

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«Гомельский государственный технический университет
имени П. О. Сухого»

Кафедра «Информатика»

Лабораторная работа № 1
по дисциплине: **«Системный анализ и исследование операций»**

Выполнил студент
группы ИП-41
Бурцев В. В.

Проверил преподаватель
Бородин Н.Н.

ЛАБОРАТОРНАЯ РАБОТА №1

Программная реализация решения задач целочисленного программирования

Цель работы: получить практические навыки решения задач целочисленного программирования.

Задание

- 1) Решить задачи целочисленного программирования (варианты см. в таблице 1.1);
- 2) Решить (см. общую постановку ЗЦП и таблицу вариантов ниже):
 - а) частично целочисленную задачу (x_1 – любое, x_2 – целочисленное);
 - б) полностью целочисленную задачу (x_1 и x_2 – целочисленные)средствами пакетов Mathcad (графическим методом и с помощью блока решения) и MS Excel (с помощью надстройки «Поиск решения»). Сравнить полученные результаты.

$$z(x_1, x_2) = ax_1 + bx_2 \rightarrow \max$$

$$cx_1 + dx_2 \leq e$$

$$fx_1 + gx_2 \leq h$$

$$px_1 + rx_2 \geq q$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Вариант 1

Вариант	a	b	c	d	e	f	g	h	p	r	q
1	6	5	7	-8	3	-7	6	5	2	3	1

$$z(x_1, x_2) = 6x_1 + 5x_2 \rightarrow \max$$

$$7x_1 - 8x_2 \leq 3$$

$$-7x_1 + 6x_2 \leq 5$$

$$2x_1 + 3x_2 \leq 15$$

$$x_1 \geq 0, x_2 \geq 0$$

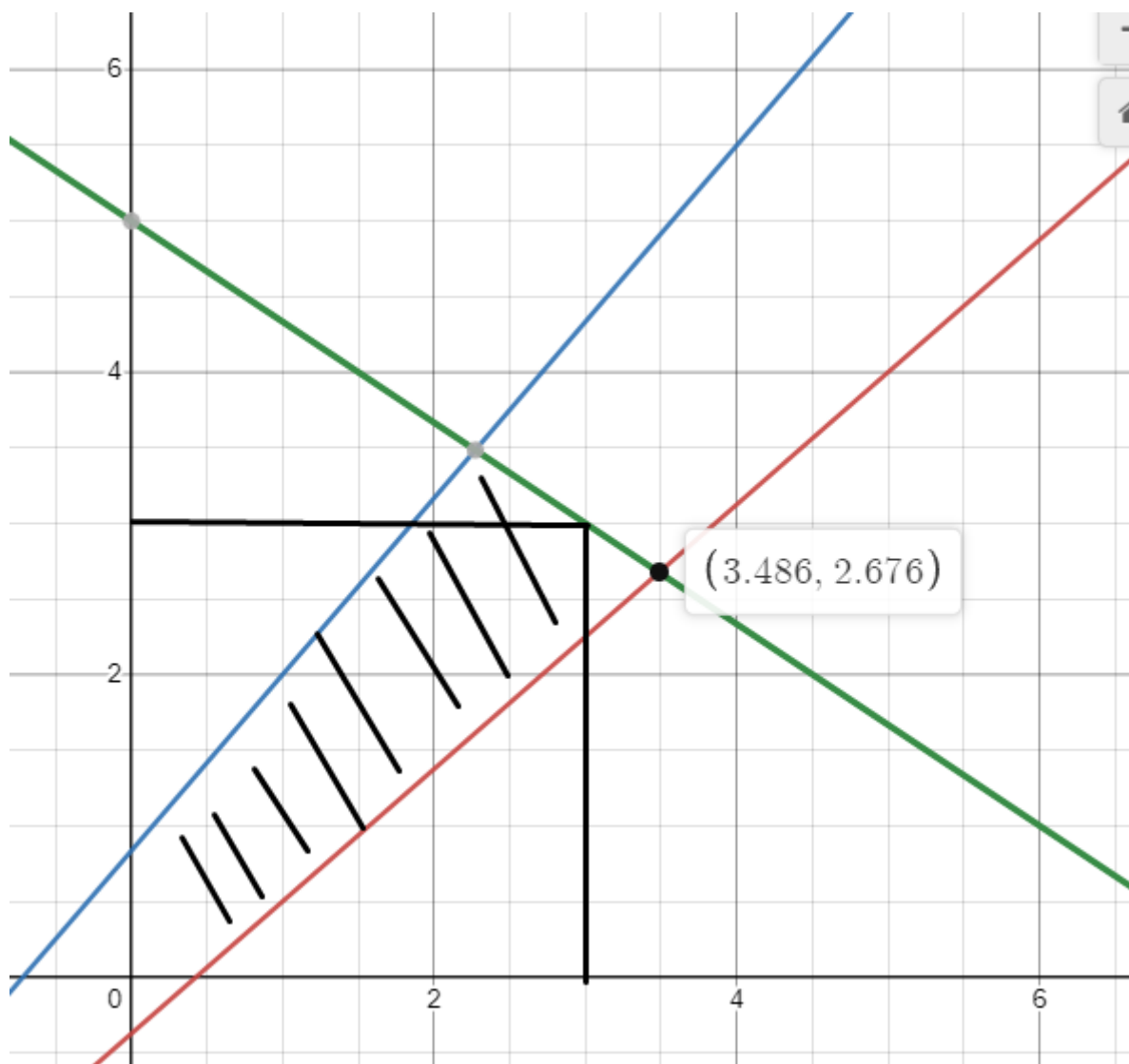


Рисунок 1 – График

Из графика видно, что x_1, x_2 будут равны 3, 3. Значит $F = 6 * 3 + 5 * 3 = 33$

Программная реализация:

```
import numpy as np
import matplotlib.pyplot as plt
from pulp import LpMaximize, LpProblem, LpStatus, lpSum, LpVariable

def show_point(model) -> None:
    x = [data_point.value() for data_point in model.variables()]
    print(f'x1 = {x[0]}\n' + f'x2 = {x[1]}')
    plt.scatter(x[0], x[1], color="red")

def show_info(model) -> None:
    print(f'status: {model.status}, {LpStatus[model.status]}')
    print(f'objective: {model.objective.value()}')
    for name, constraint in model.constraints.items():
        print(f'{name}: {constraint.value()}')

def create_model(*args: tuple):
    a, b, c, d, e, f, g, h, p, r, q = args
    model = LpProblem(name="small-problem", sense=LpMaximize)
    x_1 = LpVariable(name="x1", lowBound=0)
    x_2 = LpVariable(name="x2", lowBound=0, cat="Integer")
    model += (c * x_1 + d * x_2 <= e, "first_constraint")
    model += (f * x_1 + g * x_2 <= h, "second_constraint")
    model += (p * x_1 + r * x_2 <= q, "third_constraint")
    model += lpSum([a * x_1, b * x_2])
    status = model.solve()
    return model

def create_landscape(*args):
    a, b, c, d, e, f, g, h, p, r, q, x = args
    y2 = (e - c * x) / d
    y3 = (h - f * x) / g
    y4 = (q - p * x) / r
    y = [y2, y3, y4]
    for y_variable in y:
        plt.plot(x, y_variable)
        plt.plot(x, y_variable)
        plt.plot(x, y_variable)

    return y

def work(*args: tuple):
    a, b, c, d, e, f, g, h, p, r, q = args
    x = np.linspace(0, a, 2000)

    y = create_landscape(*args, x)
    plt.xlim((0, a))
    plt.xlim((0, a))
    y5 = np.minimum(y[0], y[2])
    y6 = y[1]
    plt.fill_between(x, y5, y6, where=y5 > y6, color='grey', alpha=0.5)
    model = create_model(*args)
    show_point(model)
    show_info(model)

    plt.grid()
    plt.show()
```

Результат:

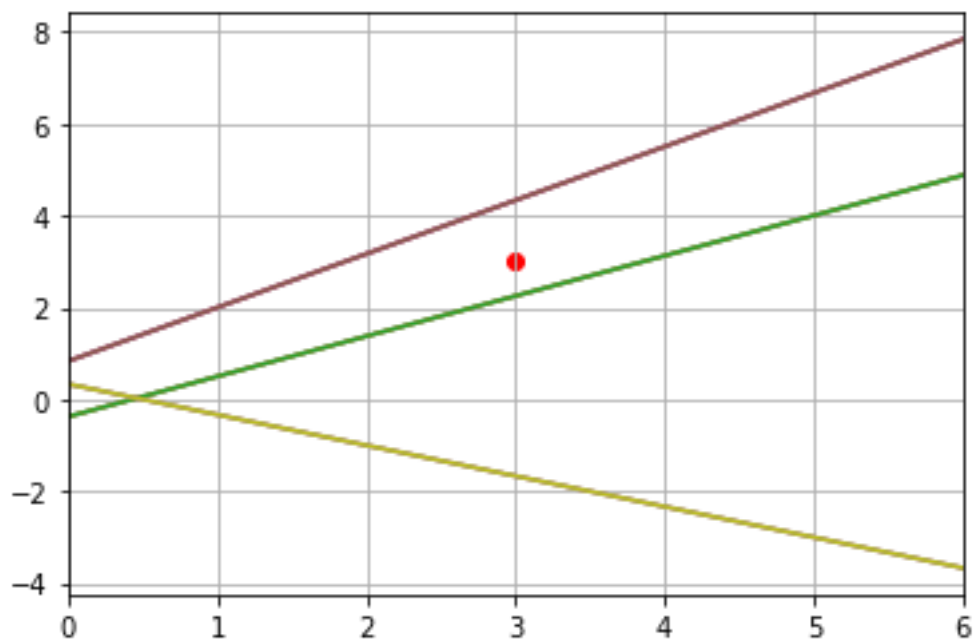


Рисунок 2 – Результат программы

```
x1 = 3.0  
x2 = 3.0  
status: 1, Optimal  
objective: 33.0
```

Рисунок 3 – Результат программы

Вывод: были получены практические навыки решения задач целочисленного программирования.