Нуриева Джамиля Эльхановна, 1 группа 2 подгруппа, 30.10.2025

Лабораторная работа №5

**Теоретическая часть**

**Линейное программирование (ЛП)** — это фундаментальный раздел математического программирования, посвящённый исследованию и решению экстремальных задач с линейной целевой функцией и линейными ограничениями. Основная особенность ЛП заключается в том, что как целевая функция, так и все ограничения представляют собой линейные зависимости от переменных решения.

**Ключевые характеристики задач ЛП:**

- **Линейность** — все математические зависимости в задаче являются линейными функциями

- **Детерминированность** — все параметры задачи считаются известными и постоянными

- **Пропорциональность** — вклад каждой переменной в целевую функцию и ограничения пропорционален её значению

- **Аддитивность** — общий эффект от всех переменных равен сумме эффектов от каждой переменной в отдельности

Стандартные формы задач линейного программирования

1. Основная форма задачи максимизации

```

Максимизировать: F(x) = c₁x₁ + c₂x₂ + ... + cₙxₙ

При условиях:

a₁₁x₁ + a₁₂x₂ + ... + a₁ₙxₙ ≤ b₁

a₂₁x₁ + a₂₂x₂ + ... + a₂ₙxₙ ≤ b₂

...

aₘ₁x₁ + aₘ₂x₂ + ... + aₘₙxₙ ≤ bₘ

x₁, x₂, ..., xₙ ≥ 0

```

2. Каноническая форма задачи минимизации

```

Минимизировать: F(x) = cᵀx

При условиях:

A\_eq · x = b\_eq

x ≥ 0

```

3. Общая форма с различными типами ограничений

```

Минимизировать (или максимизировать): F(x) = cᵀx

При условиях:

A\_ub · x ≤ b\_ub (ограничения-неравенства)

A\_eq · x = b\_eq (ограничения-равенства)

l ≤ x ≤ u (границы переменных)

```

Математические методы решения задач ЛП

Историческая справка: Разработан Джорджем Данцигом в 1947 году, остаётся одним из наиболее широко используемых алгоритмов.

**Основные принципы работы:**

1. Геометрическая интерпретация: Допустимое множество задачи ЛП образует выпуклый многогранник (симплекс) в n-мерном пространстве

2. Фундаментальная теорема ЛП: Если оптимальное решение существует, то оно достигается хотя бы в одной из вершин многогранника

3. Итерационная процедура: Алгоритм последовательно переходит от одной вершины к смежной, улучшая значение целевой функции

**Пошаговый алгоритм:**

1. Приведение задачи к канонической форме

2. Нахождение начального допустимого базисного решения

3. Проверка текущего решения на оптимальность

4. Если решение не оптимально:

- Выбор переменной для ввода в базис

- Выбор переменной для вывода из базиса

- Пересчёт симплекс-таблицы

5. Повторение шагов 3-4 до достижения оптимальности

**Преимущества симплекс-метода:**

- Высокая эффективность для большинства практических задач

- Возможность получения информации о двойственных переменных

- Хорошая обусловленность для хорошо масштабированных задач

**Ограничения:**

- Теоретическая сложность в худшем случае — экспоненциальная

- Чувствительность к масштабированию задачи

Методы внутренней точки: современный подход

История развития: Получили широкое распространение после работы Кармаркара (1984 год)

**Классификация методов:**

- Методы проекции градиента

- Барьерные методы (логарифмические барьерные функции)

- Методы потенциальных функций

- Методы прогноз-корректор

**Основные принципы:**

1. Работа во внутренности допустимой области, а не на границе

2. Использование специальных траекторий (центральный путь)

3. Приближение к решению через последовательность внутренних точек

**Преимущества методов внутренней точки:**

- Полиномиальная сложность в худшем случае

- Эффективность для крупных разреженных задач

- Устойчивость к вырожденности

4. Теория двойственности и условия оптимальности

**Функция Лагранжа и условия ККТ**

**Для задачи условной оптимизации:**

```

Минимизировать: f(x)

При условиях: g\_i(x) ≤ 0, i = 1,...,m

h\_j(x) = 0, j = 1,...,p

```

Функция Лагранжа имеет вид:

```

L(x, μ, λ) = f(x) + Σμ\_i·g\_i(x) + Σλ\_j·h\_j(x)

```

Условия Каруша-Куна-Таккера (KKT) — необходимые условия оптимальности:

1. Стационарность:

∇ₓL(x\*, μ\*, λ\*) = 0

1. Допустимость прямого решения:

g\_i(x\*) ≤ 0, h\_j(x\*) = 0

1. Допустимость двойственного решения:

μ\_i\* ≥ 0

4. Дополняющая нежёсткость:

μ\_i\* · g\_i(x\*) = 0

**Экономическая интерпретация множителей Лагранжа**

Множители Лагранжа (теневые цены) имеют важную экономическую интерпретацию:

- Для ограничения g\_i(x) ≤ b\_i: множитель μ\_i показывает, насколько изменится оптимальное значение целевой функции при увеличении правой части b\_i на единицу

- Для ограничения h\_j(x) = b\_j: множитель λ\_j показывает скорость изменения целевой функции при изменении b\_j

5. Практическая реализация в Python

**Использование scipy.optimize.linprog**

from scipy.optimize import linprog

result = linprog(c, A\_ub=A\_ub, b\_ub=b\_ub, A\_eq=A\_eq, b\_eq=b\_eq,

bounds=bounds, method='highs')

**Детальное описание параметров**

- c — вектор коэффициентов целевой функции (для минимизации)

- A\_ub, b\_ub — матрица и вектор ограничений-неравенств (A\_ub @ x ≤ b\_ub)

- A\_eq, b\_eq — матрица и вектор ограничений-равенств (A\_eq @ x = b\_eq)

- bounds — границы переменных в формате [(lower, upper), ...]

- method — алгоритм решения:

- `highs' — современные реализации методов внутренней точки

- 'simplex'— классический симплекс-метод

- 'interior-point' — метод внутренней точки

**Особенности практического использования**

Преобразование задач максимизации:

# Для максимизации cᵀx → max эквивалентно минимизации -cᵀx → min

c\_max = [8000, 12000] # Исходные коэффициенты для максимизации

c\_min = [-8000, -12000] # Преобразованные для linprog

Обработка ограничений разных типов:

# Ограничение 3x₁ + 2x₂ ≥ 6 преобразуется в -3x₁ - 2x₂ ≤ -6

A\_ub = [[-3, -2]]

b\_ub = [-6]

# Ограничение равенства x₁ + x₂ = 15 остаётся без изменений

A\_eq = [[1, 1]]

b\_eq = [15]

**Анализ результатов**

Структура объекта результата:

result = linprog(...)

print("Статус:", result.message)

print("Оптимальное решение:", result.x)

print("Оптимальное значение:", result.fun)

print("Количество итераций:", result.nit)

print("Множители Лагранжа:")

print(" - Для неравенств:", result.slack)

print(" - Для равенств:", result.con)

6. Области применения и практическая значимость

Линейное программирование находит применение в различных областях:

- Экономика и финансы: Оптимизация портфеля, планирование производства

- Логистика: Транспортные задачи, управление цепями поставок

- Техника: Оптимизация энергосистем, проектирование сетей

- Военное дело: Распределение ресурсов, планирование операций

Теоретическая основа ЛП обеспечивает строгий математический аппарат для решения практических задач оптимизации, а современные вычислительные средства делают эти методы доступными для широкого круга приложений.

**Задача 1: Оптимизация производства электроники**

Постановка:  
Компания производит смартфоны и планшеты. Требуется составить план производства, максимизирующий прибыль.

Переменные принят(ы): x1 — количество смартфонов, x2 — количество планшетов.

Математическая модель:

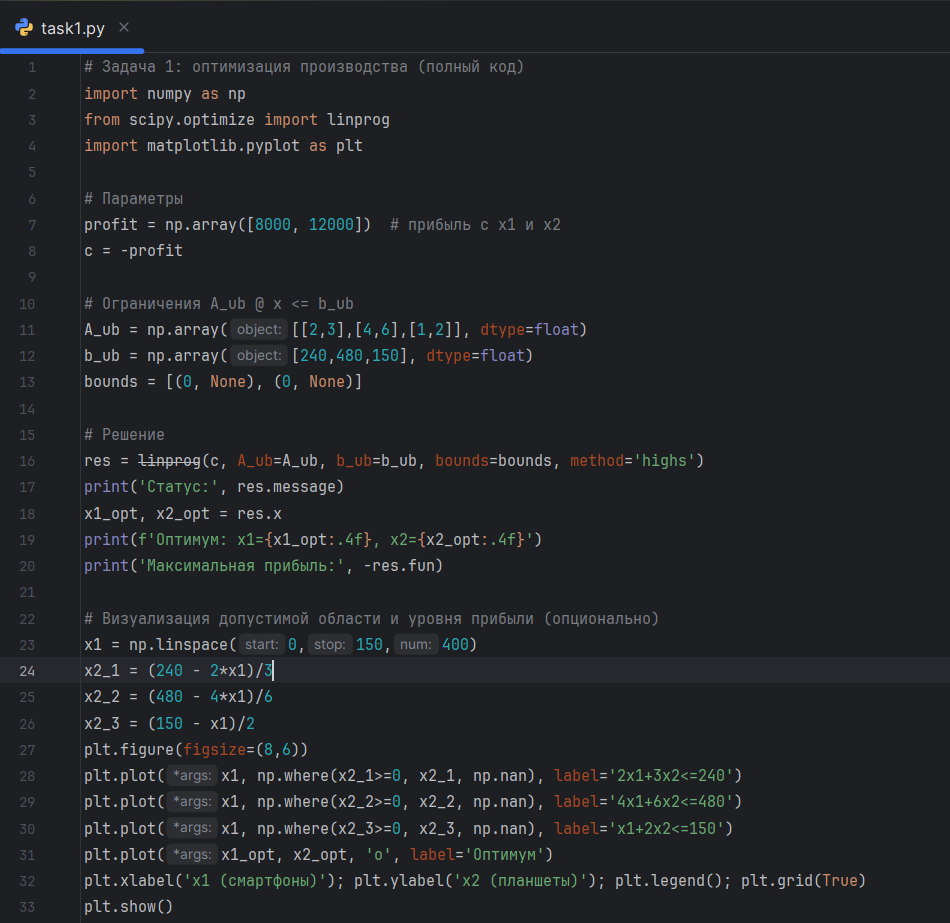
Целевая функция (максимизация прибыли):  
P(x) = 8000 x1 + 12000 x2 → max

Ограничения:  
2x1 + 3x2 ≤ 240 (процессорное время)  
4x1 + 6x2 ≤ 480 (оперативная память)  
x1 + 2x2 ≤ 150 (аккумуляторы)  
x1, x2 ≥ 0

**Функция Лагранжа и ККТ**

Для приведения к задаче минимизации используем F = -P. Ограничения переписаны в виде g\_i(x) ≤ 0.  
Функция Лагранжа:  
L(x1,x2,μ1,μ2,μ3) = -8000 x1 - 12000 x2 + μ1(2x1+3x2-240) + μ2(4x1+6x2-480) + μ3(x1+2x2-150),  
где μ\_i ≥ 0 — множители Лагранжа.  
  
Условия ККТ включают стационарность (градиент L по x = 0), допустимость по первичной и двойственной переменным,  
а также условие дополнительной нежёсткости μ\_i \* g\_i(x) = 0.

**Код решения**

  
**Результаты (Задача 1)**

Оптимальное число смартфонов x1 = 30.00

Оптимальное число планшетов x2 = 60.00

Максимальная прибыль = 960000.00 руб.

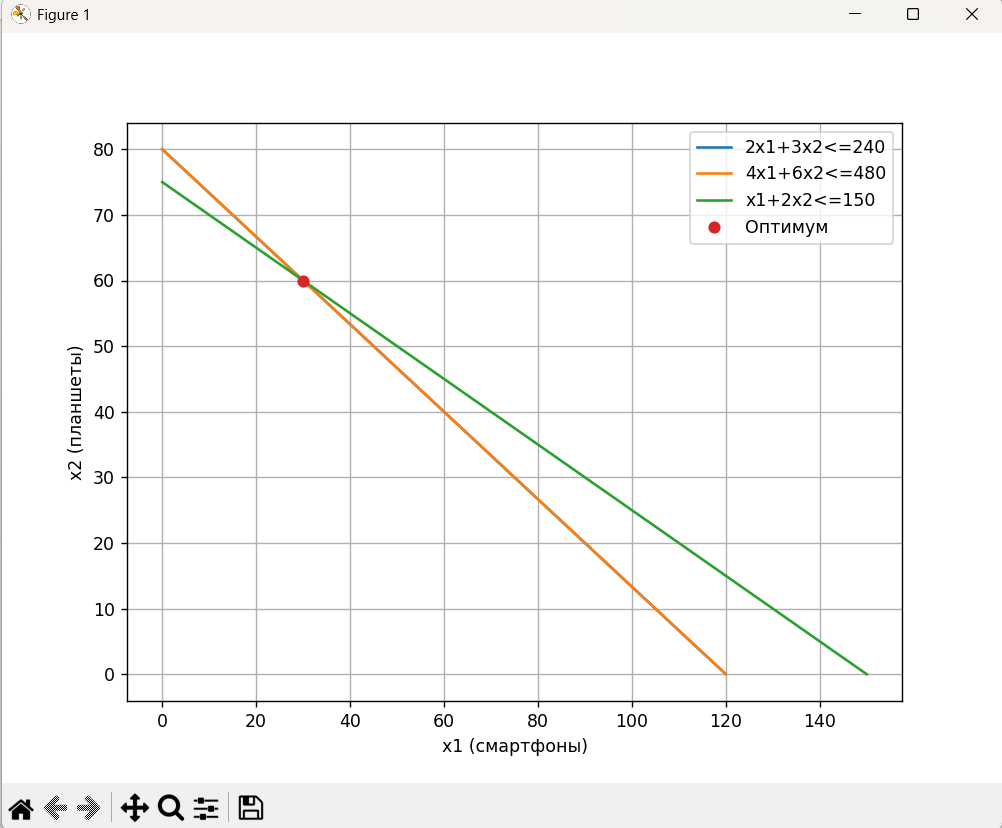
Активность ограничений (True = ограничение активно / исчерпано):

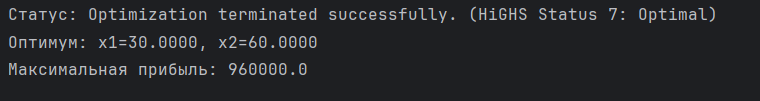
Ограничение 1: активно (запас = 0.000000)

Ограничение 2: активно (запас = 0.000000)

Ограничение 3: активно (запас = 0.000000)

График допустимой области и оптимума:





**Задача 2**

Постановка:  
Два склада поставляют три базы. Требуется минимизировать суммарную стоимость перевозок при полном удовлетворении потребностей баз.

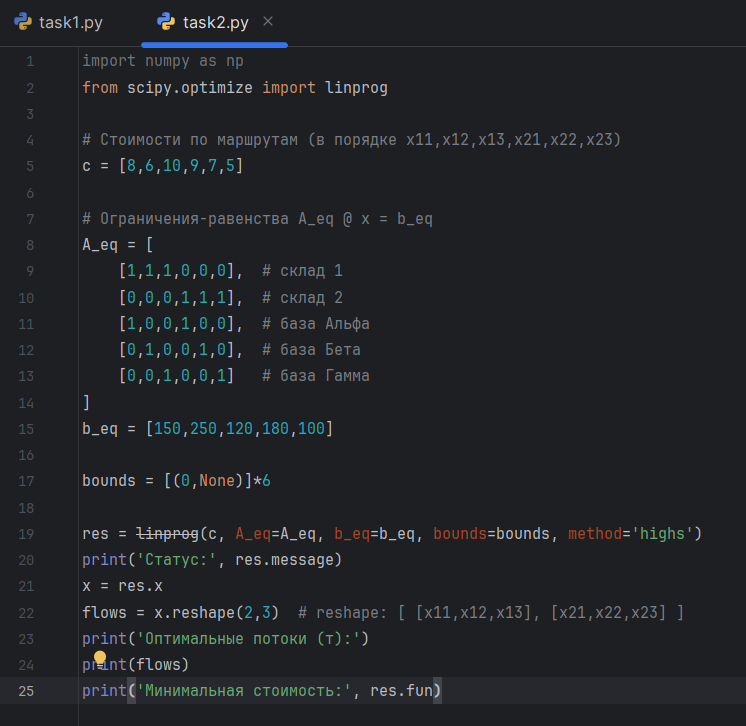
**Математическая модель**

Переменные: x11, x12, x13, x21, x22, x23 — объёмы (тонны) перевозок:  
Целевая функция:  
Z = 8x11 + 6x12 + 10x13 + 9x21 + 7x22 + 5x23 → min  
  
Ограничения (равенства):  
Склад 1: x11 + x12 + x13 = 150  
Склад 2: x21 + x22 + x23 = 250  
База Альфа: x11 + x21 = 120  
База Бета: x12 + x22 = 180  
База Гамма: x13 + x23 = 100  
x\_ij ≥ 0

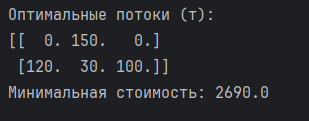
**Функция Лагранжа**

Пусть λ1, λ2 — множители для ограничений складов, ν1,ν2,ν3 — для ограничений баз.  
Лагранж: L = Z + λ1(x11+x12+x13-150) + λ2(x21+x22+x23-250)  
 + ν1(x11+x21-120) + ν2(x12+x22-180) + ν3(x13+x23-100)  
В оптимуме для используемых маршрутов выполняется условие λ\_i + ν\_j = c\_ij.

**Код решения**



**Результаты (Задача 2)**



**Анализ результатов и чувствительность**

Задача 1 (производство):  
- Оптимум: x1 = 30.00, x2 = 60.00, прибыль = 960000.00 руб.  
- Если добавить 10 часов процессорного времени (240 -> 250), решив задачу заново, можно увидеть изменение прибыли  
 и значений переменных — это и есть простой анализ чувствительности по ресурсу.  
  
Задача 2 (снабжение):  
- Минимальная стоимость = 2690.00 усл.ед.  
- Все потребности баз удовлетворены, суммарный запас на складах равен суммарной потребности (задача сбалансирована).  
- Изменение стоимости маршрута или потребностей может привести к перераспределению потоков; для анализа достаточно менять  
 соответствующие элементы в векторе c или в b\_eq и решать задачу повторно.

**Выводы**

- Методы ЛП (симплекс и внутренние точки) удобны для задач оптимизации линейной структуры.  
- Использование scipy.optimize.linprog позволяет быстро получать оптимальные планы и проводить анализ чувствительности простыми перезапусками.  
- Множители Лагранжа дают экономический смысл дополнительной единицы ресурса (теневая цена).