НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи
з дисципліни «АЛГОРИТМИ І СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ 1.МАТЕМАТИЧНІ АЛГОРИТМИ»
на тему
"Лінійне програмування"

Виконала:
студентка групи КМ-01
Резниченко Є. С.

Перевірила:
Асистент кафедри ПМА
Ковальчук-Химюк Л. О.

Київ — 2023

Зміст

Вступ

Метою роботи є побудова математичних моделей лінійного програмування, практичне розв’язання задач лінійного програмування з використанням СКМ.

Основна частина

Варіант 6

Задача о сумішах

Види ресурсів	К-сть одиниць поживних речовин на одиницю продукції				Максимальна норма поживних речовин	Прибуток від реалізації одиниці продукції			
	P_1	P_2	P_3	P_4		C_{P_1}	C_{P_2}	C_{P_3}	C_{P_4}
S_1	3	1	1	0	9				
S_2	1	2	0	0	8	5	6	–	–
S_3	1	6	0	0	12				

Математична модель:

$$\max(f(x)) = 5x_1 + 6x_2$$

$$\begin{cases} 3x_1 + 1x_2 + 1x_3 \le 9 \\ 1x_1 + 2x_2 \le 8 \\ 1x_1 + 6x_2 \le 12 \end{cases}$$

$$x_1, x_2, x_3 \ge 0$$

Метод розв’язання – симплекс [3]

Теоретичні відомості

1. Об'єктивна функція (функція максимізації прибутку):

Об'єктивна функція у задачі лінійного програмування на максимізацію прибутку представляє собою лінійну функцію, яка виражає відношення між змінними рішення та прибутком. Наприклад, якщо у вас є змінні рішення x_1, x_2, …, x_n, а ціна кожного товару або послуги, які ви продаєте, відома, то об'єктивна функція може виглядати наступним чином:

Максимізувати P = c_1x_1 + c_2x_2 + … + c_n*x_n,

де P - прибуток, а c_1, c_2, …, c_n - ціни на відповідні товари або послуги.

2. Обмеження:

Лінійне програмування також включає систему обмежень, які відображають реальні обмеження чи обставини, з якими стикається організація чи особа. Обмеження можуть впливати на кількість ресурсів, виробничі можливості та інші фактори. Наприклад:

a_11*x_1 + a_12*x_2 + … + a_1n*x_n <= b_1,

a_21*x_1 + a_22*x_2 + … + a_2n*x_n <= b_2,

a_m1*x_1 + a_m2*x_2 + … + a_mn*x_n <= b_m,

де a_ij - коефіцієнти, що визначають вплив змінної x_j у обмеженні i, а b_i - права частина обмеження i.

3. Не від'ємність змінних:

Зазвичай вводиться умова, що змінні рішення повинні бути не від'ємними:

x_1 >= 0,

x_2 >= 0,

x_n >= 0.

Метою лінійного програмування є знайти значення змінних рішення, які максимізують (чи мінімізують) об'єктивну функцію при виконанні всіх обмежень. Рішення задачі лінійного програмування може бути знайдено за допомогою різних алгоритмів, таких як симплекс-метод [3].

Симплекс-метод - це алгоритм для розв'язання задач лінійного програмування

1. Тип задачі: Симплекс-метод використовується для знаходження оптимального рішення задачі лінійного програмування, де шукається максимум чи мінімум лінійної функції за певними обмеженнями у вигляді лінійних рівнянь і нерівностей.
2. Базис і план: Алгоритм працює з базисним планом, який представляє собою набір змінних, які не рівні нулю, і визначається обмеженнями задачі. Ці змінні входять в базис, а інші - невходять
3. Ітерації: Симплекс-метод виконує ітерації, на кожній з яких покращується поточний базисний план. В кожній ітерації вибирається вихідна змінна і вхідна змінна, щоб покращити значення цільової функції.
4. Покращення плану: Змінюючи базис, метод спрямований на поступове покращення значення цільової функції до досягнення оптимуму.
5. Зупинка: Алгоритм завершується, коли більше немає можливості покращити значення цільової функції.
6. Практичні використання: Симплекс-метод застосовується в областях економіки, фінансів, транспортної логістики та інших галузях для оптимізації рішень при умовах обмежень.

Алгоритми розв’язання

1. Читання даних з файлу:
   - Зчитуються дані з файлу 'umova.txt', який містить інформацію про прибуток, матрицю та запаси.
   - Дані розділені за категоріями, такими як 'Прибуток', 'Матриця' та 'Запаси'.
2. Підготовка даних:
   - Функція `umova` повертає підготовлені дані про прибуток (`c`), матрицю (`A`), і запаси (`b`).
3. Вирішення задачі лінійного програмування з Python:
   - Використовується функція `linprog` [2] з бібліотеки Scipy [1] для знаходження максимального прибутку та оптимального плану виробництва.
   - Результати виводяться на екран, включаючи максимальний прибуток та оптимальний план випуску продукції.
4. Передача даних до Octave і вирішення задачі:
   - Використовується вбудована функція `glpk` для знаходження максимального прибутку та оптимального плану виробництва.
   - Результати виводяться на екран.

Код використовує дві мови програмування (Python та Octave) для розв'язання тієї ж самої задачі лінійного програмування і порівняння результатів.

Висновки

У результаті розв’язку задачі лінійного програмування на знаходження максимальної кількості поживних речовин для створення оптимального плану раціону харчування симлекс методом досягнуто таких результатів:

Розв'язок на мові Python

```
Максимальна кількість поживних речовин: 21.882
Оптимальний план раціону харчування:
P1 = 2.471
P2 = 1.588
P3 = 0.0
P4 = 0.0
```

Розв'язок на мові Octave

```
Розв'язок на мові Octave
Максимальний прибуток: 21.882353
Оптимальний план випуску продукції:
P1 = 2.470588
P2 = 1.588235
P3 = 0.000000
P4 = 0.000000
```

В результаті отримали дуже схожу максимальну кількість поживних речовин та оптимальний раціон харчування.

Відповіді на контрольні питання

1. Формулювання задачі лінійного програмування.
   Задача лінійного програмування (ЛП) полягає в пошуку оптимального рішення для задачі оптимізації, де цільова функція та обмеження представлені лінійними виразами. Основна мета полягає у мінімізації (або максимізації) цільової функції при заданих обмеженнях.
2. Еквівалентні форми задачі лінійного програмування, шляхи переходу від однієї форми до іншої - задачу ЛП можна перетворити з канонічної форми до інших форм, таких як стандартна форма, транспортна форма, та інші. Перетворення відбуваються шляхом введення нових змінних, заміни нерівностей на рівності, та інших математичних операцій. Це допомагає зручно вирішувати задачі в різних контекстах.
3. Геометрична інтерпретація області допустимих значень.
   Геометрично область допустимих значень в задачі лінійного програмування є областю в геометричному просторі, де всі можливі розв'язки задачі задовольняють обмеженням. Ця область може бути представлена як політоп, обмежений площинами, що відображають лінійні обмеження в задачі. Розв'язок задачі ЛП знаходиться в точці цього політопу, яка оптимізує цільову функцію.
4. Сутність сімплекс-методу - це алгоритм для розв'язання задач лінійного програмування, особливо в канонічній формі. Він базується на ітеративних кроках, в яких обчислюються нові точки на політопі, які ведуть до зменшення значення цільової функції. Сімплекс-метод рухається від поточного розв'язку до оптимального, шляхом обміну базової та небазової змінної, з метою зменшення значення цільової функції. Цей метод дозволяє знаходити оптимальне розв'язок відносно заданих обмежень.
5. Модифікований сімплекс-метод - це вдосконалена версія сімплекс-методу, яка розроблена для вирішення задач ЛП в ситуаціях, де можуть виникнути обмеження на допустиму множину або надмірність змінних базису. Цей метод може бути більш ефективним в деяких випадках, де класичний сімплекс-метод може бути менш продуктивним.
6. Сімплекс-метод зі штучним базисом використовується для розв'язання задач ЛП, які не мають початкового допустимого базису. Він включає штучні змінні в цільову функцію та починає з штучного базису. Після цього сімплекс-метод використовується для пошуку оптимального розв'язку, при цьому штучні змінні відкидаються на кожному кроці. Після видалення штучних змінних отримується дійсний базис, і сімплекс-метод продовжує пошук оптимального розв'язку.

Перелік посилань

1. Scipy - https://scipy.org
2. Scipy.linprog - https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html
3. Симплекс метод – https://elib.lntu.edu.ua/sites/default/files/elib_upload/Готовий%20підручник/page6.html

Додаток А – Код програми

Вміст файлу main.py :

```
from scipy.optimize import linprog


def umova():
    data = {'Вартість': [], 'Матриця': [], 'Норми': []}
    current_list = None

    with open('umova.txt', 'r', encoding='utf-8') as file:
        for line in file:
            line = line.strip()
            if line.startswith('#'):
                current_list = line[1:].strip()
            elif current_list:
                values = [float(x) for x in line.split()]
                if current_list == 'Матриця':
                    data[current_list].append(values)
                else:
                    data[current_list].extend(values)

    data['Матриця'] = [row for row in data['Матриця'] if row]

    return data['Вартість'], data['Матриця'], data['Норми']

c, A, b = umova()


# Виклик функції лінійного програмування з методом 'highs' для максимізації
прибутку
result = linprog(c, A_ub=A, b_ub=b, method='highs')

print('\n\n\nРезниченко Є. С. Варіант 6 Лаб 5\n')
if result.success:
    print('Максимальна кількість поживних речовин:', -round(result.fun, 3))
    print("Оптимальний план раціону харчування:")
    print('P1 =', round(result.x[0], 3))
    print('P2 =', round(result.x[1], 3))
    print('P3 =', round(result.x[2], 3))
    print('P4 =', round(result.x[3], 3))
else:
    print('Задачу не вдалося вирішити.')
```

Вміст файлу main.m :

% Читаємо дані з файлу

```
fid = fopen('D:\\KPI\\ASKM\\Labs\\Reznichenko\\Lab_5\\umova.txt', 'r');
data = struct('Вартість', [], 'Матриця', [], 'Норми', []);
current_list = [];

while ~feof(fid)
    line = fgetl(fid);
    line = strtrim(line);

    if strncmp(line, '#', 1)
        current_list = strtrim(line(2:end));
    elseif ~isempty(current_list)
        values = sscanf(line, '%f');

        if strcmp(current_list, 'Матриця')
            data.(current_list) = [data.(current_list); values'];
        else
            data.(current_list) = [data.(current_list), values];
        end
    end
end

fclose(fid);

% Визначаємо необхідні змінні
c = data.('Вартість');
A = data.('Матриця');
b = data.('Норми');
lb = [0; 0; 0; 0];
ub = [];
vartype = 'CCCC';
```

```
ctype = 'UUU';
sense = 1;

% Викликаємо функцію glpk
[x, fval, status] = glpk(c, A, b, lb, ub, ctype, vartype, sense);

% Виводимо результати
if status == 5
    fprintf('Задачу не вдалося вирішити.\n');
else
    fprintf('\nРозв''язок на мові Octave\n');
    fprintf('Максимальний прибуток: %f\n', -fval);
    fprintf('Оптимальний план випуску продукції:\n');
    fprintf('P1 = %f\n', x(1));
    fprintf('P2 = %f\n', x(2));
    fprintf('P3 = %f\n', x(3));
    fprintf('P4 = %f\n', x(4));
end
```