ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

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

Кафедра Математическая кибернетика и информационные технологии

САМОСТОЯТЕЛЬНАЯ РАБОТА

по дисциплине

Теория оптимизации

на тему:

Решение транспортной задачи с использованием библиотек Python: pulp, cvxopt and optimize.

Выполнил: студент гр. 3МПП1901

Рошорвиджов С. Р.

Принял: доцент, к.т.н.

Городничев М. Г.

Москва 2019 г.

**Содержание**

1. Постановка задачи2
2. Решение задачи 4
3. Листинг программы (Python)6
4. Сравнение библиотек7

Список использованных источников8

1. **Постановка задачи**

Обозначения:

an – количество товара на складе;

bn – количество заказанного товара;

xn – количество перевозимого товара;

dn – стоимость доставки ($);

Для определения стоимости доставки (dn) построим граф, где вершинами будут склады (an) и заказчики товаров (bn), а вес ребер между ними – стоимость доставки (dn). На основе графа заполним таблицу 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **an / bn** | **b1 = 20** | **b2 = 70** | **b3 = 43** | **b4 = 25** | **b5 = 40** |
| **a1 = 30** | d1 = 5 | d2 = 2 | d3 = 1 | d4 = 3 | d5 = 7 |
| **a2 = 50** | d6 = 4 | d7 = 4 | d8 = 3 | d9 = 5 | d10 = 6 |
| **a3 = 45** | d11 = 5 | d12 = 5 | d13 = 4 | d14 = 8 | d15 = 1 |
| **a4 = 70** | d16 = 1 | d17 = 7 | d18 = 8 | d19 = 3 | d20 = 4 |
| **a5 = 65** | d21 = 3 | d22 = 6 | d23 = 2 | d24 = 7 | d25 = 5 |

Таблица 1 – Стоимость доставки

1. **Решение задачи**

Если обозначить через dn количество перевозимого груза, тогда функцией цели будет общая стоимость перевозки:

F(x)=d1\*x1+d2\*x2+ ... + d25\*x25.

Условия:

xn > 0;

dn > 0;

x1 + x2 + x3 + x4 + x5 ≤ 30;

x6 + x7 + x8 + x9 + x10 ≤50;

x11 + x12 + x13 + x14 + x15 ≤45;

x16 + x17 + x18 + x19 + x20 ≤70;

x21 + x22 + x23 + x24+ x25 ≤65;

x1 + x6 + x11 + x16 + x21 = 20;

x2 + x7 + x12 + x17 + x22 = 70;

x3 + x8 + x13 + x18 + x23 = 43;

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

x5 + x10 + x15 + x20+ x25 = 40.

1. **Листинг программы (Python)**

from pulp import \*

import time

start = 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)

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

problem += -5\*x1 - 2\*x2 - 1\*x3 - 3\*x4 -7\*x5 -4\* x6 -4\*x7 -3\*x8 -5\*x9 -6\*x10 -5\*x11 -5\*x12-4\*x13-8\*x14-1\*x15-1\*x16-7\*x17-8\*x18-3\*x19-4\*x20-3\*x21-6\*x22-2\*x23-7\*x24-5\*x25, "Функция цели"

problem += x1 + x2 + x3 + x4 + x5 <= 30,"1"

problem += x6 + x7 + x8 + x9 + x10 <= 50,"2"

problem += x11 + x12 + x13 + x14 + x15 <= 45,"3"

problem += x16 + x17 + x18 + x19 + x20 <= 70,"4"

problem += x21 + x22 + x23 + x24 + x25 <= 65,"5"

problem += x1 + x6 + x11 + x16 + x21 == 20,"6"

problem += x2 + x7 + x12 + x17 + x22 == 70,"7"

problem += x3 + x8 + x13 + x18 + x23 == 43,"8"

problem += x4 + x9 + x14 + x19 + x24 == 25,"9"

problem += x5 + x10 + x15 + x20 + x25 == 40,"10"

problem.solve()

print ("Результат:")

for variable in problem.variables():

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

print ("Стоимость доставки:")

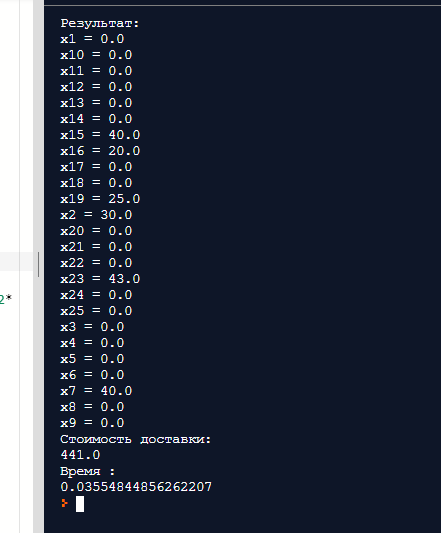
print (abs(value(problem.objective)))

stop = time.time()

print ("Время :")

print(stop - start)

Листинг 1 – Решение с использованием библиотеки pulp



Листинг 2 – Результат выполнения (pulp)

from cvxopt.modeling import variable, op

import time

start = time.time()

x = variable(25, 'x')

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

mass1 = (x[0] + x[1] + x[2] + x[3] + x[4] <= 30)

mass2 = (x[5] + x[6] + x[7] + x[8] + x[9] <= 50)

mass3 = (x[10] + x[11] + x[12] + x[13] + x[14] <= 45)

mass4 = (x[15] + x[16] + x[17] + x[18] + x[19] <= 70)

mass5 = (x[20] + x[21] + x[22] + x[23] + x[24] <= 65)

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

mass7 = (x[1] + x[6] + x[11] + x[16] + x[21] == 70)

mass8 = (x[2] + x[7] + x[12] + x[17] + x[22] == 43)

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

mass10 = (x[4] + x[9] + x[14] + x[19] + x[24] == 40)

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

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

problem.solve(solver='glpk')

problem.status

print("Результат:")

print(x.value)

print("Стоимость доставки:")

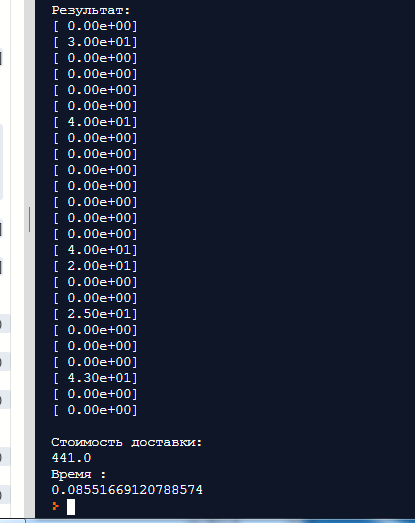
print(problem.objective.value()[0])

stop = time.time()

print ("Время :")

print(stop - start)

Листинг 3 – Решение с использованием библиотеки cvxopt



Листинг 4 - Результат выполнения (cvxopt)

from scipy.optimize import linprog

import time

start = time.time()

d = [5, 2, 1,3,7,4,4,3,5,6,5,5,4,8,1,1,7,8,3,4,3,6,2,7,5]

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\_eq = [20,70,43,25,40]

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\_ub = [30,50,45,70,65]

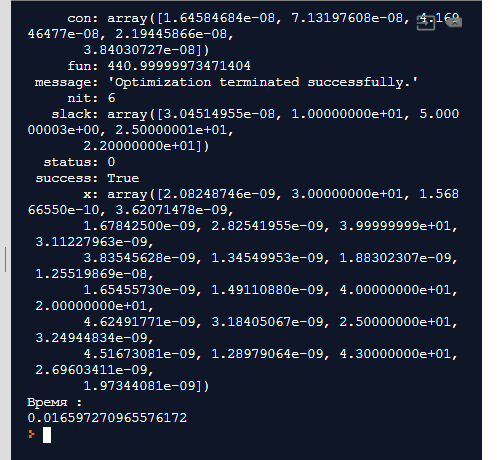
print(linprog(d, A\_ub, b\_ub, A\_eq, b\_eq))

stop = time.time()

print ("Время :")

print(stop - start)

Листинг 5 - Решение с использованием библиотеки scipy.optimize



Листинг 4 - Результат выполнения (scipy.optimize)

1. **Сравнение библиотек**

Анализ решения двух типовых задач линейного программирования с помощью трёх библиотек аналогичного назначения не вызывает сомнения в выборе библиотеки scipy. optimize, как лидера по компактности ввода данных и быстродействию.

**Список использованных источников**

1. Кочегурова Е.А. Теория и методы оптимизации. – 171 с.
2. Ашманов С.А. Линейное программирование. – 171 с. Москва «Наука». – 303 с.
3. Решение задач линейного программирования с использованием Python [Электронный ресурс]. URL: <https://habr.com/ru/post/330648> .