ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

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

Кафедра Математической кибернетики и информационных технологий

Домашняя работа

по дисциплине

Теория оптимизации

Выполнил: студент гр. 3МПП1901 Иванюк П. Г.,

Проверил: к. т. н. Городничев М. Г.

Москва 2019 г.

**Транспортная задача**

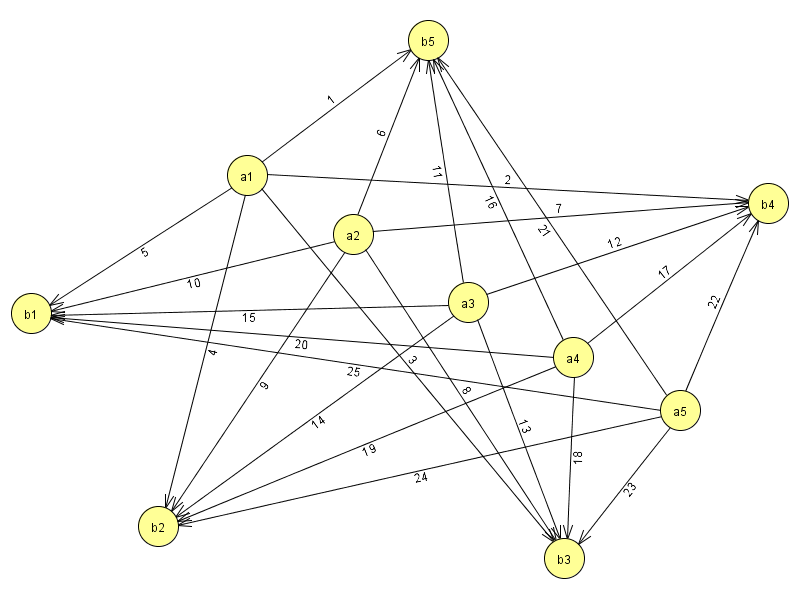
В разных пяти городах находятся со склады с одинаковыми товарами - a1, a2, a3, а4, а5.

В пяти других городах находятся потребители, которым нужно привести товар со складов в объёмах b1, b2, b3, b4, b5 соответственно.

Стоимости каждой товара со склада к потребителю указаны в таблице.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Граф транспортной задачи**



**Решение**

Обозначим через Xn количество перевозимого груза, тогда функцией цели будет общая стоимость перевозки:

F(x)=c1\*x1+c2\*x2+c3\*x3+c4\*x4+c5\*x5+c6\*x6+c7\*x7+c8\*x8+c9\*x9+c10\*x10+c11\*x11+c12\*x12+c13\*x13+c14\*x14+c15\*x15+c16\*x16+c17\*x17+c18\*x18+c19\*x19+c20\*x20+c21\*x21+c22\*x22+c23\*x23+c24\*x24+c25\*x25

Условия, описывающие неравенства:

x1+x2+x3 + x4 + x5 <= 50

x6+x7+x8 + x9 + x10<= 60

x11 +x12 +x13 + x14 + x15 <= 70

x16 + x17 + x18 + x19 + x20 <= 30

x21 + x22 + x23 + x24 + x25 <= 20

Условия, которые записываются в виде равенств:

x1+ x6+ x11 + x16 + x21 <= 55

x2 +x7 +x12 + x17 + x22 <= 62

x3 +x8 +x13 + x18 + x23 <= 68

x4 + x9 + x14 + x19 + x24 <= 25

x5 + x10 + x15 + x20 + x25 <= 20

**код на python, с использованием библиотеки pulp.**

#coding: utf8

from pulp import \*

import time

s = time.time()

x1 = pulp.LpVariable("x1", lowBound=0)

x2 = pulp.LpVariable("x2", lowBound=0)

x3 = pulp.LpVariable("x3", lowBound=0)

x4 = pulp.LpVariable("x4", lowBound=0)

x5 = pulp.LpVariable("x5", lowBound=0)

x6 = pulp.LpVariable("x6", lowBound=0)

x7 = pulp.LpVariable("x7", lowBound=0)

x8 = pulp.LpVariable("x8", lowBound=0)

x9 = pulp.LpVariable("x9", lowBound=0)

x10 = pulp.LpVariable("x10", lowBound=0)

x11 = pulp.LpVariable("x11", lowBound=0)

x12 = pulp.LpVariable("x12", lowBound=0)

x13 = pulp.LpVariable("x13", lowBound=0)

x14 = pulp.LpVariable("x14", lowBound=0)

x15 = pulp.LpVariable("x15", lowBound=0)

x16 = pulp.LpVariable("x16", lowBound=0)

x17 = pulp.LpVariable("x17", lowBound=0)

x18 = pulp.LpVariable("x18", lowBound=0)

x19 = pulp.LpVariable("x19", lowBound=0)

x20 = pulp.LpVariable("x20", lowBound=0)

x21 = pulp.LpVariable("x21", lowBound=0)

x22 = pulp.LpVariable("x22", lowBound=0)

x23 = pulp.LpVariable("x23", lowBound=0)

x24 = pulp.LpVariable("x24", lowBound=0)

x25 = pulp.LpVariable("x25", lowBound=0)

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

p += -5\*x1-4\*x2-3\*x3-2\*x4-1\*x5-10\*x6-9\*x7-8\*x8-7\*x9-6\*x10-15\*x11-14\*x12-13\*x13-12\*x14-11\*x15-20\*x16-21\*x17-22\*x18-23\*x19-24\*x20-25\*x21-24\*x22-23\*x23-22\*x24-21\*x25

p += x1+x2+x3+x4+x5 <= 50

p += x6+x7+x8+x9+x10 <= 60

p += x11+x12+x13+x14+x15 <= 70

p += x16+x17+x18+x19+x20 <= 30

p += x21+x22+x23+x24+x25 <= 20

p += x1+x6+x11+x16+x21 == 55

p += x2+x7+x12+x17+x22 == 62

p += x3+x8+x13+x18+x23 == 68

p += x4+x9+x14+x19+x24 == 25

p += x5+x10+x15+x20+x25 == 20

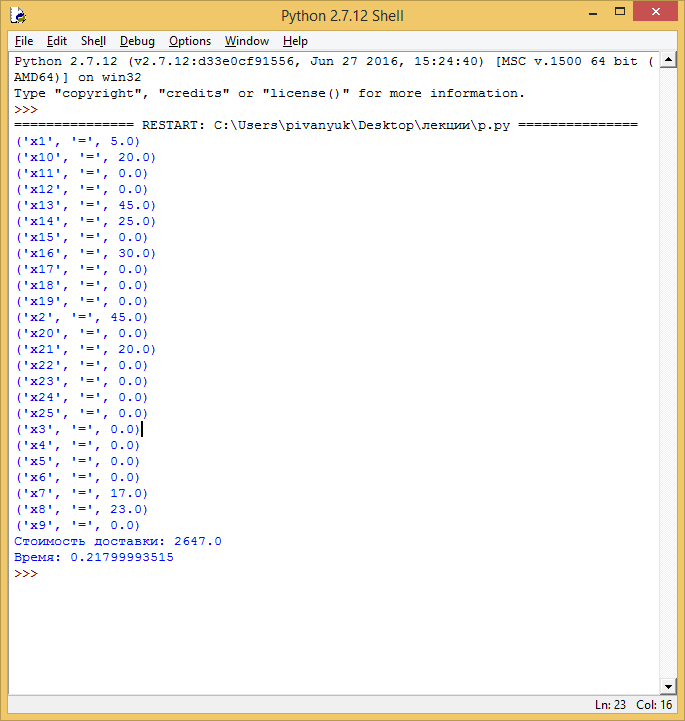
p.solve()

for v in p.variables():

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

print u"Стоимость доставки:", abs(value(p.objective))

print u"Время:", (time.time() - s)



**код на python, с использованием библиотеки cvxopt.**

#coding: utf8

from cvxopt.modeling import variable, op

import time

start = time.time()

x = variable(25, 'x')

z=(5\*x[0]+4\*x[1]+3\*x[2]+2\*x[3]+1\*x[4]+10\*x[5]+9\*x[6]+8\*x[7]+7\*x[8]+6\*x[9]+15\*x[10]+14\*x[11]+13\*x[12]+12\*x[13]+11\*x[14]+20\*x[15]+19\*x[16]+18\*x[17]+17\*x[18]+16\*x[19]+25\*x[20]+24\*x[21]+23\*x[22]+22\*x[23]+21\*x[24])

mass1 = (x[0]+x[1]+x[2]+x[3]+x[4] <= 50)

mass2 = (x[5]+x[6]+x[7]+x[8]+x[9] <= 60)

mass3 = (x[10]+x[11]+x[12]+x[13]+x[14] <= 70)

mass4 = (x[15]+x[16]+x[17]+x[18]+x[19] <= 30)

mass5 = (x[20]+x[21]+x[22]+x[23]+x[24] <= 20)

mass6 = (x[0]+x[5]+x[10]+x[15]+x[20] == 55)

mass7 = (x[1]+x[6]+x[11]+x[16]+x[21] == 62)

mass8 = (x[2]+x[7]+x[12]+x[17]+x[22] == 68)

mass9 = (x[3]+x[8]+x[13]+x[18]+x[23] == 25)

mass10 = (x[4]+x[9]+x[14]+x[19]+x[24] == 20)

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

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

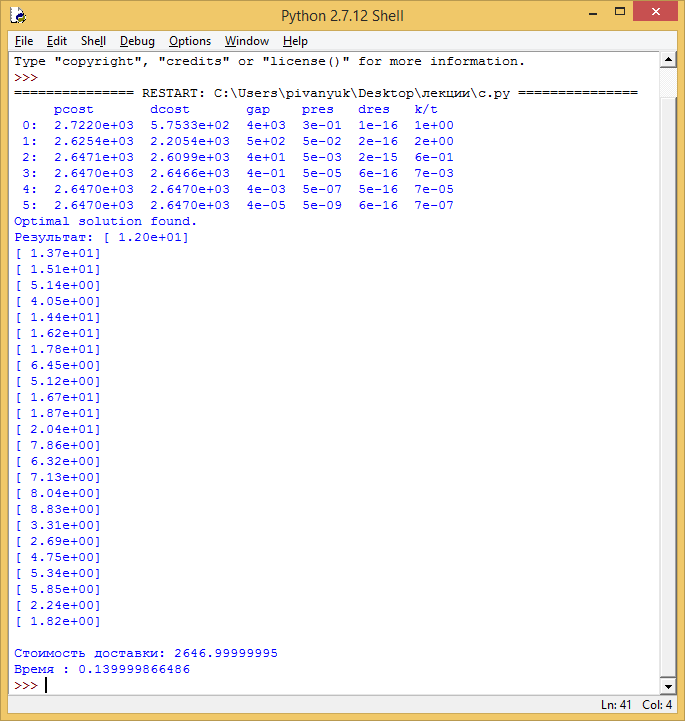
p.solve(solvers='glpk')

p.status

print u"Результат:",x.value

print u"Стоимость доставки:",p.objective.value()[0]

print u"Время :",time.time() - start



**код на python, с использованием библиотеки scipy.optimize.**

#coding: utf8

from scipy.optimize import linprog

import time

start = time.time()

cost = [5,4,3,2,1,10,9,8,7,6,15,14,13,12,11,20,19,18,17,16,25,24,23,22,21]

A\_ub = [[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],

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

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

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

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

b\_ub = [50,60,70,30,20]

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

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

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

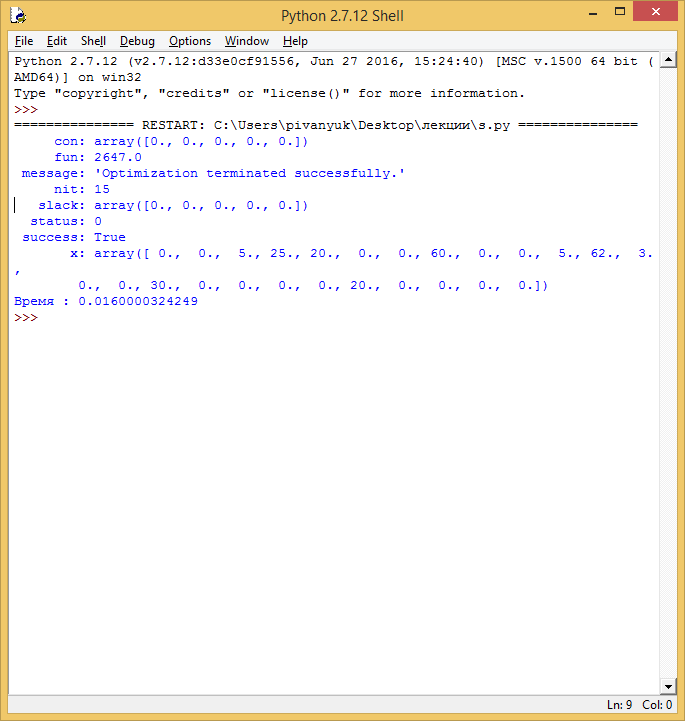
[0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0],

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

b\_eq = [55,62,68,25,20]

print linprog(cost, A\_ub, b\_ub, A\_eq, b\_eq)

print u"Время :", time.time() - start



**Вывод**

Вычисление транспортной задачи с помощью библиотеки pulp занимает 0,218 секунд.

Вычисление транспортной задачи с помощью библиотеки cvxopt занимает 0,14 секунд.

Вычисление транспортной задачи с помощью библиотеки scipy.optimize занимает 0,016 секунд.

Таким образом выбор библиотеки scipy.optimize приводит к наибольшему быстродействию при полученных, одинаковых результатах равных 2647,0.

А также использование библиотеки scipy.optimize приводит к компактности по вводу данных. Результирующий вывод библиотеки данной библиотеки выполняется в короткой и компактной форме. Таким образом библиотека scipy.optimize позволяет наиболее быстрее и проще решать задачи оптимизации.

**Литература**

1. С. А. Ашманов Линейное программирование - М.: Наука. Главная редакция физико-математической литературы, 1981. - 340 с.
2. В. Г. Карманов Математическое программирование: Учеб. пособие. - 5-е издание стереотип. - М.: ФИЗМАТЛИТ, 2004. - 2664 с.
3. Электронный ресурс - https://sohabr.net/habr/post/330138/