|  |  |
| --- | --- |
|  | **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 | |  |  | Губин Е.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил: | |  |  | Никитенко У.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | | | |
|  | - Балльная оценка:  - Оценка: | | | |

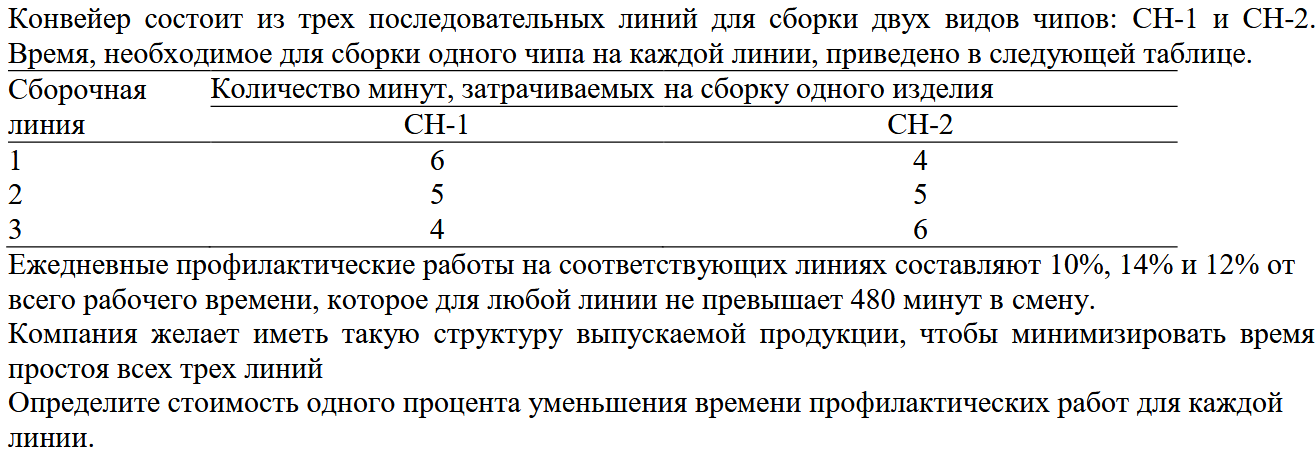
Калуга, 2025

**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:**



**Результаты выполнения программы:**

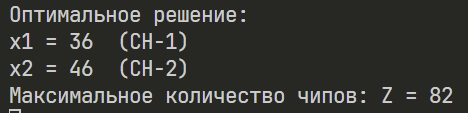


Рисунок 1 Оптимальное решение и максимальное количество чипов

На рисунке 2 показано графическое решение: для каждой линии построены графики (синий, оранжевый и зелёный). Область, которая находится ниже графика для каждой лини, 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.

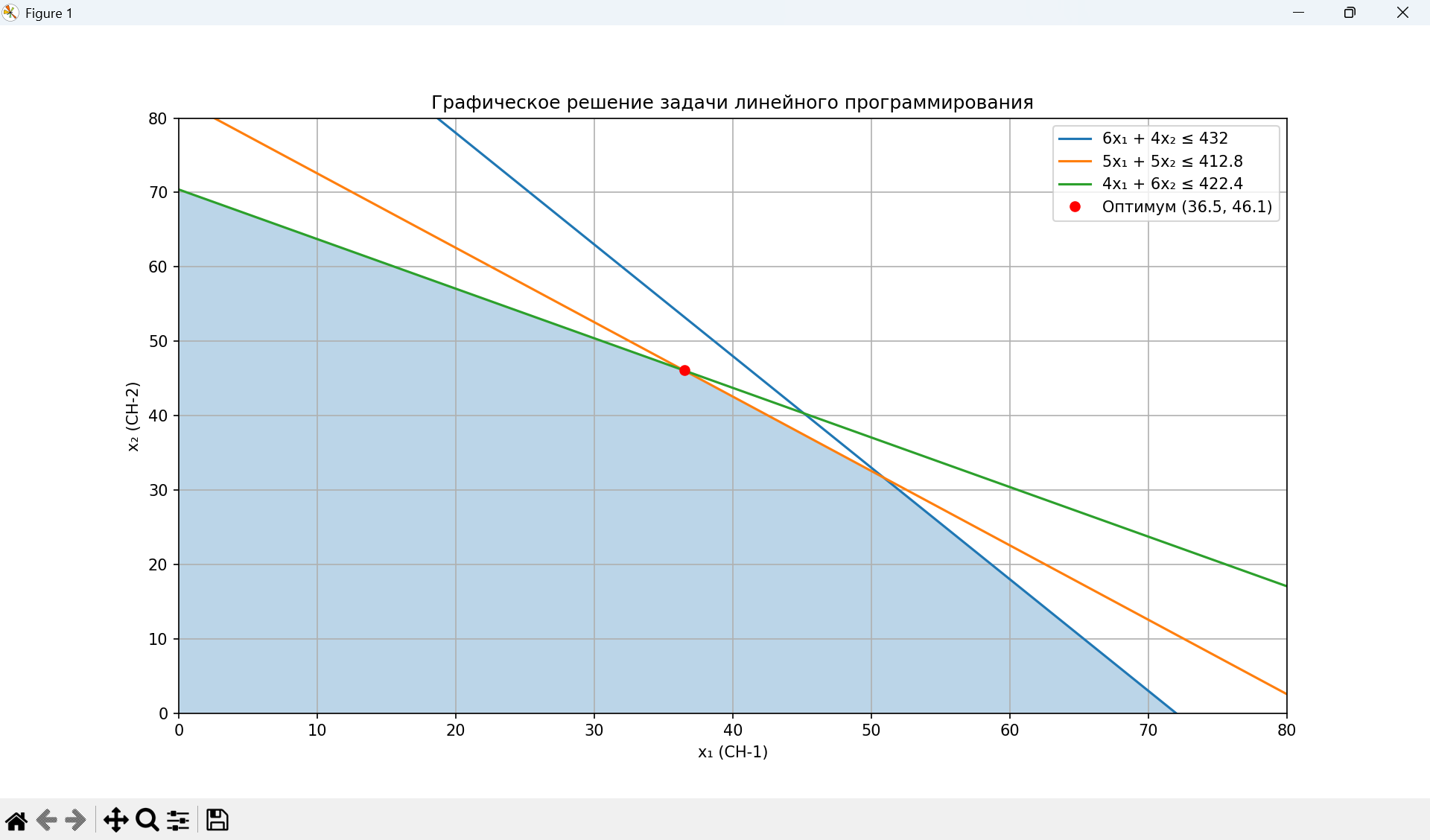


Рисунок 2 Графическое решение задачи

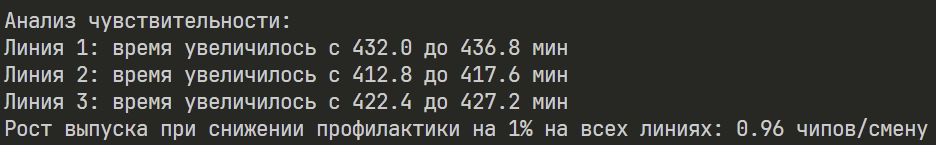


Рисунок 3 Анализ чувствительности

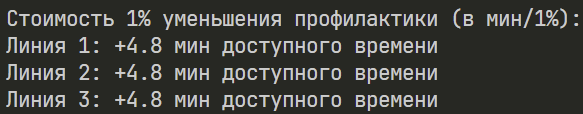


Рисунок 4 Стоимость уменьшения профилактики для каждой линии

**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.