



Министерство науки и высшего образования Российской Федерации  
Калужский филиал федерального государственного автономного  
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

**«Линейное программирование. Графический метод.»**

**по дисциплине: «Методы принятия решений в программной инженерии»**

Выполнил: студент группы ИУК4-72Б

\_\_\_\_\_  
(Подпись)

Губин Е.В.

\_\_\_\_\_  
(И.О. Фамилия)

Проверил:

\_\_\_\_\_  
(Подпись)

Никитенко У.В.

\_\_\_\_\_  
(И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

### Этапы выполнения работы:

- Составить математическую модель
- Описать переменные, параметры модели
- Составить целевую функцию и ограничения
- Применить графический метод для решения задачи
- Провести анализ чувствительности к входным данным

### Вариант 8

#### Формулировка задания:

Конвейер состоит из трех последовательных линий для сборки двух видов чипов: СН-1 и СН-2. Время, необходимое для сборки одного чипа на каждой линии, приведено в следующей таблице.

Сборочная линия	Количество минут, затрачиваемых на сборку одного изделия	
	СН-1	СН-2
1	6	4
2	5	5
3	4	6

Ежедневные профилактические работы на соответствующих линиях составляют 10%, 14% и 12% от всего рабочего времени, которое для любой линии не превышает 480 минут в смену.

Компания желает иметь такую структуру выпускаемой продукции, чтобы минимизировать время простоя всех трех линий

Определите стоимость одного процента уменьшения времени профилактических работ для каждой линии.

#### Результаты выполнения программы:

```
Оптимальное решение:  
x1 = 36 (СН-1)  
x2 = 46 (СН-2)  
Максимальное количество чипов: Z = 82
```

Рисунок 1 Оптимальное решение и максимальное количество чипов

На рисунке 2 показано графическое решение: для каждой линии построены графики (синий, оранжевый и зелёный). Область, которая находится ниже графика для каждой линии, является недопустимой, а область выше – допустимой.

Точка оптимума лежит на пересечении двух активных (ограничивающих) прямых, то есть там, где ресурсы (время линий) используются полностью, без простоев.

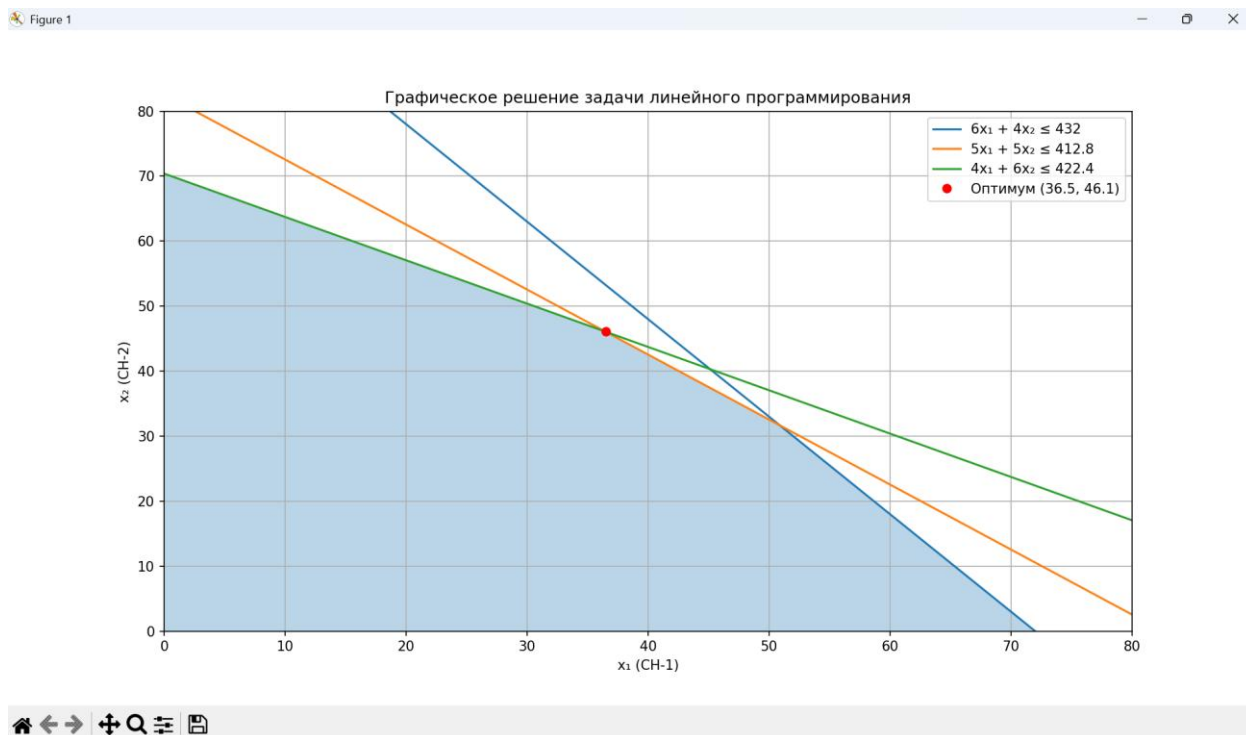


Рисунок 2 Графическое решение задачи

#### Анализ чувствительности:

Линия 1: время увеличилось с 432.0 до 436.8 мин

Линия 2: время увеличилось с 412.8 до 417.6 мин

Линия 3: время увеличилось с 422.4 до 427.2 мин

Рост выпуска при снижении профилактики на 1% на всех линиях: 0.96 чипов/смену

Рисунок 3 Анализ чувствительности

**Стоимость 1% уменьшения профилактики (в мин/1%):**  
 Линия 1: +4.8 мин доступного времени  
 Линия 2: +4.8 мин доступного времени  
 Линия 3: +4.8 мин доступного времени

Рисунок 4 Стоимость уменьшения профилактики для каждой линии

#### Листинг программы:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import linprog

A = [
    [6, 4],
    [5, 5],
    [4, 6]
]

T = [
    480 * (1 - 0.10),
    480 * (1 - 0.14),
```

```

    480 * (1 - 0.12)
]

c = [-1, -1]

res = linprog(c, A_ub=A, b_ub=T, bounds=[(0, None), (0, None)],
method='highs')

x1, x2 = res.x
max_value = -res.fun

print("Оптимальное решение:")
print(f"x1 = {x1:.2f} (CH-1)")
print(f"x2 = {x2:.2f} (CH-2)")
print(f"Максимальное количество чипов: Z = {max_value:.2f}")

x = np.linspace(0, 100, 200)
y1 = (T[0] - 6*x)/4
y2 = (T[1] - 5*x)/5
y3 = (T[2] - 4*x)/6

plt.figure(figsize=(8,6))
plt.plot(x, y1, label='6x1 + 4x2 ≤ 432')
plt.plot(x, y2, label='5x1 + 5x2 ≤ 412.8')
plt.plot(x, y3, label='4x1 + 6x2 ≤ 422.4')

plt.fill_between(x, 0, np.minimum(np.minimum(y1, y2), y3), alpha=0.3)
plt.plot(x1, x2, 'ro', label=f'Оптимум ({x1:.1f}, {x2:.1f})')

plt.xlim(0, 80)
plt.ylim(0, 80)
plt.xlabel("x1 (CH-1)")
plt.ylabel("x2 (CH-2)")
plt.legend()
plt.title("Графическое решение задачи линейного программирования")
plt.grid(True)
plt.show()

reductions = [0.09, 0.13, 0.11]
T_new = [480 * (1 - r) for r in reductions]

res_new = linprog(c, A_ub=A, b_ub=T_new, bounds=[(0, None), (0, None)],
method='highs')
new_value = -res_new.fun

delta = new_value - max_value
print("\nАнализ чувствительности:")
for i, (old, new) in enumerate(zip(T, T_new), 1):
    print(f"Линия {i}: время увеличилось с {old:.1f} до {new:.1f} мин")

print(f"Рост выпуска при снижении профилактики на 1% на всех линиях:
{delta:.2f} чипов/смену")

print("\nСтоимость 1% уменьшения профилактики (в мин/1%):")
for i in range(3):
    print(f"Линия {i+1}: +{480 * 0.01:.1f} мин доступного времени")

```

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки по решению задачи линейного программирования.