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 №2

«Графический метод решения задачи математического 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: изучение математического аппарата математического программирования на примере задач небольшой размерности, допускающих графическое решение.

Задачи: представить графическое решение, реализованное на языке высокого уровня.

Вариант 7

Задание 1.
Решить задачу нелинейного программирования графическим методом:

$$z = x_1 + x_2 \to (max, min)$$

$$\begin{cases} x_1 x_2 \geq 1 \\ x_1^2 + x_2^2 \leq 9 \end{cases}$$

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

Решение:

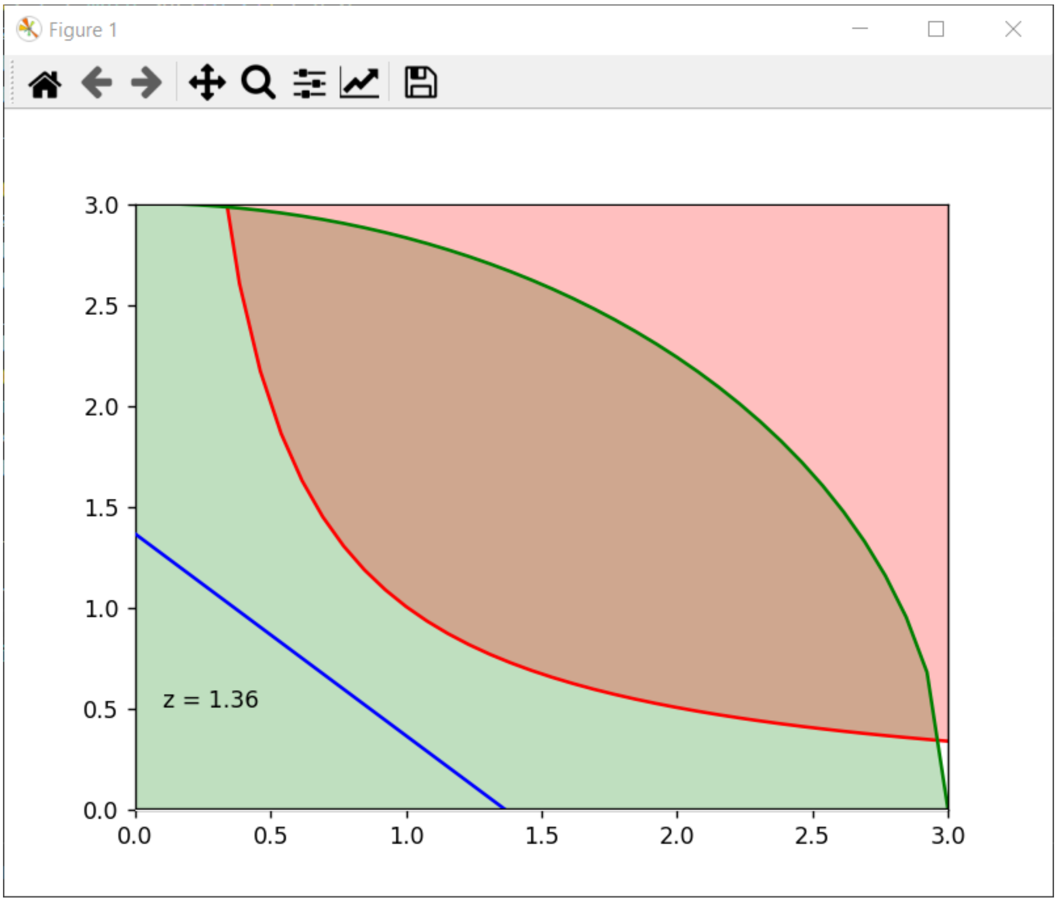

Рис. 1.1. Решение

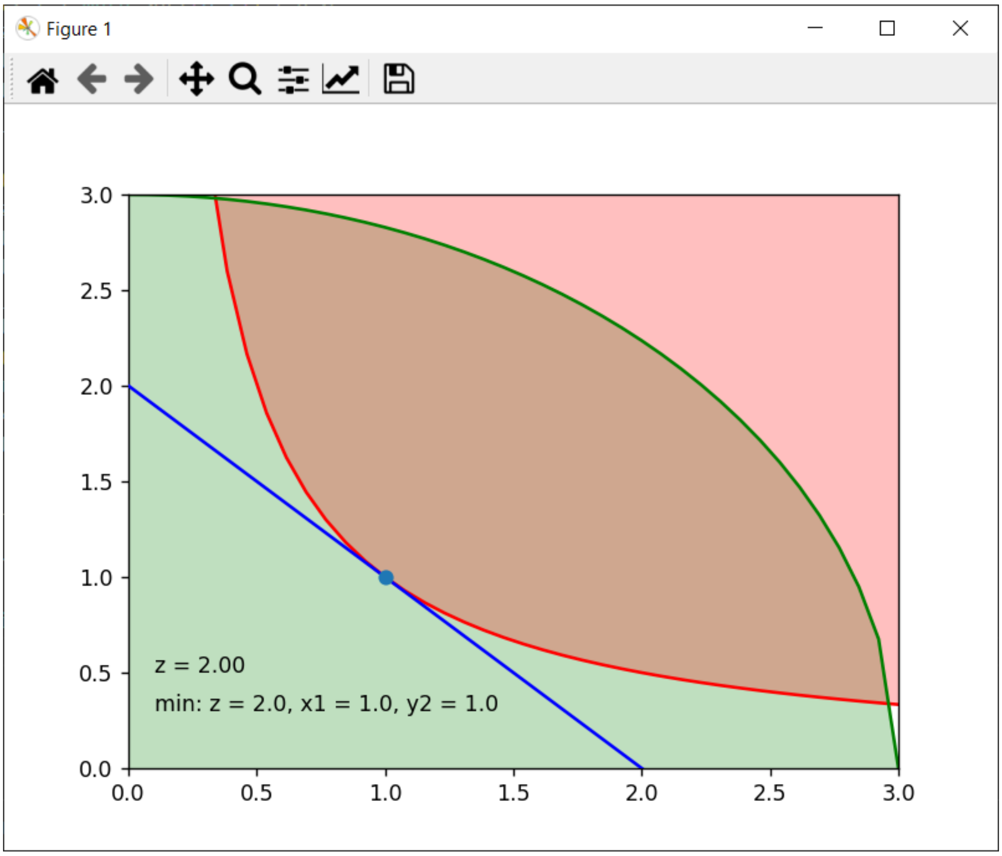

Рис. 1.2. Решение

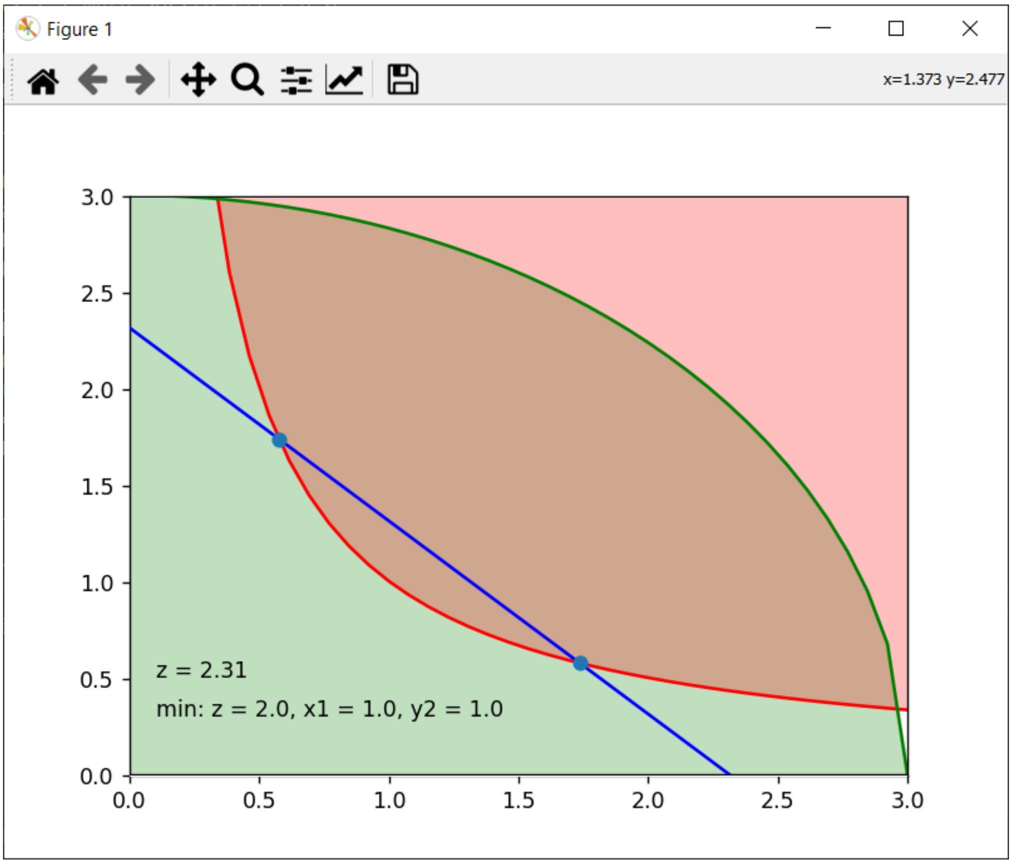

Рис. 1.3. Решение

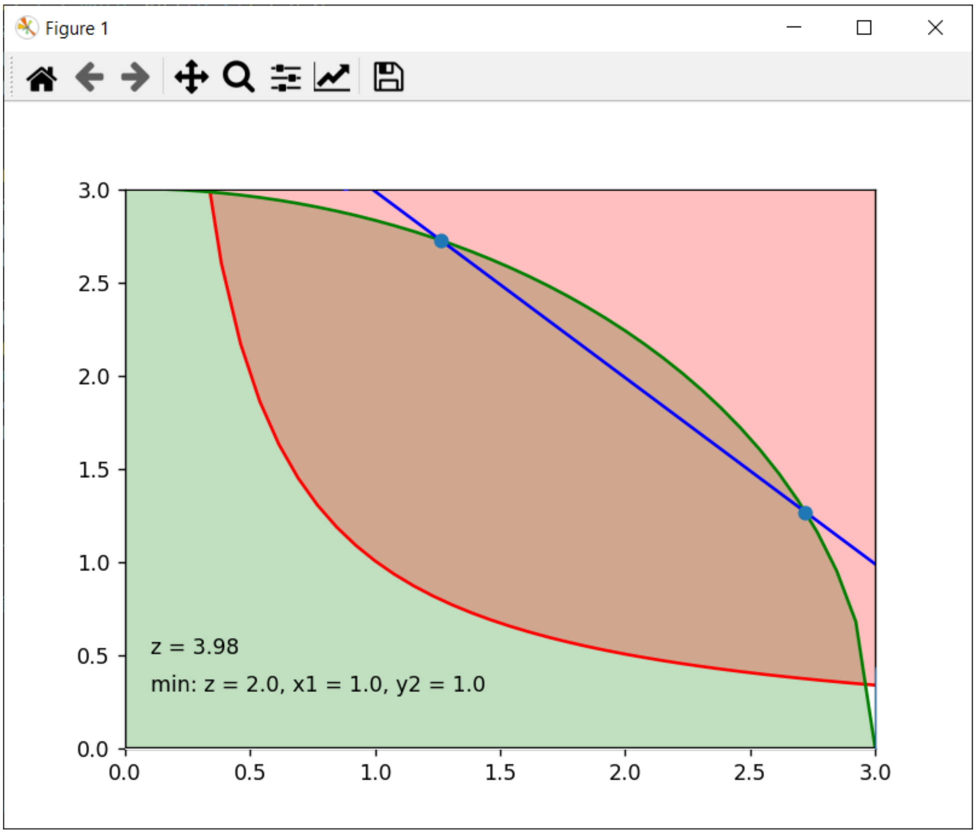

Рис. 1.4. Решение

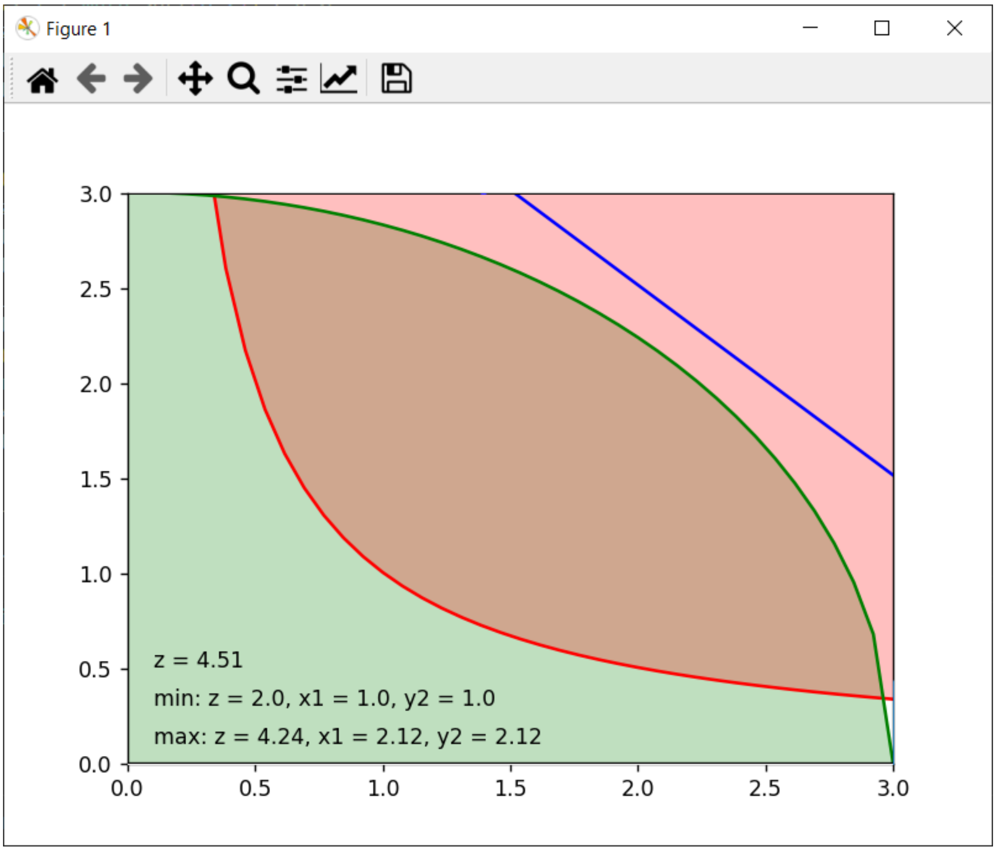

Рис. 1.5. Решение

Задание 2.
Найти условный экстремум функции методом множителей Лагранжа:

$$z = 4x_1 + 9x_2 - 25 \to extr$$
$$\text{при условии } 4x_1^2 + 36x_2^2 = 9$$

Решение:

$$L = f(x_1, x_2) + \lambda\varphi(x_1, x_2)$$
$$\varphi(x_1, x_2) = 4x_1^2 + 36x_2^2 - 9 = 0$$
$$L = 4x_1 + 9x_2 - 25 + \lambda(4x_1^2 + 36x_2^2 - 9)$$

$$L'_{x_1} = 4 + 8\lambda x_1$$
$$L'_{x_2} = 9 + 72\lambda x_2$$

$$\begin{cases} L'_{x_1} = 0 \\ L'_{x_2} = 0 \\ \varphi(x_1, x_2) = 0 \end{cases} \to \begin{cases} 4 + 8\lambda x_1 = 0 \\ 9 + 72\lambda x_2 = 0 \\ 4x_1^2 + 36x_2^2 - 9 = 0 \end{cases}$$

$$x_1 = -\frac{1}{2\lambda}$$
$$x_2 = -\frac{1}{8\lambda}$$

$$\frac{1}{\lambda^2} + \frac{9}{16\lambda^2} - 9 = 0$$
$$\frac{25}{16\lambda^2} = 9$$
$$\lambda^2 = \frac{25}{144}$$

$$\lambda = \frac{5}{12}$$
$$x_1 = -1.2$$
$$x_2 = -0.3$$
$$z = -32.5$$

$$\lambda = -\frac{5}{12}$$
$$x_1 = 1.2$$
$$x_2 = 0.3$$
$$z = -17.5$$

$$L''_{x_1x_1} = 8\lambda$$
$$L''_{x_1x_2} = 0$$
$$L''_{x_2x_2} = 72\lambda$$

$$d^2L = L''_{x_1x_1}(dx)^2 + 2L''_{x_1x_2}dxdy + L''_{x_2x_2}(dy)^2$$
$$d^2L = 8\lambda(dx)^2 + 72\lambda(dy)^2$$

$$\lambda = \frac{5}{12}$$
$$d^2L = \frac{10\lambda}{3}(dx)^2 + 30\lambda(dy)^2 > 0$$
$$M_1(-1.2, -0.3) - \text{минимум}$$

$$\lambda = -\frac{5}{12}$$
$$d^2L = -\frac{10\lambda}{3}(dx)^2 - 30\lambda(dy)^2 < 0$$
$$M_2(1.2, 0.3) - \text{максимум}$$

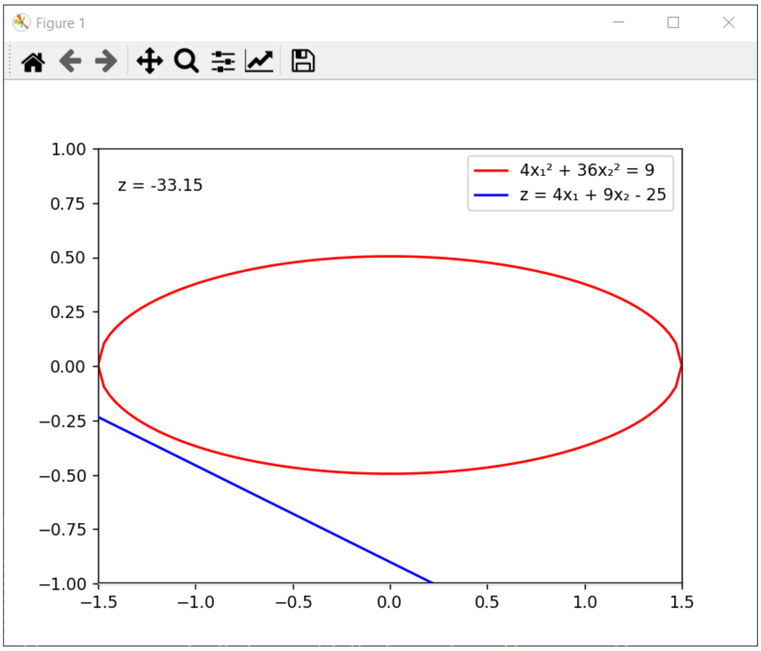

Рис. 2.1. Решение

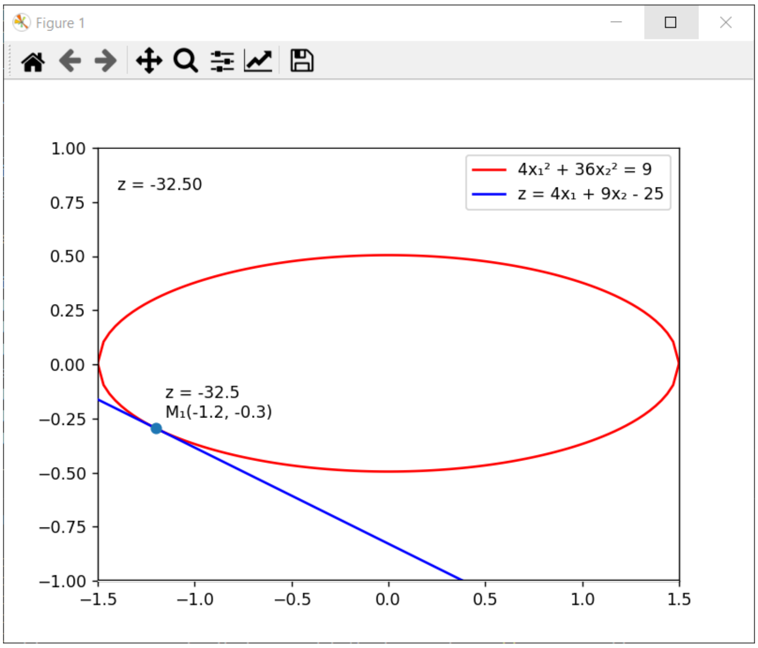

Рис. 2.2. Решение

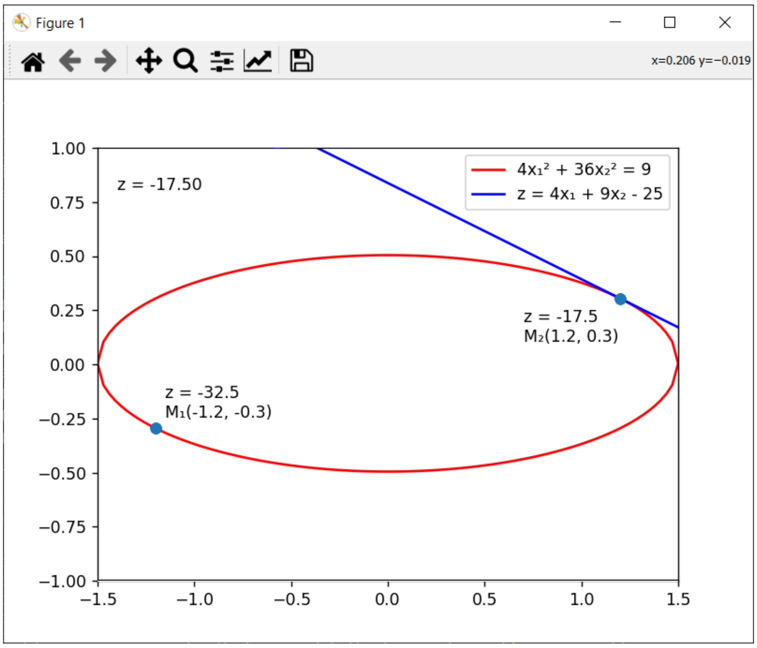

Рис. 2.3. Решение

Задание 3.

Фабрика производит мужские сорочки и женские блузки для конкретного заказчика. Заказчик принимает всю продукцию, вырабатываемой фабрикой. Производство швейного изделия состоит из раскроя, пошива и пакетирования готового изделия. На участке раскроя работают 25 человек, непосредственно на пошиве изделий – 35 человек и пакетируют готовые изделия 5 человек. Фабрика работает в одну смену (8 часов) пять дней в неделю. Трудозатраты на выпускаемые фабрикой изделия и доход от них показаны в следующей таблице.

Изделие	**Раскрой**	**Пошив**	**Пакетирование**	**Доход**
	(минуты на изделие)			(в д.е. на изделие)
Рубашка	20	70	12	2,50
Блузка	60	60	4	3,20

Определите оптимальную структуру еженедельного производства для этой швейной фабрики. Если магазину потребуется не менее 2000 рубашек и 3000 блузок в неделю, то сможет ли швейная фабрика выполнить этот заказ при 5-дневной рабочей неделе? Если нет, то какой выход из этой ситуации вы можете предложить и какое оптимальное решение возможно в этом случае? Определите стоимость одного часа рабочего времени, затрачиваемого отдельно на раскрой, пошив и пакетирование. Предположим, что можно организовать сверхурочную работу на участках раскроя и пошива. Какую максимальную почасовую добавку за сверхурочные может предложить швейная фабрика?

Решение:

$$F = 2.5x_1 + 3.2x_2 \rightarrow max$$

$$\begin{cases} 20x_1 + 60x_2 \leq 25 \cdot 5 \cdot 8 \cdot 60 \\ 70x_1 + 60x_2 \leq 35 \cdot 5 \cdot 8 \cdot 60 \\ 12x_1 + 4x_2 \leq 5 \cdot 5 \cdot 8 \cdot 60 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

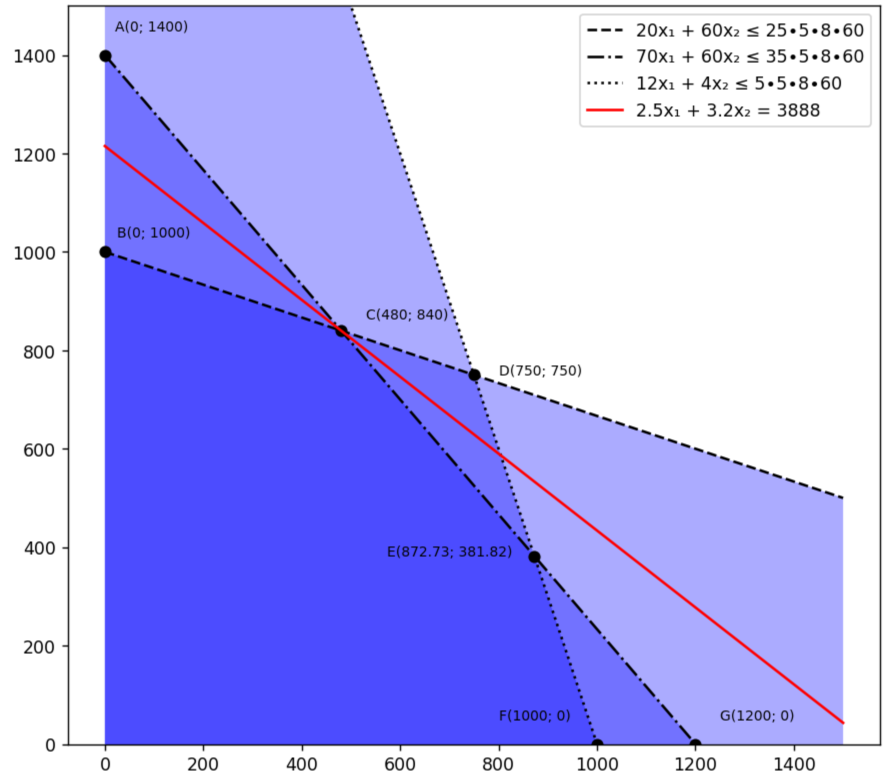

Рис. 3.1. Оптимум

$$x_1 = 480$$
$$x_2 = 840$$
$$F = 3888$$

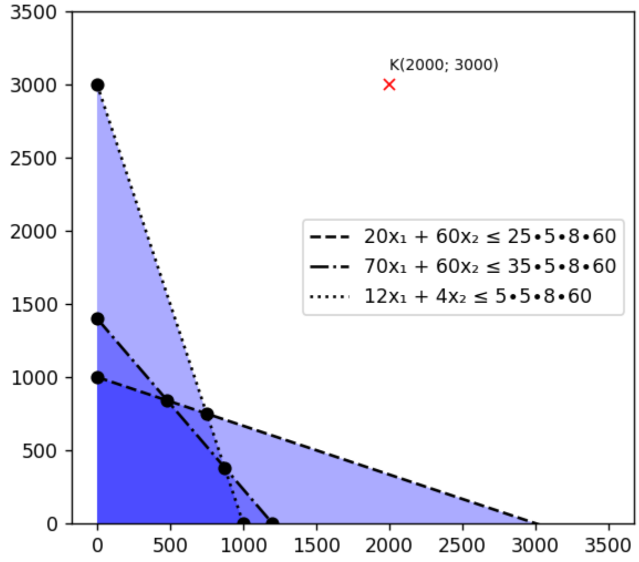

Рис. 3.2. Заказ

Точка K не входит в пространство решений, следовательно, заказ не выполним при исходном рабочем графике. Возможные выходы из ситуации:

- Произвести максимум с точки зрения дохода: 480 рубашек и 840 блузок;
- Произвести максимум рубашек: 1000;
- Произвести максимум блузок: 1000;
- Увеличить срок с 5 дней (недели) до 20 (месяца) (см. рис. 3.3).

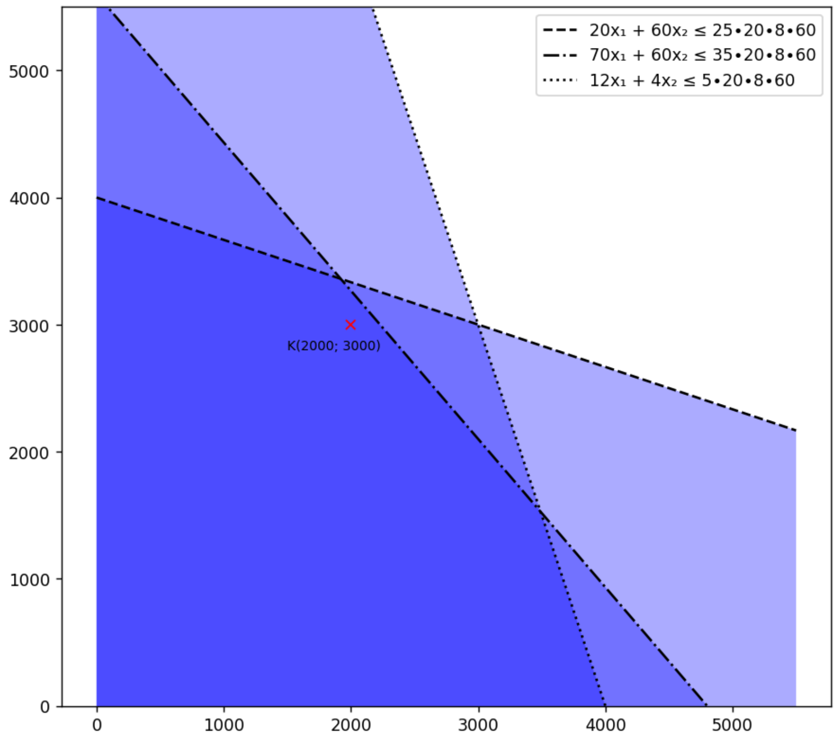

Рис. 3.3. Пространство решений при 20 днях работы

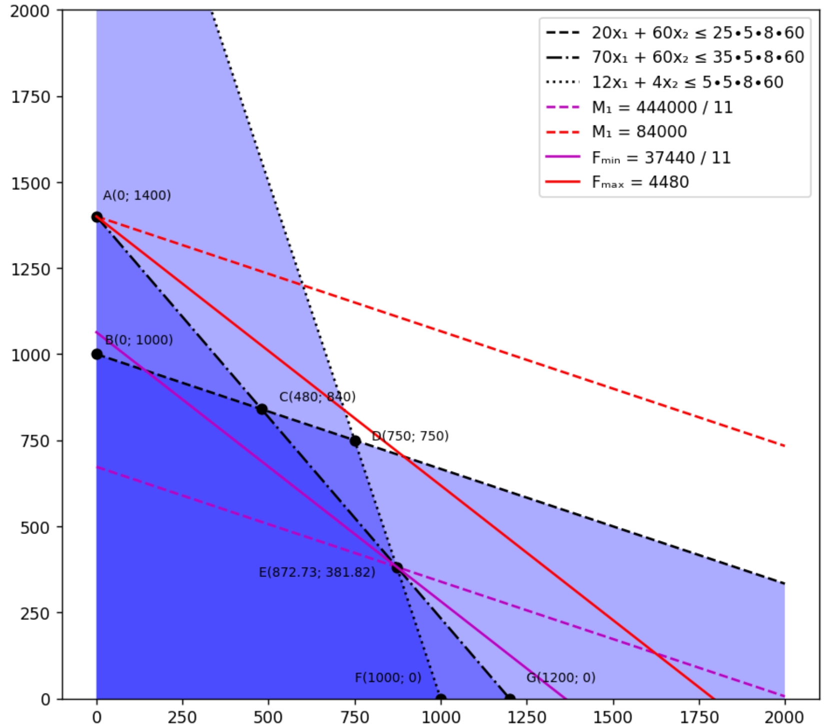

Рис. 3.4. Интервал осуществимости для раскроя

$$y_1 = \frac{4480 - \dfrac{37440}{11}}{84000 - \dfrac{444000}{11}} \cdot 60 = 1.48$$

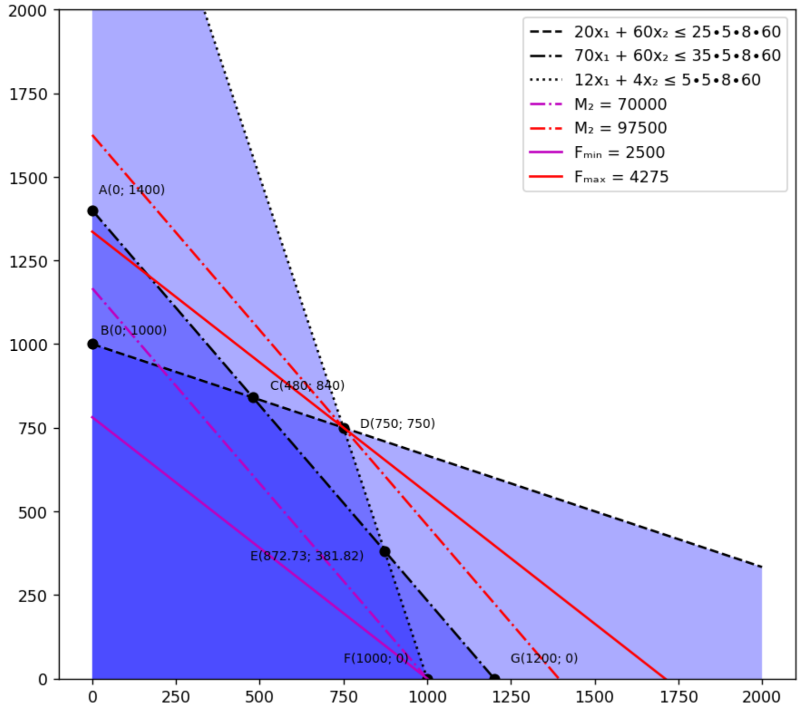

Рис. 3.5. Интервал осуществимости для пошива

$$y_2 = \frac{4275 - 2500}{97500 - 70000} \cdot 60 \approx 3{,}87$$

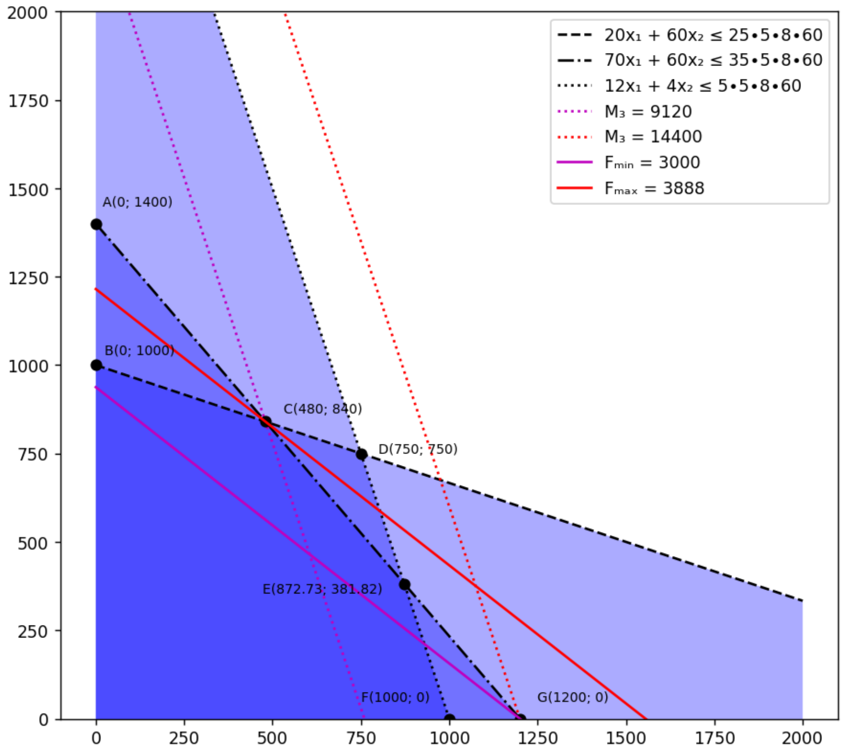

Рис. 3.6. Интервал осуществимости для пакетирования

$$y_3 = \frac{3888 - 3000}{14400 - 9120} \cdot 60 \approx 10{,}09$$

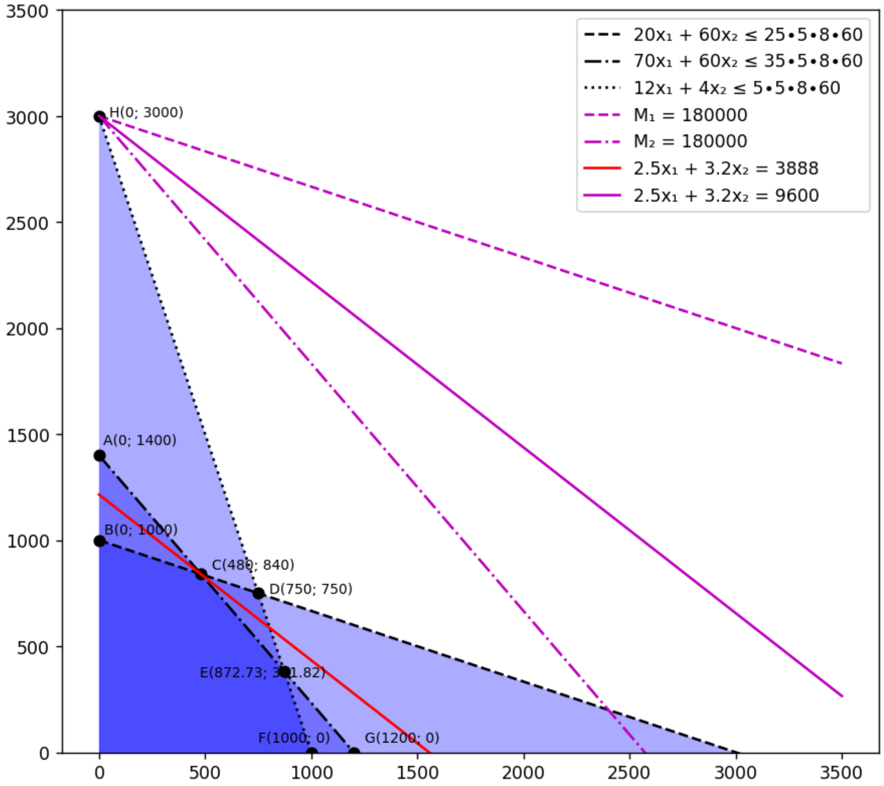

Рис. 3.7. Интервалы осуществимости для раскроя и пошива при сверхурочных

$$y_1^* = \frac{9600 - 3888}{180000 - 25 \cdot 5 \cdot 8 \cdot 60} \cdot 60 = 28{,}56$$

$$y_2^* = \frac{9600 - 3888}{180000 - 35 \cdot 5 \cdot 8 \cdot 60} \cdot 60 = 3{,}57$$

Вывод: в ходе выполнения лабораторной работы был изучен математический аппарат математического программирования на примере задач небольшой размерности, допускающих графическое решение.

ПРИЛОЖЕНИЯ

Листинг:

LW2_1.py:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import math

X = np.linspace(0, 3, 40)
Y_1 = 1 / X
Y_2 = np.sqrt(9 - X**2)

red_fill_x = [X[-1]]
for x in X:
    red_fill_x.append(x)
red_fill_x.append(X[-1])

red_fill_y = [Y_1[1]]
for y in Y_1:
    red_fill_y.append(y)
red_fill_y.append(Y_1[1])

green_fill_x = [X[0]]
for x in X:
    green_fill_x.append(x)
green_fill_x.append(X[0])

green_fill_y = [Y_2[-1]]
for y in Y_2:
    green_fill_y.append(y)
green_fill_y.append(Y_2[-1])


fig, ax = plt.subplots()
ln, = ax.plot([], [], 'b')
dots, = ax.plot([], [], 'o')
min_text = ax.text(0.1, 0.3, '', fontsize=10)
max_text = ax.text(0.1, 0.1, '', fontsize=10)
z_text = ax.text(0.1, 0.5, '', fontsize=10)

ax.fill(green_fill_x, green_fill_y, 'green', alpha=0.25)
ax.fill(red_fill_x, red_fill_y, 'red', alpha=0.25)

ax.plot(X, Y_1, 'r')
ax.plot(X, Y_2, 'g')

z = 0
last_x_points = None

def init():
```

```
    ax.set_xlim(0, 3)
    ax.set_ylim(0, 3)
    return ln, dots, min_text, max_text, z_text,

def update(frame):
    global z
    global last_x_points

    ydata = frame - X
    ln.set_data(X, ydata)

    z = frame
    z_text.set_text(f'z = {z:.2f}')

    x_dots = []
    y_dots = []

    D_1 = z**2 - 4
    if D_1 == 0:
        x = z / 2.0
        y = z - x
        x_dots.append(x)
        y_dots.append(y)

        min_text.set_text(f'min: z = {z}, x1 = {x}, y2 = {y}')
    elif D_1 > 0:
        x = (z + math.sqrt(D_1)) / 2.0
        y = z - x
        x_dots.append(x)
        y_dots.append(y)

        x = (z - math.sqrt(D_1)) / 2.0
        y = z - x
        x_dots.append(x)
        y_dots.append(y)

    D_2 = 72 - 4*z**2
    if D_2 == 0:
        x = 2*z / 4.0
        y = z - x
        x_dots.append(x)
        y_dots.append(y)

        max_text.set_text(f'max: z = {z}, x1 = {x}, y2 = {y}')
    elif D_2 > 0:
        last_x_points = []

        x = (2*z + math.sqrt(D_2)) / 4.0
        y = z - x
        x_dots.append(x)
        y_dots.append(y)
        last_x_points.append(x)
```

```
        x = (2*z - math.sqrt(D_2)) / 4.0
        y = z - x
        x_dots.append(x)
        y_dots.append(y)
        last_x_points.append(x)
    elif last_x_points != None:
        x = (last_x_points[0] + last_x_points[1]) / 2.0
        y = math.sqrt(9 - x**2)
        z = x + y
        max_text.set_text(f'max: z = {z:.2f}, x1 = {x:.2f}, y2 = {y:.2f}')

    dots.set_data(x_dots, y_dots)

    return ln, dots, min_text, max_text, z_text

ani = FuncAnimation(fig, update, frames=np.linspace(1, 5, 501),
                    init_func=init, blit=True, interval=10)
plt.show()
```

LW2_2.py:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

X = np.linspace(-1.5, 1.5, 100)

Y_1 = np.sqrt(9 - 4*X**2) / 6
Y_2 = -np.sqrt(9 - 4*X**2) / 6

fig, ax = plt.subplots()
ax.plot(X, Y_1, 'r', label='4x₁² + 36x₂² = 9')
ax.plot(X, Y_2, 'r')

z_line, = ax.plot([], [], 'b', label='z = 4x₁ + 9x₂ - 25')
dots, = ax.plot([], [], 'o')
z_text = ax.text(-1.4, 0.8, '', fontsize=10)
min_text = ax.text(-1.15, -0.25, '', fontsize=10)
max_text = ax.text(0.7, 0.1, '', fontsize=10)

x_dots = []
y_dots = []
min_is_placed = False
max_is_placed = False

def init():
    ax.set_xlim(-1.5, 1.5)
    ax.set_ylim(-1.0, 1.0)
    return z_line, dots, z_text, min_text, max_text,

def update(frame):
```

```
    global x_dots
    global y_dots
    global min_is_placed
    global max_is_placed

    z_line.set_data(X, (frame - 4*X + 25) / 9)

    if not min_is_placed and frame == -32.5:
        x_dots.append(-1.2)
        y_dots.append(-0.3)
        min_text.set_text(f'z = -32.5\nM₁(-1.2, -0.3)')
    if not max_is_placed and frame == -17.5:
        x_dots.append(1.2)
        y_dots.append(0.3)
        max_text.set_text(f'z = -17.5\nM₂(1.2, 0.3)')
    dots.set_data(x_dots, y_dots)

    z_text.set_text(f'z = {frame:.2f}')

    return z_line, dots, z_text, min_text, max_text,

ani = FuncAnimation(fig, update, frames=np.linspace(-35, -15, 401),
                    init_func=init, blit=True, interval=5)
ax.legend()
plt.show()
```

LW2_3.py:

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

X = np.linspace(0, 2000, 2001)

days = 5
hours = 8
Y_1 = (25*days*hours*60 - 20*X) / 60
Y_2 = (35*days*hours*60 - 70*X) / 60
Y_3 = (5*days*hours*60 - 12*X) / 4

fig, ax = plt.subplots()
ax.set_ylim(0, 2000)

ax.fill([0,         X[0],   X[-1],      X[-1]],
        [Y_3[-1],   Y_1[0], Y_1[-1],    Y_3[-1]],
        'blue', alpha=0.33)
ax.fill([0,         X[0],   X[-1],      X[-1]],
        [Y_3[-1],   Y_2[0], Y_2[-1],    Y_3[-1]],
        'blue', alpha=0.33)
ax.fill([0,         X[0],   X[-1],      X[-1]],
        [Y_3[-1],   Y_3[0], Y_3[-1],    Y_3[-1]],
        'blue', alpha=0.33)
```

```
ax.plot(X, Y_1, 'k--', label='20x₁ + 60x₂ ≤ 25·5·8·60')
ax.plot(X, Y_2, 'k-.', label='70x₁ + 60x₂ ≤ 35·5·8·60')
ax.plot(X, Y_3, 'k:', label='12x₁ + 4x₂ ≤ 5·5·8·60')

Y_1_min = (444000/11 - 20*X) / 60
Y_1_max = (84000 - 20*X) / 60
Y_2_min = (70000 - 70*X) / 60
Y_2_max = (97500 - 70*X) / 60
Y_3_min = (9120 - 12*X) / 4
Y_3_max = (14400 - 12*X) / 4

# ax.plot(X, Y_1_min, 'm--', label='M₁ = 444000 / 11')
# ax.plot(X, Y_1_max, 'r--', label='M₁ = 84000')
# ax.plot(X, Y_2_min, 'm-.', label='M₂ = 70000')
# ax.plot(X, Y_2_max, 'r-.', label='M₂ = 97500')
# ax.plot(X, Y_3_min, 'm:', label='M₃ = 9120')
# ax.plot(X, Y_3_max, 'r:', label='M₃ = 14400')

Y_1_ot = (180000 - 20*X) / 60
Y_2_ot = (180000 - 70*X) / 60

# ax.plot(X, Y_1_ot, 'm--', label='M₁ = 180000')
# ax.plot(X, Y_2_ot, 'm-.', label='M₂ = 180000')

# ax.plot(2000, 3000, 'rx')
# ax.text(2000 - 500, 3000 - 200, 'K(2000; 3000)', fontsize=8)

ax.plot(480, 840, 'ko')
ax.plot(9600/11, 4200/11, 'ko')
ax.plot(0, 35*5*8, 'ko')
ax.plot(0, 25*5*8, 'ko')
ax.plot(5*5*8*5, 0, 'ko')
ax.plot(5*5*8*6, 0, 'ko')
ax.plot(750, 750, 'ko')
ax.plot(0, 3000, 'ko')

ax.text(480 + 50, 840 + 25, 'C(480; 840)', fontsize=8)
ax.text(9600/11 - 400, 4200/11 - 25, f'E({9600/11:.2f}; {4200/11:.2f})',
fontsize=8)
ax.text(0 + 20, 35*5*8 + 50, f'A(0; {35*5*8})', fontsize=8)
ax.text(0 + 25, 25*5*8 + 30, f'B(0; {25*5*8})', fontsize=8)
ax.text(5*5*8*5 - 250, 0 + 50, f'F({5*5*8*5}; 0)', fontsize=8)
ax.text(5*5*8*6 + 50, 0 + 50, f'G({5*5*8*6}; 0)', fontsize=8)
ax.text(750 + 50, 750, f'D(750; 750)', fontsize=8)
ax.text(0 + 50, 3000, f'H(0; 3000)', fontsize=8)

F = 3888
Y = (F - 2.5*X) / 3.2
# ax.plot(X, Y, 'r', label='2.5x₁ + 3.2x₂ = 3888')

F_ot = 9600
Y_ot = (F_ot - 2.5*X) / 3.2
```

```
# ax.plot(X, Y_ot, 'm', label='2.5x₁ + 3.2x₂ = 9600')

F_min = 3000
F_max = 3888
Y_min = (F_min - 2.5*X) / 3.2
Y_max = (F_max - 2.5*X) / 3.2
# ax.plot(X, Y_min, 'm', label='Fₘᵢₙ = 3000')
# ax.plot(X, Y_max, 'r', label='Fₘₐₓ = 3888')

plt.gca().set_aspect('equal')

ax.legend()
plt.show()
```