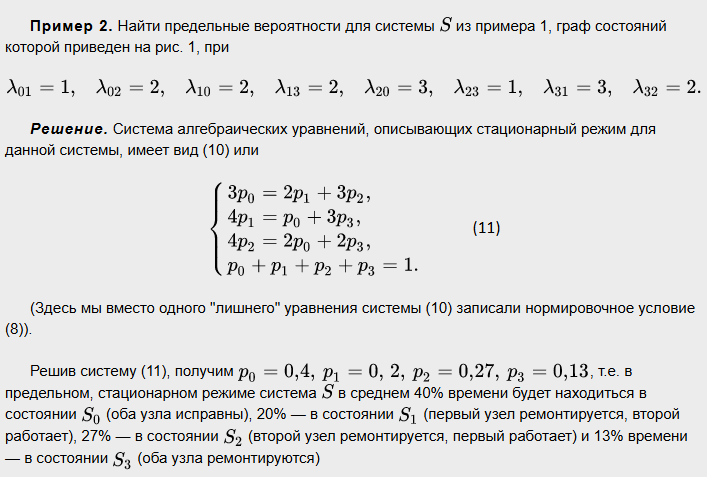
# Лабораторная работа №2

Студент: Болдинов Алексей ЭФМО-02-24

Сайт: <https://mathhelpplanet.com/static.php?p=uravneniya-kolmogorova>

**Пример 2. (с сайта)**

Постановка задачи: см. файл Решение задач оптимизации в среде MS Excel (Александр Леоненков) стр. 201 (по учебнику).



Решение на Питоне:

from cvxopt.modeling import variable, op  
import time  
  
# start = time.time()  
p = variable(4, 'p')  
lam = [[0, 1, 2, 0], [2, 0, 0, 2], [3, 0, 0, 1], [0, 3, 2, 0]]  
z = 0  
mass1 = ((lam[0][1] + lam[0][2]) \* p[0] == lam[1][0] \* p[1] + lam[2][0] \* p[2])  
mass2 = ((lam[1][0] + lam[1][3]) \* p[1] == lam[0][1] \* p[0] + lam[3][1] \* p[3])  
mass3 = ((lam[2][0] + lam[2][3]) \* p[2] == lam[0][2] \* p[0] + lam[3][2] \* p[3])  
mass4 = ((lam[3][1] + lam[3][2]) \* p[3] == lam[1][3] \* p[1] + lam[2][3] \* p[2])  
mass5 = (p[0] + p[1] + p[2] + p[3] == 1)  
x\_0\_1 = (0 <= p <= 1)  
problem = op(z, [mass1, mass2, mass3, mass4, mass5, x\_0\_1])  
problem.solve(solver='glpk')  
  
print("Результат Xopt:")  
for i in p.value:  
 round(i, 2)  
# print("Стоимость доставки:", problem.objective.value()[0])  
# stop = time.time()  
# print("Время :", stop - start)

Ответ программы:

GLPK Simplex Optimizer 5.0

9 rows, 4 columns, 20 non-zeros

0: obj = 0.000000000e+00 inf = 1.000e+00 (1)

4: obj = 0.000000000e+00 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

Результат Xopt:

0.4

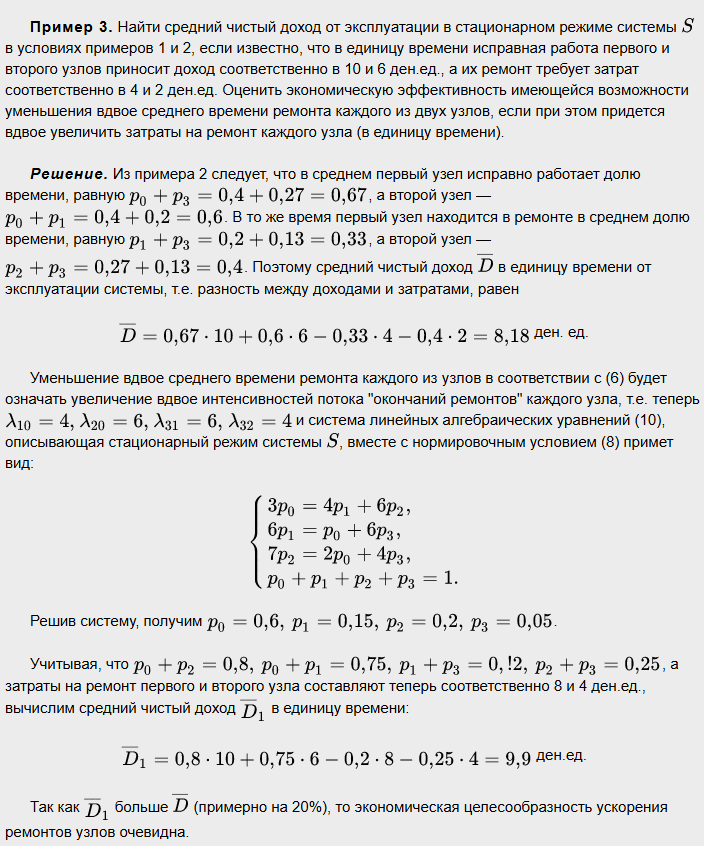
0.2

0.27

0.13

## Пример 3 (с сайта)

Постановка задачи: см. файл Решение задач оптимизации в среде MS Excel (Александр Леоненков) стр. 366 (по учебнику).



Решение на Питоне:

def primer2():  
 from cvxopt.modeling import variable, op  
 import time  
  
 # start = time.time()  
 p = variable(4, 'p')  
 lam = [[0, 1, 2, 0], [2, 0, 0, 2], [3, 0, 0, 1], [0, 3, 2, 0]]  
 z = 0  
 mass1 = ((lam[0][1] + lam[0][2]) \* p[0] == lam[1][0] \* p[1] + lam[2][0] \* p[2])  
 mass2 = ((lam[1][0] + lam[1][3]) \* p[1] == lam[0][1] \* p[0] + lam[3][1] \* p[3])  
 mass3 = ((lam[2][0] + lam[2][3]) \* p[2] == lam[0][2] \* p[0] + lam[3][2] \* p[3])  
 mass4 = ((lam[3][1] + lam[3][2]) \* p[3] == lam[1][3] \* p[1] + lam[2][3] \* p[2])  
 mass5 = (p[0] + p[1] + p[2] + p[3] == 1)  
 x\_0\_1 = (0 <= p <= 1)  
 problem = op(z, [mass1, mass2, mass3, mass4, mass5, x\_0\_1])  
 problem.solve(solver='glpk')  
  
 print("Результат Xopt:")  
 for i in p.value:  
 print(round(i, 2))  
 # stop = time.time()  
 # print("Время :", stop - start)  
 return [round(i, 2) for i in p.value]  
  
  
def primer3():  
 from cvxopt.modeling import variable, op  
 import time  
 pEarly = primer2()  
 print(pEarly)  
 \_D = (pEarly[0] + pEarly[2]) \* 10 + (pEarly[0] + pEarly[1]) \* 6 - (pEarly[1] + pEarly[3]) \* 4 - (  
 pEarly[2] + pEarly[3]) \* 2  
  
 print("\_D = ", \_D)  
  
 p = variable(4, 'p')  
 lam = [[0, 1, 2, 0], [4, 0, 0, 2], [6, 0, 0, 1], [0, 6, 4, 0]]  
 z = 0  
 mass1 = ((lam[0][1] + lam[0][2]) \* p[0] == lam[1][0] \* p[1] + lam[2][0] \* p[2])  
 mass2 = ((lam[1][0] + lam[1][3]) \* p[1] == lam[0][1] \* p[0] + lam[3][1] \* p[3])  
 mass3 = ((lam[2][0] + lam[2][3]) \* p[2] == lam[0][2] \* p[0] + lam[3][2] \* p[3])  
 mass4 = ((lam[3][1] + lam[3][2]) \* p[3] == lam[1][3] \* p[1] + lam[2][3] \* p[2])  
 mass5 = (p[0] + p[1] + p[2] + p[3] == 1)  
 x\_0\_1 = (0 <= p <= 1)  
 problem = op(z, [mass1, mass2, mass3, mass4, mass5, x\_0\_1])  
 problem.solve(solver='glpk')  
  
 p = [round(i, 2) for i in p.value]  
  
 for i in p:  
 print("new p -", i)  
  
 \_D1 = (p[0] + p[2]) \* 10 + (p[0] + p[1]) \* 6 - (p[1] + p[3]) \* 8 - (p[2] + p[3]) \* 4  
  
 print("\_D1 = ", \_D1)  
  
 if \_D1 > \_D:  
 print(  
 f"экономическая целесообразность ускорения\n ремонтов узлов очевидна\n (\_D1 > \_D примерно на {int(((round(\_D1 / \_D, 2)) \* 100) - 100)}%)")  
 else:  
 print(  
 f"экономическая целесообразность ускорения\n ремонтов узлов очевидна\n (\_D > \_D1 примерно на {int(((round(\_D / \_D1, 2)) \* 100) - 100)}%)")  
  
  
primer3()

Ответ программы:

GLPK Simplex Optimizer 5.0

9 rows, 4 columns, 20 non-zeros

0: obj = 0.000000000e+00 inf = 1.000e+00 (1)

4: obj = 0.000000000e+00 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

[0.4, 0.2, 0.27, 0.13]

\_D = 8.18

GLPK Simplex Optimizer 5.0

9 rows, 4 columns, 20 non-zeros

0: obj = 0.000000000e+00 inf = 1.000e+00 (1)

4: obj = 0.000000000e+00 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

new p - 0.6

new p - 0.15

new p - 0.2

new p - 0.05

\_D1 = 9.9

экономическая целесообразность ускорения

ремонтов узлов очевидна

(\_D1 > \_D примерно на 21%)