**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Государственное образовательное учреждение

высшего профессионального образования

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра Математической кибернетики и информационных технологий

Транспортная задача

по дисциплине «Теория оптимизации»

Выполнила: студ. гр. ЗМПП1901

Иванова Е.В.

Проверил: И.о. зав. кафедрой: доц., к.т.н. Городничев Михаил Геннадьевич

Москва 2019

**Постановка задачи**

Есть фирма, которая оказывает транспортные услуги, есть поставщики товара со складами в разных шести городах, причём объёмы однородной продукции на этих складах соответственно равны a1÷a6. Есть и потребители в других шести городах которым нужно привести товар от поставщиков в объёмах b1÷b6 соответственно. Известны также стоимости доставки с1÷с9 товаров от поставщиков к потребителям.

**Алгоритм решения транспортной задачи с классическими условиями**

Рассмотрим пример исходных данных в виде таблицы (матрицы).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **b1** | **b2** | **b3** | **b4** | **b5** | **b6** |
| **a1** | С[0]\*X[0] | С[1]\*X[1] | С[2]\*X[2] | С[3]\*X[3] | С[4]\*X[4] | С[5]\*X[5] |
| **a2** | С[6]\*X[6] | С[7]\*X[7] | С[8]\*X[8] | С[9]\*X[9] | С[10]X[10] | С[11]X[11] |
| **a3** | С[12]X[12] | С[13]X[13] | С[14]X[14] | С[15]X[15] | С[16]X[16] | С[17]X[17] |
| **a4** | С[18]X[18] | С[19]X[19] | С[20]X[20] | С[21]X[21] | С[22]X[22] | С[23]X[23] |
| **a5** | С[24]X[24] | С[25]X[25] | С[26]X[26] | С[27]X[27] | С[28]X[28] | С[29]X[29] |
| **a6** | С[30]X[30] | С[31]X[31] | С[32]X[32] | С[33]X[33] | С[34]X[34] | С[35]X[35] |

где C– список стоимости перевозки единицы товара от заказчиков к потребителям;

X – список объёмов перевозимых товаров, обеспечивающих минимальные затраты;

a– список объёмов однородных товаров на складах поставщиков;

b– список объёмов спроса заказчиками однородных товаров.

Следовательно, функция цели:

F(x)= C[0]\*x[0] + C[1]\*x[1] +C[2]\* x[2] +C[3]\*x[3] + C[4]\*x[4] +C[5]\* x[5]+C[6]\*x[6] + C[7]\*x[7] +C[8]\* x[8]+ [9]\*X[9]+ [10]\*X[10]+ [11]\*X[11] + [12]\*X[12] + [13]\*X[13]+ [14]\*X[14]+ [15]\*X[15] + C[16]\*x[16] + C[17]\*x[17] +C[18]\* x[18] +C[19]\*x[19] + C[20]\*x[20] +C[21]\* x[21]+C[22]\*x[22] + C[23]\*x[23] +C[24]\* x[24]+ [25]\*X[25]+ [26]\*X[26]+ [27]\*X[27] + [28]\*X[28] + [29]\*X[29]+ [30]\*X[30]+ [31]\*X[31]+ [32]\*X[32] + [33]\*X[33]+ [34]\*X[34]+ [35]\*X[35]

Условия, которые записываются в виде неравенств:  
x[0] + x[1] +x[2] + x[3] + x[4] +x[5] <= a1

x[6] + x[7] + x[8] + x[9] + x[10] + x[11] <= a2  
x[12] + x[13] + x[14] + x[15] + x[16] + x[17] <= a3

x[18] + x[19] + x[20] + x[21] + x[22] + x[23] <= a4  
x[24] + x[25] + x[26] + x[27] + x[28] + x[29] <= a5  
x[30] + x[31] + x[32] + x[33] + x[34] + x[35] <= a6

Условия, которые записываются в виде равенств:  
x[0] + x[6] + x[12] + x[18] + x[24] +x[30] == b1

x[1] + x[7] + x[13] + x[19] + x[25] +x[31] == b2  
x[2] + x[8] + x[14] + x[20] + x[26] +x[32] == b3

x[3] + x[9] + x[15] + x[21] + x[27] +x[33] == b4

x[4] + x[10] + x[16] + x[22] + x[28] +x[34] == b5  
x[5] + x[11] + x[17] + x[23] + x[29] +x[35] == b6

Для линейного программирования в Python известны три специализированные библиотеки. Необходимо оценить быстродействие и удобство использования библиотек.

**Оптимизация с библиотекой pulp**

from pulp import \*

import time

import random

start = time.time()

var = ["x" + str(i) for i in range(36)]

x = [pulp.LpVariable(i, lowBound=0) for i in var]

а= []

for i in range(6):

а.append(random.randint(1,100))

b= []

for i in range(6):

b.append(random.randint(1,70))

с= []

for i in range(36):

с.append(random.randint(1,10))

problem = pulp.LpProblem('0',pulp.LpMaximize)

problem += -lpSum([с[i]\*x[i] for i in range(36)]), "Функция цели"

problem +=x[0] + x[1] +x[2]+ x[3] + x[4] +x[5] <= а[0],"1"

problem +=x[6] + x[7] +x[8]+ x[9] + x[10] +x[11] <= а[1], "2"

problem +=x[12] + x[13]+ x[14]+ x[15] + x[16] +x[17] <= а[2], "3"

problem +=x[18] + x[19] +x[20]+ x[21] + x[22] +x[23] <= а[3],"4"

problem +=x[24] + x[25] +x[26]+ x[27] + x[28] +x[29] <= а[4], "5"

problem +=x[30] + x[31]+ x[32]+ x[33] + x[34] +x[35] <= а[5], "6"

problem +=x[0]+ x[6]+ x[12]+ x[18]+x[24]+ x[30] == b[0], "7"

problem +=x[1]+ x[7]+ x[13]+ x[19]+x[25]+ x[31] == b[1], "8"

problem +=x[2]+ x[8]+ x[14]+ x[20]+x[26]+ x[32] == b[2], "9"

problem +=x[3]+ x[9]+ x[15]+ x[21]+x[27]+ x[33] == b[3], "10"

problem +=x[4]+ x[10]+ x[16]+ x[22]+x[28]+ x[34] == b[4], "11"

problem +=x[5]+ x[11]+ x[17]+ x[23]+x[29]+ x[35] == b[5], "12"

problem.solve()

print ("Результат:")

for variable in problem.variables():

print (variable.name, "=", variable.varValue)

print ("Стоимость доставки:")

print (abs(value(problem.objective)))

stop = time.time()

print ("Время :")

print(stop - start)

**Результат:**

x0 = 15.0

x1 = 0.0

x10 = 31.0

x11 = 0.0

x12 = 0.0

x13 = 26.0

x14 = 0.0

x15 = 1.0

x16 = 0.0

x17 = 0.0

x18 = 0.0

x19 = 17.0

x2 = 0.0

x20 = 0.0

x21 = 0.0

x22 = 0.0

x23 = 0.0

x24 = 6.0

x25 = 0.0

x26 = 0.0

x27 = 0.0

x28 = 0.0

x29 = 0.0

x3 = 0.0

x30 = 5.0

x31 = 0.0

x32 = 58.0

x33 = 0.0

x34 = 10.0

x35 = 0.0

x4 = 0.0

x5 = 14.0

x6 = 0.0

x7 = 0.0

x8 = 0.0

x9 = 16.0

Стоимость доставки:

663.0

Время :

0.07116103172302246

**Оптимизация с библиотекой scipy.optimize**

from scipy.optimize import linprog

import time

import random

start = time.time()

а= []

for i in range(6):

а.append(random.randint(1,100))

b= []

for i in range(6):

b.append(random.randint(1,70))

с= []

for i in range(36):

с.append(random.randint(1,10))

A\_ub = [[1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1]]

A\_eq = [[1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0],

[0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0],

[0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0],

[0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0],

[0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0],

[0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1]]

print(linprog(с, A\_ub, а, A\_eq, b))

stop = time.time()

print ("Время :")

print(stop - start)

**Результат:**

con: array([2.01358617e-07, 4.16141077e-07, 8.72554118e-08, 1.34239087e-07,

2.08070603e-07, 1.87934724e-07])

fun: 472.9999990292109

message: 'Optimization terminated successfully.'

nit: 6

slack: array([3.08325966e-07, 2.79606393e+01, 1.42060972e+01, 3.06777648e-09,

1.90730965e+01, 1.76016788e+00])

status: 0

success: True

x: array([9.84070214e-09, 4.79999996e+01, 3.28103592e-08, 8.62340172e-09,

6.94094292e-09, 5.39356551e-08, 1.37477844e-08, 1.15621878e-08,

1.89999998e+01, 1.40873042e-08, 8.68889897e-09, 1.50393608e+01,

8.82044774e-09, 6.07309613e+00, 2.33131362e-08, 8.80597181e-09,

3.47601677e+01, 1.89606389e+01, 6.83359431e-09, 1.99999995e+00,

1.11136604e-08, 7.53848798e-09, 8.45125906e-09, 1.10659285e-08,

3.59999997e+01, 1.19269039e+01, 9.96576844e-09, 2.59999998e+01,

8.86908322e-09, 1.16792880e-08, 1.61117693e-08, 2.47183493e-08,

7.34407463e-09, 1.60723711e-08, 2.23983204e+00, 1.51013258e-08])

Время :

0.008198261260986328

**Оптимизация с библиотекой cvxopt**

from cvxopt.modeling import variable, op

import time

import random

start = time.time()

а= []

for i in range(6):

а.append(random.randint(1,100))

b= []

for i in range(6):

b.append(random.randint(1,70))

с= []

for i in range(36):

с.append(random.randint(1,10))

x = variable(36, 'x')

z=(с[0]\*x[0] + с[1]\*x[1] +с[2]\* x[2] +с[3]\*x[3] + с[4]\*x[4] +с[5]\*x[5] + с[6]\*x[6] + с[7]\*x[7] +с[8]\* x[8] + с[9]\* x[9]+ с[10]\*x[10] + с[11]\*x[11] + с[12]\*x[12] + с[13]\*x[13] + с[14]\*x[14]+ с[15]\*x[15]+ с[16]\*x[16]+с[17]\*x[17]+с[18]\*x[18] + с[19]\*x[19] + с[20]\*x[20] + с[21]\*x[21] +с[22]\*x[22] +с[23]\*x[23]+ с[24]\*x[24]+ с[25]\*x[25]+с[26]\*x[26]+с[27]\*x[27] +с[28]\*x[28] + с[29]\*x[29] + с[30]\*x[30] + с[31]\*x[31] + с[32]\*x[32]+с[33]\*x[33]+ с[34]\*x[34]+с[35]\*x[35])

mass1 = (x[0] + x[1] +x[2] + x[3] + x[4] +x[5] <= а[0])

mass2 = (x[6] + x[7] +x[8] + x[9] + x[10] +x[11] <= а[1])

mass3 = (x[12] + x[13] + x[14] + x[15] + x[16] +x[17] <= а[2])

mass4 = (x[18] + x[19] +x[20] + x[21] + x[22] +x[23] <= а[3])

mass5 = (x[24] + x[25] +x[26] + x[27] + x[28] +x[29] <= а[4])

mass6 = (x[30] + x[31] + x[32] + x[33] + x[34] +x[35] <= а[5])

mass7 = (x[0] + x[6] + x[12] + x[18] + x[24] +x[30] == b[0])

mass8 = (x[1] + x[7] + x[13] + x[19] + x[25] +x[31] == b[1])

mass9 = (x[2] + x[8] + x[14] + x[20] + x[26] +x[32]== b[2])

mass10 = (x[3] + x[9] + x[15] + x[21] + x[27] +x[33] == b[3])

mass11 = (x[4] +x[10] + x[16] + x[22] + x[28] +x[34] == b[4])

mass12 = (x[5] + x[11] + x[17] + x[23] + x[29] +x[35]== b[5])

x\_non\_negative = (x >= 0)

problem = op (z, [mass1, mass2, mass3, mass4, mass5, mass6, mass7, mass8, mass9, mass10, mass11, mass12, x\_non\_negative])

problem.solve(solver='glpk')

problem.status

print("Результат:")

print(x.value)

print("Стоимость доставки:")

print(problem.objective.value()[0])

stop = time.time()

print ("Время :")

print(stop - start)

**Результат:**

[ 0.00e+00]

[ 4.00e+01]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 6.20e+01]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 1.30e+01]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

[ 1.00e+01]

[ 1.60e+01]

[ 0.00e+00]

[ 7.00e+00]

[ 2.00e+00]

[ 6.90e+01]

[ 0.00e+00]

[ 0.00e+00]

[ 0.00e+00]

Стоимость доставки:

425.0

Время :

0.0175168514251709

**Вывод**

Проанализировав решение задачи линейного программирования с помощью трех библиотек аналогичного назначения, можно сделать вывод о том, что при использовании библиотеки scipy.Optimize ввод данных более компактен, а также ее использование приводит к более быстродейственному решению задачи, чем использование двух других библиотек.

**Список используемых источников и литературы**

1. Теория и методы оптимизации: учебное пособие/ Кочегурова Е.А.
2. <https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html>
3. https://buildmedia.readthedocs.org/media/pdf/cvxopt/dev/cvxopt.pdf
4. https://pythonhosted.org/PuLP/