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"

Виконав:
студент групи КМ-03
Шаповалов. Г. Г.

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 15

Задача на максимальний прибуток

Види станків	Тривалість обробки виробу на станку			Прибуток від реалізації одиниці продукції			Запас потужності станків
	P_1	P_2	P_3	C_{P_1}	C_{P_2}	C_{P_3}	
T_1	2	0	–	6	6	–	20
T_2	1	2	–				37
T_3	1	4	–				30

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

$$\begin{cases} 2x_1 + 0x_2 \leq 20 \\ 1x_1 + 2x_2 \leq 37 \\ 1x_1 + 4x_2 \leq 30 \end{cases}$$

$$x_1, x_2 \geq 0$$

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 [4]**.

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` з бібліотеки Sciру 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:**
   - Дані передаються до Octave для вирішення тієї ж задачі.
   - Викликається функція `glpk` в Octave, і результати отримуються з Octave.
   - Результати знову 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 максимального прибутку симлекс методом досягнуто таких результатів:

Розв'язок на мові Python	Розв'язок на мові Octave
Максимальний прибуток: 90.0	Максимальний прибуток: 90.0
Оптимальний план випуску продукції:	Оптимальний план випуску продукції:
P1 = 10.0	P1 = 10.0
P2 = 5.0	P2 = 5.0
P3 = 0.0	P3 = 0.0

Отримано однакові результати. P3 = 0 оскільки за умовами задачі тривалість обробки виробу №3 та прибуток від його продажу дорівнює 0.

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. Oct2py - https://pypi.org/project/oct2py/
2. Scipy - https://scipy.org
3. Scipy.linprog - https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html
4. 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
from oct2py import octave


def umova():
    data = {'Прибуток': [], 'Матриця': [], 'Запаси': []}
    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['Прибуток'], data['Матриця'], data['Запаси']

c, A, b = umova()


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

print('\n\n\nШаповалов Г. Г. Варіант 15 Лаб 5\n')
if result.success:
    print('\nРозв\'язок на мові Python')
    print('Максимальний прибуток:', -round(result.fun, 2))
    print("Оптимальний план випуску продукції:")
    print('P1 =', round(result.x[0], 2))
    print('P2 =', round(result.x[1], 2))
    print('P3 =', round(result.x[2], 2))
else:
    print('Задачу не вдалося вирішити.')



lb = [0, 0, 0]
ub = []
```

```
vartype = 'CCC'
ctype = 'UUU'
sense = 1

# Передаємо дані до Octave
OCTAVE_DATA = {'c': c, 'A': A, 'b': b, 'lb': lb, 'ub': ub, 'vartype': vartype,
'ctype': ctype, 'sense': sense}

for key, value in OCTAVE_DATA.items():
    octave.push(key, value)


# Викликаємо функцію glpk з Octave
octave.eval('[x, fval, status] = glpk(c, A, b, lb, ub, ctype, vartype, sense);')

# Отримуємо результати з Octave
x = octave.pull('x')
fval = octave.pull('fval')
status = octave.pull('status')

if status == 5:  # Статус 5 вказує на невирішену задачу
    print("Задачу не вдалося вирішити.")
else:
    print('\nРозв\'язок на мові Octave')
    print("Максимальний прибуток:", -fval)
    print("Оптимальний план випуску продукції:")
    print(f"P1 = {x[0][0]}")
    print(f"P2 = {x[1][0]}")
    print(f"P3 = {x[2][0]}")
```