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-01
Романецький М.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 5

Задача на використання ресурсів

Види ресурсів	Витрати ресурсів на одиницю продукції			Запаси ресурсів	Прибуток від реалізації одиниці продукції		
	P_1	P_2	P_3		C_{P_1}	C_{P_2}	C_{P_3}
S_1	4	2	1	150000	100	150	200
S_2	6	0	2	170000			
S_3	0	2	4	100000			
S_4	8	7	0	200000			

У ході розв'язку задачі лінійного програмування на використання ресурсів треба знайти максимально можливий прибуток та скласти оптимальний план продажу продукції. Але спершу складемо ***математичну модель*** цієї задачі:

$$\max\big(f(x)\big) = 100x_1 + 150x_2 + 200x_3$$

$$\begin{cases} 4x_1 + 2x_2 + x_3 \le 150000 \\ 6x_1 + 2x_3 \le 170000 \\ 2x_2 + 4x_3 \le 100000 \\ 8x_1 + 7x_2 \le 200000 \end{cases}$$

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

Вимоги до ПЗ

1. Реалізувати перевірки на некоректний ввід (порожнє введення, символи замість числа, тощо).
2. У програмі повинно бути передбачено можливість гнучкого налагодження розмірності розв'язуваної задачі.
3. Має зберігатись логічно правильне розв'язання (нижня межа інтегрування не має перевищувати верхню, тощо)

Теоретичні відомості

Для виготовлення n продукції P1,...,Pn підприємство використовує m видів ресурсів S1,...,Sm. Запаси кожного виду обмежені і рівні b1,...,bm. На

виготовлення одиниці продукції j-го виду (j= m,1) витрачається aij одиниць i-го ресурсу (i= n,1). При реалізації j-ї продукції підприємство одержує cj одиниць прибутку. Необхідно скласти такий план випуску продукції, щоб при її реалізації отримати максимальний прибуток.

Задачі лінійного програмування (ЛП) про використання ресурсів є підкласом ЛП, в яких основною метою є ефективне розподілення ресурсів для досягнення максимальної користі або мінімізації витрат. Такі задачі виникають в багатьох галузях, де необхідно раціонально використовувати обмежені ресурси, такі як працівники, сировина, обладнання, час, інвестиції тощо. Основні характеристики задач ЛП про використання ресурсів включають:

1. Цільова функція: Зазвичай, це лінійна функція, яка виражає об'єктив оптимізації. Наприклад, це може бути максимізація прибутку або мінімізація витрат, пов'язаних з використанням ресурсів.
2. Ресурси: Це обмежені ресурси, які доступні для використання. Ресурси можуть бути виражені в різних одиницях вимірювання, таких як години праці, кількість сировини, обсяги обладнання, бюджет і т. д.
3. Обмеження: Обмеження в задачі ЛП вказують, як ресурси повинні бути розподілені між різними варіантами використання. Обмеження зазвичай виражаються у вигляді лінійних рівнянь або нерівностей. Наприклад, обмеження може обмежувати кількість працівників, які можуть бути виділені для конкретного завдання.
4. Змінні рішення: Це рішення, які визначають, скільки ресурсів виділяється для кожного варіанту використання. Зазвичай, це невідомі величини, які потрібно знайти в процесі оптимізації.
5. Не-від'ємність: Зазвичай, у задачах про використання ресурсів вимагається, щоб змінні рішення були не-від'ємними, оскільки від'ємні значення не мають фізичного змісту в контексті виділення ресурсів.

Приклади задач ЛП про використання ресурсів включають оптимізацію виробництва, розподіл транспортних потоків, розподіл робочого графіка працівників, планування проектів тощо. Лінійне програмування допомагає знайти оптимальні рішення в умовах обмежених ресурсів, що допомагає зекономити час, гроші та інші ресурси.

Алгоритми розв’язання

Імпортуємо бібліотеки Oct2py [1], Scipy [2]

Створюємо власну функцію umova, яка буде читати текстовий файл data.txt у якому будуть збереженні вхідні дані індивідуального варіанту. Ці дані

програма записує собі в пам’ять та приступає до виконання завдання. Першим на черзі йде код на мові Python. Тут за допомогою функції linprog [3] із бібліотеки Sciру будемо розв’язувати задачу лінійного програмування (далі ЛП). У якості параметрів передаємо обмеження нижньої межі, щоб не вийшло так, що ми взяли від’ємну кількість якогось товару. Далі робимо перевірку чи успішно виконали пошук. Якщо так, то виводимо відповіді на екран, якщо ні, то виводимо ‘Задачу не вдалося вирішити.’

Далі програма за допомогою бібліотеки Oct2py створює міст між Octave та Python. Вносить початкові дані в середовище октав за допомогою циклу, викликає вбудовану функцію glpk для максимізації нашої функції і пошуку максимального прибутку. Якщо програма повертає статус рівний 5, то задачу не вдалось розв’язати, якщо щось інше, то виводимо на екран відповіді.

Висновки

У результаті виконання роботи досягнуто таких результатів:

```
Розв'язок на мові Python

Оптимальний план випуску продукції:
P1 = 20800.0
P2 = 4800.0
P3 = 22600.0
Максимальний прибуток: 7320000.0



Розв'язок на мові Octave

Оптимальний план випуску продукції:
P1 = 20800.0
P2 = 4800.0
P3 = 22600.0
Максимальний прибуток: 7320000.0
```

Як бачимо, результати є ідентичними. Це чудово !

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

Додаток А – Код програми

Вміст файлу main.ipynb :

```
from scipy.optimize import linprog
from oct2py import octave
```

```
def umova():
    data = {'Список 1': [], 'Матриця 1': [], 'Список 2': []}
    current_list = None

    with open('data.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 1':
                    data[current_list].append(values)
                else:
                    data[current_list].extend(values)

    data['Матриця 1'] = [row for row in data['Матриця 1'] if row]

    return data['Список 1'], data['Матриця 1'], data['Список 2']
c, A, b = umova()
```

```
# Визначення обмежень для змінних (повинні бути невід'ємні)
x_bounds = [(0, None), (0, None), (0, None)]

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

if result.success:
    print('Розв\'язок на мові Python\n')
    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))
    print('Максимальний прибуток:', -round(result.fun, 2))
else:
    print('Задачу не вдалося вирішити.')
```

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

```
vartype = 'CCC'
ctype = 'UUUU'
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('Розв\'язок на мові Octave\n')
    print("Оптимальний план випуску продукції:")
    print(f"P1 = {x[0][0]}")
    print(f"P2 = {x[1][0]}")
    print(f"P3 = {x[2][0]}")
    print("Максимальний прибуток:", -fval)
```