3	0	0	0	1/3
16	-3	x1	1	1	0	[2/3]	0	1/2	-1/2	1/6
17				0	0	3	0	3/2	-M - 3/2	1/2 - M

图2　单纯形表

单纯形表的整体结构按照教材中的标准形式进行设计，其中以中括号标记的红色数据为当前迭代过程中的主元素。

在单纯形法计算过程中，单纯形表中经常出现非整数结果。为使演示结果更贴合教材中的手算过程，程序采用分数形式显示相关计算结果。SymPy提供了相应的符号和分数运算功能，可以较好地满足这一需求。

5.2　单纯形表演示实现

单纯形表演示程序主要包括两个部分：(1)将单个单纯形表的数据组织为二维字符串数组；(2)将所有迭代产生的单纯形表依次输出至Excel文件。

5.3　将单纯形表数据放入数组

首先，创建NumPy二维数组com，并按照单纯形表的结构将$\boldsymbol{A}$，$\boldsymbol{b}$，$\boldsymbol{C}$和sigma等数据分别写入对应位置：

```
com[0,3:]=C
com[1:rows+1,2]=b
com[1:rows+1,3:]=A
com[-1,3:]=sigma
```

其次，将各基变量在目标函数中的系数以及基变量名称写入com：

```
xs=[C[i] for i in JI]
com[1:rows+1,0]=xs
xji=['x'+str(e+1) for e in JI]
com[1:rows+1,1]=xji
```

完成数据组织后，将com转换为Python字符串二维数组：

```
comlist=com.tolist()
comstr=[[str(e) for e in row]for row in comlist]
```

为在演示结果中突出显示主元素，根据主元素所在的行、列位置，将其数据用中括号括起来：

```
row+=1
col+=3
comstr[row][col]='['+comstr[row][col]+']'
```

5.4　将单纯形表数组输出至Excel

程序首先利用win32com包相关函数创建Excel文件，并将工作表中的单元格设置为文本格式：

```
app = win32.DispatchEx("Excel.Application")
wb = app.Workbooks.Add()
sheet=wb.Worksheets[0]
sheet.Cells.NumberFormat='@'
```

其次，通过循环将各次迭代产生的单纯形表依次写入Excel：

```
for data in datas:
    ……
```

假设datas为所有单纯形表组成的数组，则可以通过循环完成数据写入。在循环过程中，首先将前述Python字符串二维数组写入Excel单元格：

```
        rows=len(data)
        cols=len(data[0])
        r = sheet.Range(sheet.Cells(currow,1), sheet.Cells(currow+rows-1,cols))
        r.Value=data
```

随后，为当前单纯形表添加边框：

```
        r.Borders(7).LineStyle =1
        r.Borders(8).LineStyle =1
```

```
r.Borders(9).LineStyle =1
r.Borders(10).LineStyle =1
```

最后，通过查找以左中括号开头的数据定位主元素，并将对应单元格的字体设置为暗红色，以便在Excel中突出显示主元素：

```
mask = np.char.startswith(data, "[")
p=np.where(mask)
if len(p[0])! =0:
    row,col=p[0][0],p[1][0]
    sheet.Range(sheet.Cells(currow+row,1+col), sheet.Cells(currow+row,1+col)).Font.Color = 160
```

5.5 其他求解信息的演示

除单纯形表外，还需要向学生说明单纯形法求解过程中的换入基、换出基等决策。这些决策过程通常可以通过一行文字进行描述，因此程序将相关计算过程转换为文本，并写入Excel单元格，与单纯形表结合形成完整的过程演示。

以确定换出基为例，当换入基列向量$\boldsymbol{P}$为[3,-2,6]，右端项向量b为[3,1,2]时，只对$\boldsymbol{P}$中大于0的元素计算相应比值，可得到“换入基列向量$\boldsymbol{P}$为[3,-2,6]，$\boldsymbol{b}$=[3,1,2]。$\boldsymbol{P}$中大于0的元素分别与$\boldsymbol{b}$中对应元素相除，得到[1,不参与计算,1/3]。其中，最小值为1/3，对应第3行，因此第3行的基变量为换出基变量”。

6 结束语

本文利用Python的矩阵计算、符号计算和数据处理能力，实现了单纯形法主要求解步骤的程序化演示。程序能够完成初始单纯形表构建、初始基可行解确定、检验数计算、换入基与换出基确定以及行变换等操作，并将各次迭代形成的单纯形表及相关决策过程输出至Excel文件。

通过程序化演示，可以将传统教学中较为抽象、复杂的单纯形表迭代过程直观呈现出来，使学生能够观察每一步计算及单纯形表的变化，加深对单纯形法求解逻辑的理解。同时，该方法能够减少教师在课堂演示中的重复计算工作，将更多时间用于讲解求解原理和分析计算结果，从而提高单纯形法的教学效率和教学效果。

参考文献：

[1] 胡运权，胡祥培.运筹学基础及应用[M].北京：科学出版社，2021.
[2] SWEIGART A L.Python编程快速上手：让繁琐工作自动化[M].王海鹏，译.北京：人民邮电出版社，2021.
[3] 张若愚.Python科学计算[M].北京：机械工业出版社，2012.
[4] 唐利红.基于Python语言绘图的编程思维探究[J].软件，2025，46(9)：11-13.

作者简介：

刘华煜(1976—)，硕士，讲师，研究方向：计算机应用，E-mail：liuhuayu76@outlook.com。

(上接第36页)

中的实践研究[D].贵阳：贵州师范大学，2024.
[2] 胡伟伟.基于OBE理念的中职《JS程序设计》课程混合式教学研究与实践[D].桂林：广西师范大学，2024.
[3] 邓晓珂.基于OBE理念的混合式教学在中职《计算机网络技术》课程中的研究与实践[D].济南：山东师范大学，2024.
[4] 刘燕桃.基于OBE理念和项目式学习的中职《计算机应用基础》教学设计研究[D].武汉：华中师范大学，2021.

作者简介：

岑德飞(1982—)，本科，讲师，研究方向：计算机应用。