Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ***ИУК «Информатика и управление»***

КАФЕДРА ***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

ЛАБОРАТОРНАЯ РАБОТА №4

«Задачи целочисленного линейного программирования»

ДИСЦИПЛИНА: «Моделирование»

Выполнил: студент гр. ИУК4-72Б ________________ (Карельский М.К.)
(Подпись)

Проверил: ________________ (Никитенко У.В.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):
- Балльная оценка:
- Оценка:

Калуга, 2023

Цель: сформировать практические навыки анализа возможностей построения и выделения наиболее важных свойств объектов моделей для моделирования и использования специализированных программных пакетов и библиотек для стандартных вычислений при решении задач целочисленного линейного программирования на основе сравнения результатов.

Задачи: применить методы отсечений и комбинаторные методы к задаче целочисленного программирования, указанной в варианте, сравнить результаты, выдвинуть и обосновать гипотезу целесообразности использования того или иного подхода в зависимости от предложенной задачи и ее вариаций, точности результата, трудоемкости, сложности алгоритма, сложности обоснования применимости метода, вычислительной эффективности алгоритма.

Вариант 7

Найдите оптимальный план задачи целочисленного линейного программирования, используя

- первый алгоритм Гомори;
- второй алгоритм Гомори (x1 – произвольное, x2 – целое);
- метод ветвей и границ (решение проиллюстрируйте схемой).

$$z = 3x_1 + 4x_2 \rightarrow min$$
$$5x_1 + 2x_2 \geq 12$$
$$2x_1 + 5x_2 \geq 14$$
$$x_1, x_2 \geq 0$$
$$x_1, x_2 - \text{целые}$$

Решение:

Первый алгоритм Гомори:

Канонический вид:

$$z = 3x_1 + 4x_2 \rightarrow min$$
$$5x_1 + 2x_2 - x_3 = 12$$
$$2x_1 + 5x_2 - x_4 = 14$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Симплекс-таблица:

Базис	x_1	x_2	x_3	x_4	План
x_1	1	0	-5/21	2/21	32/21
x_2	0	1	2/21	-5/21	46/21
z	0	0	-1/3	-2/3	40/3

Решение в произвольных числах:

$$x_1 = \frac{32}{21} \approx 1.52$$
$$x_2 = \frac{46}{21} \approx 2.19$$
$$z = \frac{40}{3} \approx 13.33$$

Целые части:

$$[x_1] = 1$$
$$[x_2] = 2$$

Дробные части:

$$\{x_1\} = \frac{11}{21} - \text{наибольшая}$$
$$\{x_2\} = \frac{4}{21}$$

Дополнительные ограничения целочисленности:

$$q_1 - q_{11}x_1 - q_{12}x_2 - q_{13}x_3 - q_{14}x_4 \leq 0$$
$$q_1 = 32/21 - 1 = 11/21$$
$$q_{11} = 1 - 1 = 0$$
$$q_{12} = 0 - 0 = 0$$
$$q_{13} = -5/21 + 1 = 16/21$$
$$q_{14} = 2/21 - 0 = 2/21$$
$$11/21 - 16/21x_3 - 2/21x_4 \leq 0$$
$$-16/21x_3 - 2/21x_4 + x_5 = -11/21$$

Добавление строки:

Базис	x_1	x_2	x_3	x_4	x_5	План
x_1	1	0	-5/21	2/21	0	32/21
x_2	0	1	2/21	-5/21	0	46/21
x_5	0	0	-16/21	-2/21	1	-11/21
z	0	0	-1/3	-2/3	0	40/3

Преобразование симплекс-таблицы:

Базис	x_1	x_2	x_3	x_4	x_5	План
x_1	1	0	0	1/8	-5/16	27/16
x_2	0	1	0	-1/4	1/8	17/8
x_3	0	0	1	1/8	-21/16	11/16
z	0	0	0	-5/8	-7/16	-167/16

Дробные части:

$$\{x_1\} = \frac{11}{16}$$
$$\{x_2\} = \frac{1}{8} - \text{наибольшая}$$
$$\{x_3\} = \frac{11}{16}$$

Дополнительные ограничения целочисленности:

$$q_1 - q_{11}x_1 - q_{12}x_2 - q_{13}x_3 - q_{14}x_4 - q_{15}x_5 \leq 0$$
$$q_1 = 27/16 - 1 = 11/16$$
$$q_{11} = 1 - 1 = 0$$
$$q_{12} = 0 - 0 = 0$$
$$q\_13 = 0 - 0 = 0$$
$$q_{14} = 1/8 - 0 = 1/8$$
$$q_{15} = -5/16 + 1 = 11/16$$
$$11/16 - 1/8x_4 - 11/16x_5 \leq 0$$
$$-1/8x_4 - 11/16x_5 + x_6 = -11/16$$

Добавление строки:

Базис	x_1	x_2	x_3	x_4	x_5	x_6	План
x_1	1	0	0	1/8	-5/16	0	27/16
x_2	0	1	0	-1/4	1/8	0	17/8
x_3	0	0	1	1/8	-21/16	0	11/16
x_6	0	0	0	-1/8	-11/16	1	-11/16
z	0	0	0	-5/8	-7/16	0	-167/16

Преобразование симплекс-таблицы:

Базис	x_1	x_2	x_3	x_4	x_5	x_6	План
x_1	1	0	0	2/11	0	-5/11	2
x_2	0	1	0	-3/11	0	2/11	2
x_3	0	0	1	4/11	0	-21/11	2
x_5	0	0	0	2/11	1	-16/11	1
z	0	0	0	-6/11	0	-7/11	-14

Оптимальный целочисленный план:

$$x_1 = 2$$
$$x_2 = 2$$
$$z = 14$$

Второй алгоритм Гомори:

Канонический вид:

$$z = 3x_1 + 4x_2 \to min$$
$$5x_1 + 2x_2 - x_3 = 12$$
$$2x_1 + 5x_2 - x_4 = 14$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Симплекс-таблица:

Базис	x_1	x_2	x_3	x_4	План
x_1	1	0	-5/21	2/21	32/21
x_2	0	1	2/21	-5/21	46/21
z	0	0	-1/3	-2/3	40/3

x_2 имеет дробную часть. Дополнительное ограничение:

$$\frac{2}{21}x_3 + \frac{\frac{46}{21}}{\frac{46}{21} - 1}\left(\frac{-5}{21}\right)x_4 \leq \frac{46}{21}$$
$$-2/21x_3 - 20/357x_4 + x_5 = -4/21$$

Добавление строки:

Базис	x_1	x_2	x_3	x_4	x_5	План
x_1	1	0	-5/21	2/21	0	32/21
x_2	0	1	2/21	-5/21	0	46/21
x_5	0	0	-2/21	-20/357	1	-4/21
z	0	0	-1/3	-2/3	0	40/3

Преобразование симплекс-таблицы:

Базис	x_1	x_2	x_3	x_4	x_5	План
x_1	1	0	0	4/17	-5/2	2
x_2	0	1	0	-5/17	1	2
x_3	0	0	1	10/17	-21/2	2
z	0	0	0	-8/17	-7/2	-14

Оптимальный план:

$$x_1 = 2$$
$$x_2 = 2$$
$$z = 14$$

Метод ветвей и границ:
График системы:

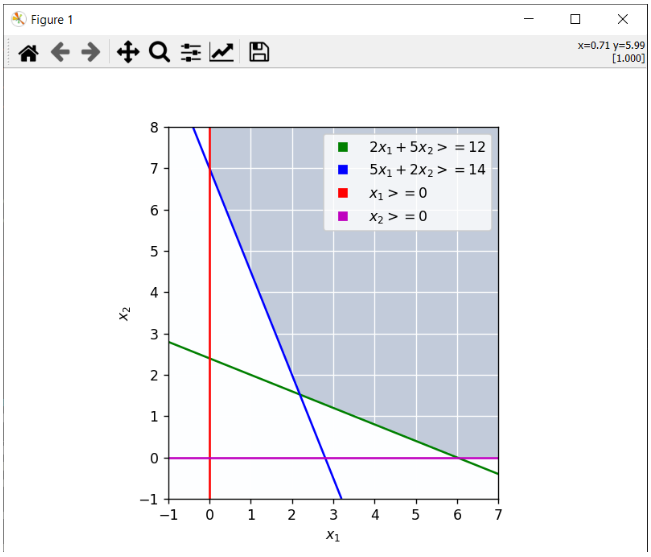

Рис. 1. График системы

Минимум z $= \frac{266}{21} \approx 12.67$ достигается при $x_1 = \frac{46}{21} \approx 2.19, x_2 = \frac{32}{21} \approx 1.52$
Разобьем задачу 1 на подзадачи 11 и 12:

- 11: $x_1 \leq 2$
- 12: $x_1 \geq 3$

Задача 11:

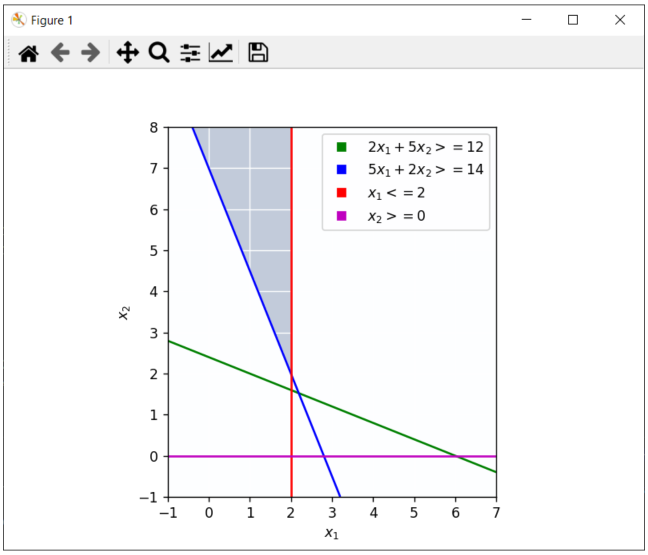

Рис. 2. Задача 11

Решение:

$$x_1 = 2$$
$$x_2 = 2$$
$$\underline{z = 14}$$

Задача 12:

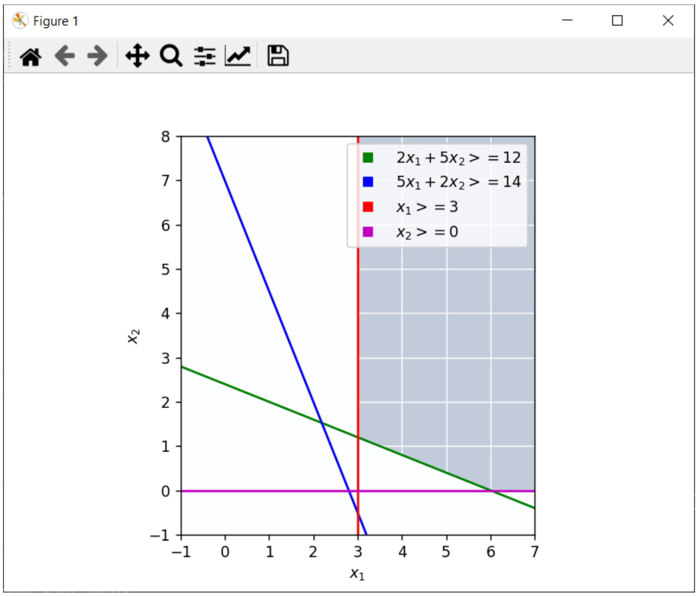

Рис. 3. Задача 12

Решение:

$$x_1 = 3$$
$$x_2 = 1.2$$
$$z = 13.8$$

Разобьем задачу 12 на подзадачи 121 и 122:

- 121: $x_1 \geq 3, x_2 \leq 1$
- 122: $x_1 \geq 3, x_2 \geq 2$

Задача 121:

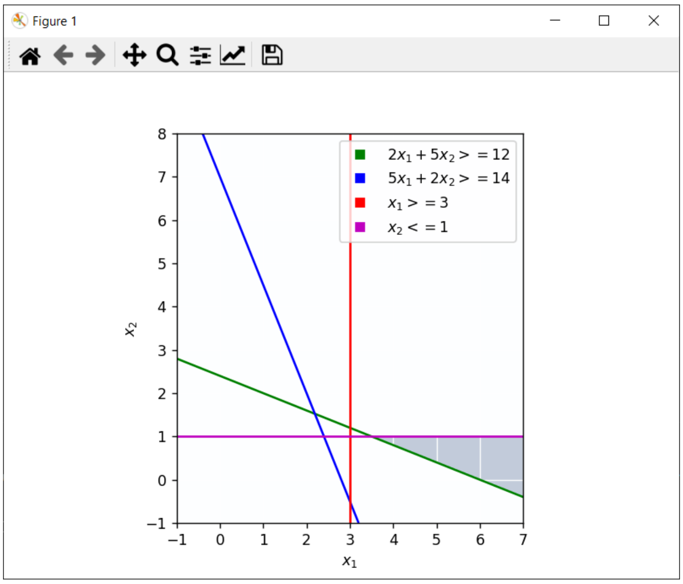

Рис. 4. Задача 121

Решение:

$$x_1 = 3.5$$
$$x_2 = 1$$
$$z = 14.5$$

Разобьем задачу 121 на подзадачи 1211 и 1212:

- 1211: $x_1 = 3, x_2 \leq 1$ – не имеет решения
- 1212: $x_1 \geq 4, x_2 \leq 1$ – дальнейшее ветвление ведет к увеличению z

Задача 122:

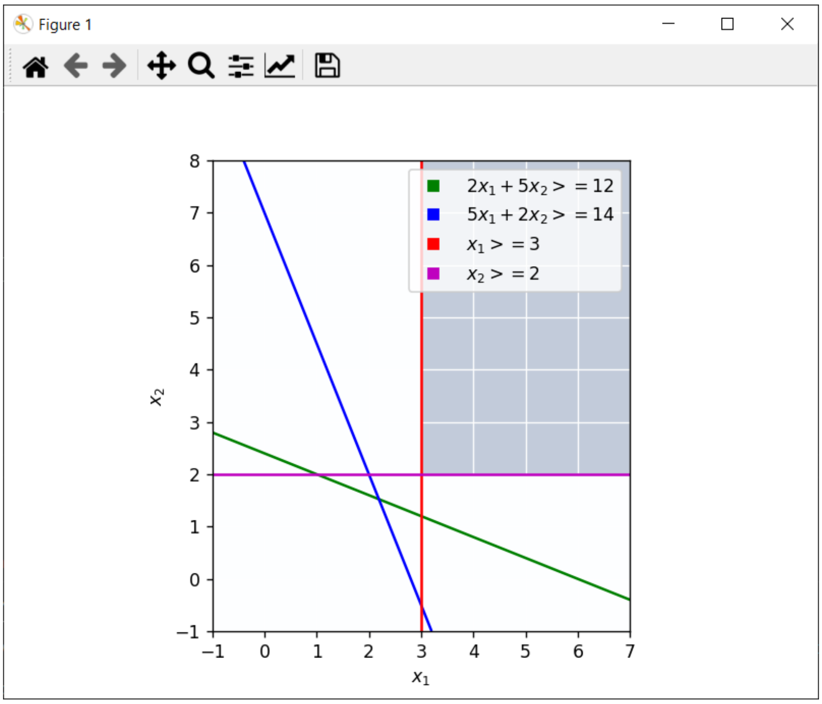

Рис. 5. Задача 122

Решение:

$$x_1 = 3$$
$$x_2 = 2$$
$$z = 17$$

Оптимальный план:

$$x_1 = 2$$
$$x_2 = 2$$
$$z = 14$$

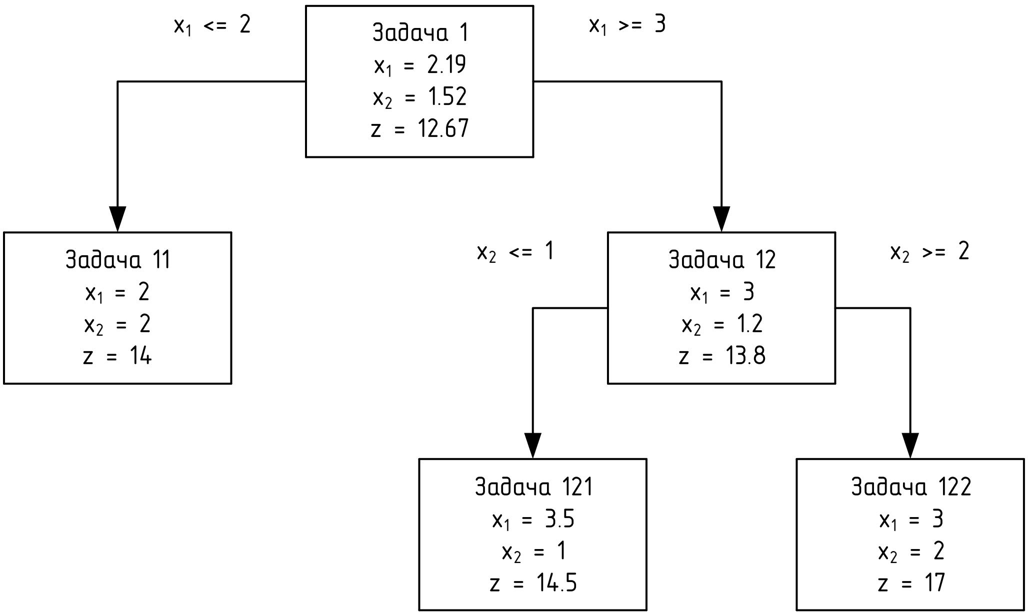

Рис. 6. Метод ветвей и границ

Вывод: в ходе выполнения лабораторной работы были получены практические навыки анализа возможностей построения и выделения наиболее важных свойств объектов моделей для моделирования и использования специализированных программных пакетов и библиотек для стандартных вычислений при решении задач целочисленного линейного программирования на основе сравнения результатов.

ПРИЛОЖЕНИЯ

Листинг:

LW4_1.py:

```
from docplex.mp.model import Model

m = Model()
x_1 = m.integer_var(name='x_1', lb=0)
x_2 = m.integer_var(name='x_2', lb=0)

m.add_constraint(2 * x_1 + 5 * x_2 >= 12)
m.add_constraint(5 * x_1 + 2 * x_2 >= 14)
m.minimize(3 * x_1 + 4 * x_2)

c = m.get_cplex()
c.parameters.simplex.limits.iterations.set(100)
c.parameters.lpmethod.set(c.parameters.lpmethod.values.primal)

while c.solution.get_status() != c.solution.status.optimal:
    c.solve()
    print("=== Симплекс-таблица ===")
    for tableau_row in c.solution.advanced.binvarow():
        print(tableau_row)

m.solve()
print("\n=== Решение задачи, где x_1 - целое, а x_2 - целое ===")
m.print_solution()
```

LW4_2.py:

```
from docplex.mp.model import Model

m = Model()
x_1 = m.continuous_var(name='x_1', lb=0)
x_2 = m.integer_var(name='x_2', lb=0)

m.add_constraint(2 * x_1 + 5 * x_2 >= 12)
m.add_constraint(5 * x_1 + 2 * x_2 >= 14)
m.minimize(3 * x_1 + 4 * x_2)

c = m.get_cplex()
c.parameters.simplex.limits.iterations.set(100)
c.parameters.lpmethod.set(c.parameters.lpmethod.values.primal)

while c.solution.get_status() != c.solution.status.optimal:
    c.solve()
    print("=== Симплекс-таблица ===")
    for tableau_row in c.solution.advanced.binvarow():
        print(tableau_row)

m.solve()
```

```
print("\n=== Решение задачи, где x_1 - произвольное, а x_2 - целое ===")
m.print_solution()
```

LW4_3.py:

```
import itertools
from functools import reduce
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import sympy
from matplotlib.ticker import MultipleLocator

if __name__ == '__main__':
    conditions = [
        lambda x_1, x_2: 2 * x_1 + 5 * x_2 >= 12,
        lambda x_1, x_2: 5 * x_1 + 2 * x_2 >= 14,
        lambda x_1, x_2: x_1 >= 3,
        lambda x_1, x_2: x_2 >= 2
    ]

    equalities = [
        lambda x_1, x_2: 2 * x_1 + 5 * x_2 - 12,
        lambda x_1, x_2: 5 * x_1 + 2 * x_2 - 14,
        lambda x_1, x_2: x_1 - 3,
        lambda x_1, x_2: x_2 - 2
    ]

    labels = [
        '$2 x_1 + 5 x_2 >= 12$',
        '$5 x_1 + 2 x_2 >= 14$',
        '$x_1 >= 3$',
        '$x_2 >= 2$'
    ]

    colors = ['g', 'b', 'r', 'm']

    x_1_bounds = (-1, 7)
    x_2_bounds = (-1, 8)
    x_1_range = np.linspace(x_1_bounds[0], x_1_bounds[1], 250)
    x_2_range = np.linspace(x_2_bounds[0], x_2_bounds[1], 250)
    x_1s, x_2s = np.meshgrid(x_1_range, x_2_range)

    axis: plt.Axes
    figure, axis = plt.subplots()
    axis.set_xlim(*x_1_bounds)
    axis.set_ylim(*x_2_bounds)

    handles = []
    for equality in equalities:
        axis.contour(
            x_1s, x_2s, equality(x_1s, x_2s), [0],
```

```python
            colors=colors[equalities.index(equality)]
        )
        handles.append(
            matplotlib.lines.Line2D(
                [], [], color=colors[equalities.index(equality)],
                marker="s", ls="",
                label=labels[equalities.index(equality)]
            )
        )

    regions = [condition(x_1s, x_2s) for condition in conditions]
    intersection = np.array(reduce(lambda _x, _y: _x & _y, regions))

    extent = (x_1s.min(), x_1s.max(), x_2s.min(), x_2s.max())
    plt.imshow(
        intersection.astype(int),
        extent=extent,
        origin="lower",
        cmap="Blues",
        alpha=0.25
    )

    plt.xlabel("$x_1$")
    plt.ylabel("$x_2$")

    axis.xaxis.set_major_locator(MultipleLocator(1))
    axis.yaxis.set_major_locator(MultipleLocator(1))

    axis.grid(color='w', linestyle='-')

    plt.legend(handles=handles)
    plt.show()

    sym_x_1 = sympy.Symbol('x_1')
    sym_x_2 = sympy.Symbol('x_2')
    for equality_1, equality_2 in list(itertools.combinations(equalities, 2)):

        solution = sympy.solve(
            [
                equality_1(sym_x_1, sym_x_2), equality_2(sym_x_1, sym_x_2)
            ],
            [sym_x_1, sym_x_2], particular=True
        )

        x_1 = solution[sym_x_1]
        x_2 = solution[sym_x_2]

        if all(ineq(x_1, x_2) for ineq in conditions):
            print('Пересечение графиков ', end='')
            print(labels[equalities.index(equality_1)], end=' и ')
            print(labels[equalities.index(equality_2)])
            print(solution)
```