

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

<u>доцент</u>	<u></u>	<u>Н. А. Волкова</u>
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Симплекс-метод решения задач

по дисциплине: Прикладные модели оптимизации

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	<u>Z1431</u>	<u></u>	<u>М.Д.Быстров</u>
	номер группы	подпись, дата	инициалы, фамилия

Студенческий билет № 2021/3572

Санкт-Петербург 2025

Оглавление

Оглавление	2
Задание.....	3
Результат выполнения программы	4
Постановка задачи.....	7
Исходный код	8
Вывод.....	13

Задание

1. Найти оптимальное решение задачи линейного программирования с использованием симплекс-метода в Excel и на Python. Привести графическую интерпретацию результатов решения задачи.

2. (Творческое) Придумать ситуацию, для которой подойдет математическая модель (написать содержательную постановку задачи).

№	Задание
4	$x_1 + 3x_2 \rightarrow \max$ $3x_1 + 2x_2 \geq 10$ $-x_1 + 4x_2 \leq 20$ $x_1 + 2x_2 \leq 16$ $-x_1 + 3x_2 \geq 4$ $x_1, x_2 \geq 0$

Результат выполнения программы

Работа выполнена в виде программы на языке программирования Python.

Ниже представлены шаги работы программы при решении заданного варианта.

При описании состава базиса используются следующие обозначения:

0 – x_1 ;

1 – x_2 ;

2 – s_1 ;

3 – s_2 ;

4 – s_3 ;

5 – s_4 .

Шаг 1

Симплекс-таблица

[[-1. -3. 0. 0. 0. 0. 0.]

[3. 2. 1. 0. 0. 0. 10.]

[-1. 4. 0. 1. 0. 0. 20.]

[1. 2. 0. 0. 1. 0. 16.]

[-1. 3. 0. 0. 0. 1. 4.]]

Базисные переменные

[2, 3, 4, 5]

Небазисные переменные

{0, 1}

Вводимая переменная: 1

Исключаемая переменная: 5, значение 1.3333333333333333

Новый базис:[2, 3, 4, 1]

Ведущий столбец: 1

Ведущая строка: 4

Ведущий элемент: 3.0

Новая ведущая строка: [-0.33 1. 0. 0. 0. 0.33 1.33]

Шаг 2

Симплекс-таблица

$\begin{bmatrix} -2. & 0. & 0. & 0. & 0. & 1. & 4. \\ 3.67 & 0. & 1. & 0. & 0. & -0.67 & 7.33 \\ 0.33 & 0. & 0. & 1. & 0. & -1.33 & 14.67 \\ 1.67 & 0. & 0. & 0. & 1. & -0.67 & 13.33 \\ -0.33 & 1. & 0. & 0. & 0. & 0.33 & 1.33 \end{bmatrix}$

Базисные переменные

[2, 3, 4, 1]

Небазисные переменные

{0, 5}

Вводимая переменная: 0

Исключаемая переменная: 2, значение 2.000000000000000004

Новый базис:[0, 3, 4, 1]

Ведущий столбец: 0

Ведущая строка: 1

Ведущий элемент: 3.6666666666666665

Новая ведущая строка: [1. 0. 0.27 0. 0. -0.18 2.]

Шаг 3

Симплекс-таблица

$\begin{bmatrix} 0. & 0. & 0.55 & 0. & 0. & 0.64 & 8. \\ 1. & 0. & 0.27 & 0. & 0. & -0.18 & 2. \\ 0. & 0. & -0.09 & 1. & 0. & -1.27 & 14. \\ 0. & 0. & -0.45 & 0. & 1. & -0.36 & 10. \\ 0. & 1. & 0.09 & 0. & 0. & 0.27 & 2. \end{bmatrix}$

Базисные переменные

[0, 3, 4, 1]

Небазисные переменные

{2, 5}

Не определено вводимой переменной - завершение

Произведено шагов: 2

Итоговая таблица:

```
[[ 0.  0.  0.55 0.  0.  0.64 8. ]
[ 1.  0.  0.27 0.  0. -0.18 2. ]
[ 0.  0. -0.09 1.  0. -1.27 14. ]
[ 0.  0. -0.45 0.  1. -0.36 10. ]
[ 0.  1.  0.09 0.  0.  0.27 2. ]]
```

В результате работы определены базисные переменные (базисное решение):
 $\{x_1, s_2, s_3, x_2\}$.

Наилучшее значение функции $x_1 + 3 \cdot x_2 = 8$.

Графическое отображение решения:

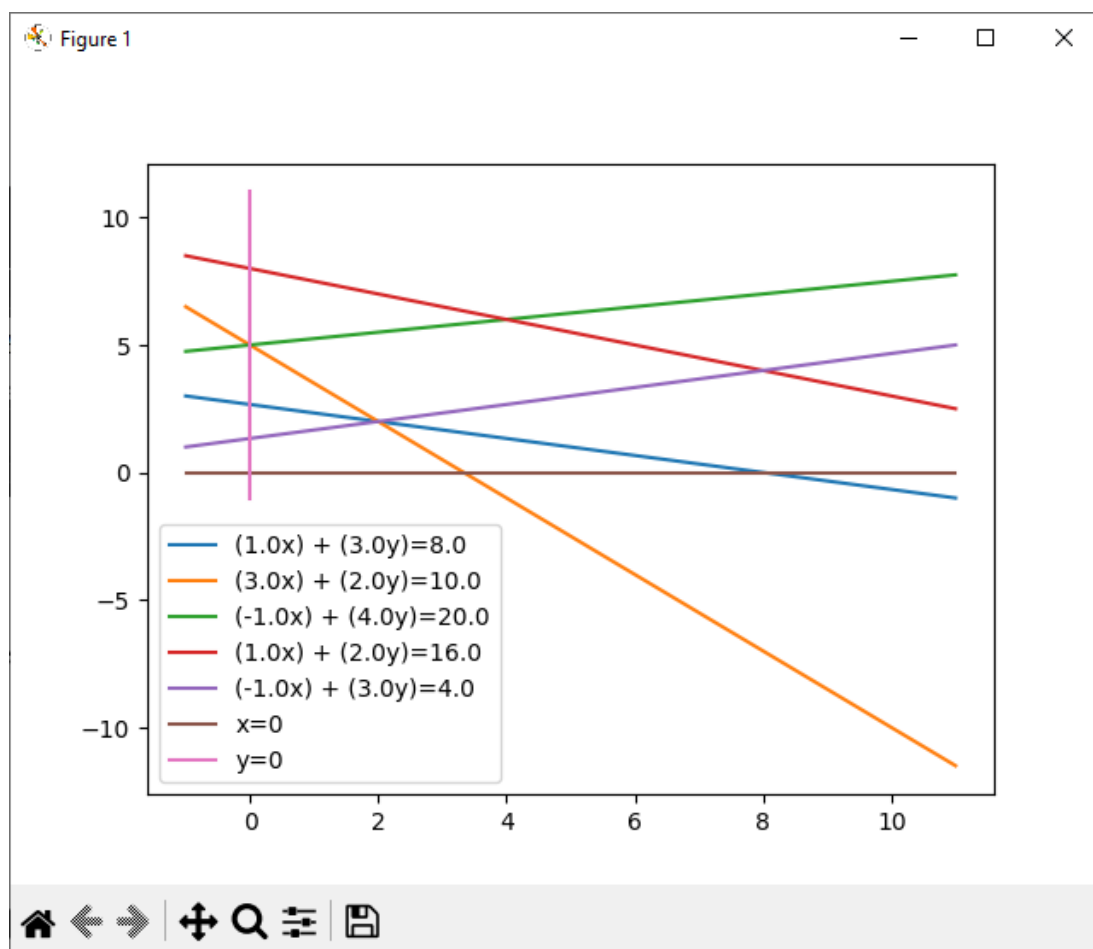


Рисунок 1 Графическое решение

На рисунке 1 видно, что с подобранным $z=8$ функция $z=x_1+3 \cdot x_2$ удовлетворяет установленным ограничениям.

Постановка задачи

Предприятие производит 2 вида продукции: X1, X2, используя сырьё четырех типов. Известны затраты и доходы сырья каждого типа на единицу продукции, запасы сырья на планируемый период, а также прибыль от единицы продукции каждого вида.

Каждый из видов ресурсов может как использоваться при производстве товара, так и быть результатом этого производства – т.е. являться побочным продуктом производства. Некоторые из побочных продуктов запланированы для реализации.

Прибыль при реализации продукции вида 1 составляет 1 у.е., при реализации продукции вида 2 составляет 3 у.е.

Сырье 1 является побочным продуктом производства для всех видов продукции.

Сырье 2 является побочным продуктом производства для продукции вида 1, и используется в производстве продукции 2.

Сырье 3 используется при производстве всех типов продукции.

Сырье 4 является побочным продуктом производства для продукции вида 2, и используется в производстве продукции 1.

Сырье	Продукция 1 Затраты на единицу	Продукция 2 Затраты на единицу	Ресурсы план	Реализация план
1	-3	-2		10
2	-1	4	20	
3	1	2	16	
4	1	-3		4

Сколько изделий каждого вида необходимо произвести, чтобы получить максимум прибыли?

Исходный код

Файл "lab1.py"

```
import numpy as np
import matplotlib.pyplot as plt

# начальный базис
def start_basis(a, number_of_main):

    # размерность базиса
    basis_len = a.shape[0];

    basis = [];

    for i in range(number_of_main, number_of_main + basis_len - 1):
        basis.append(i);

    return set(basis);

def simplex(a, number_of_main):

    # базис
    basis_list = list(start_basis(a, number_of_main));
    basis_size = len(basis_list);

    width = a.shape[1];
    height = a.shape[0];

    print("Размер базиса " + str(len(basis_list)));

    cnt = 1;

    # цикл решения
    while (1):

        print("Шаг " + str(cnt));

        print("Симплекс-таблица");
        print(np.round(a, 2));

        print("Базисные переменные");
        print(basis_list);

        # небазисные переменные
        no_basis_set = set(range(0, number_of_main + basis_size));
        no_basis_set = no_basis_set - set(basis_list);

        print("Небазисные переменные");
```



```

print(no_basis_set);

# определяем какую переменную делаем базисной (вводимой) -
# имеющей наибольший отрицательный коэффициент в z-строке
z = a[0];

next_basis = -1;
min = 1;

for no_basis in no_basis_set:
    if (z[no_basis] < 0
        and z[no_basis] < min):
        next_basis = no_basis;
        min = z[no_basis];

# нет больше кандидатов в базис - завершаем выполнение
if next_basis == -1:
    print("Не определено вводимой переменной - завершение");
    break;

print("Вводимая переменная: " + str(next_basis));

next_no_basis = -1;
min = -1;

# определяем исключаемый элемент
for row in range(1, height):
    if (a[row][next_basis] > 0):

        # решение строки делим на элемент в столбце
        # вводимой переменной
        rel = a[row][width - 1] / a[row][next_basis];

        if rel > 0:
            if rel < min or min == -1:
                next_no_basis = basis_list[row - 1];
                min = rel;

# если нет кандидатов на выход из базиса
if next_no_basis == -1:
    print("Не определено исключаемой переменной - завершение")
    break;

print("Исключаемая переменная: " + str(next_no_basis) + ", значение " +
str(min));

swap_index = basis_list.index(next_no_basis);
basis_list[swap_index] = next_basis;

print("Новый базис:" + str(basis_list));

```

```

# далее расчет нового базисного решения
# Для этого определим ведущий столбец, ассоциируемый с
# вводимой переменной, и ведущую строку,
# ассоциируемую с исключаемой переменной.
# Элемент, находящийся на пересечении ведущего столбца
# и ведущей строки называется ведущим элементом

# ведущий столбец
main_column = next_basis;

# ведущая строка
main_row = swap_index + 1; # первая строка - z, поэтому смещение

# ведущий элемент
main_element = a[main_row][main_column]; # ведущий элемент

print("Ведущий столбец: " + str(main_column));
print("Ведущая строка: " + str(main_row));
print("Ведущий элемент: " + str(main_element));

#1. Вычисление элементов новой ведущей строки:
for column in range(width):

    # Новая ведущая строка = Текущая ведущая строка / Ведущий элемент.
    a[main_row][column] = a[main_row][column] / main_element;

print("Новая ведущая строка: " + str(np.round(a[main_row], 2)));

#2. Вычисление элементов остальных строк, включая z-строку:
for row in set(range(height)) - set([main_row]): # по всем строкам кроме
главной

    # коэффициент строки в ведущем столбце
    row_main_coeff = a[row][main_column];

    for column in range(width):

        # Новая строка = Текущая строка -
        # ее коэффициент в ведущем столбце * новая ведущая строка.
        a[row][column] = (a[row][column] -
            row_main_coeff * a[main_row][column]);

    cnt += 1;

#break;

print("Произведено шагов: " + str(cnt - 1));
print("Итоговая таблица:");

```

```

print(np.round(a, 2));

# искомый коэффициент z
z = a[0][width - 1];

return [z, a];

# вариант №4
#  $x_1 + 3x_2 \rightarrow \max$ 
#  $3x_1 + 2x_2 \geq 10$ 
#  $-x_1 + 4x_2 \leq 20$ 
#  $x_1 + 2x_2 \leq 16$ 
#  $-x_1 + 3x_2 \geq 4$ 
#  $x_1, x_2 \geq 0$ 

# кол-во основных переменных
main_variables = 2;

# матрица коэффициентов z, x, s переменных
# начальное условие
A = np.array(
    [
        [-1, -3, 0, 0, 0, 0, 0], #z
        [ 3,  2, 1, 0, 0, 0, 10], #s1
        [-1,  4, 0, 1, 0, 0, 20], #s2
        [ 1,  2, 0, 0, 1, 0, 16], #s3
        [-1,  3, 0, 0, 0, 1,  4] #s4
    ],
    dtype = float);

solve = simplex(A.copy(), main_variables);

#print(A);
print(solve);

# границы для отрисовки графиков - указаны вручную
x_min = 0;
x_max = 10;
y_min = 0;
y_max = 10;

plot_dimension = 100;

# координаты для отрисовки графиков
xdata = np.linspace(x_min - 1, x_max + 1, plot_dimension);
ydata = np.linspace(y_min - 1, y_max + 1, plot_dimension);

fig, ax = plt.subplots();

for row in range(A.shape[0]):

```

```

x = A[row][0];
y = A[row][1];
z = A[row][A.shape[1] - 1];
col = '';

if (row == 0):
    x = -x;
    y = -y;
    z = solve[1][0][A.shape[1] - 1];
    col = '';

if (y != 0):
    y_dat = (z - xdata * x);
    y_dat = y_dat / y;
    x_dat = xdata;
elif (x != 0):
    x_dat = np.linspace(z/x, z/x, plot_dimension);
    y_dat = ydata;

plot, = ax.plot(x_dat, y_dat, col);
plot.set_label("(" + str(x) + "x) + (" + str(y) + "y)=" + str(z));

# графики ограничений x,y > 0
plot, = ax.plot(xdata, np.linspace(0, 0, plot_dimension));
plot.set_label("x=0");
plot, = ax.plot(np.linspace(0,0,plot_dimension), ydata);
plot.set_label("y=0");

ax.legend();
plt.show();

```

Вывод

В ходе выполнения первой лабораторной работы создана программа на языке Python для решения задачи максимизации функции с помощью симплекс-метода.

Дана содержательная (описательная) постановка задачи для имеющейся математической модели.