Кременчуцький національний університет імені Михайла Остроградського

(повне найменування вищого навчального закладу)

Факультет електроніки та комп’ютерної інженерії

(повне найменування інституту, назва факультету)

Кафедра комп’ютерних та інформаційних систем

(повна назва кафедри )

**Розрахунково-графічна робота**

з навчальної дисципліни «Алгоритми і методи обчислень»,

модуль «Методи оптимізації»

Виконав: студент 2 курсу групи КІ-22-1

ступінь вищої освіти бакалавр

(бакалавр, магістр)

спеціальність 123 «Комп’ютерна інженерія»

(шифр і назва спеціальності)

освітня програма «Комп’ютерна інженерія»

Грищенко І. В.

(прізвище та ініціали)

Керівник Сидоренко В. М.

(прізвище та ініціали)

м. Кременчук 2023 року

**МОДУЛЬ 1**

**Практичне заняття №1. Асимптотична складність алгоритмів. 𝑶() – нотація**

**Завдання № 1:**

1. 𝑓(𝑛) = 5𝑛^2 + 1:

У даному виразі маємо лише одну залежну величину 𝑛, яка зводиться до степені 2. Коефіцієнт 5 також враховується. Таким чином, асимптотична складність у 𝑂()-нотації для даного виразу становить 𝑂(𝑛^2).

1. 𝑓(𝑛) = 7𝑛^2:

Аналогічно до першого виразу, тут також маємо лише одну залежну величину 𝑛, зведену до степені 2. Таким чином, асимптотична складність у 𝑂()-нотації для даного виразу становить 𝑂(𝑛^2).

1. log(𝑛) + 2𝑛^2 = 11:

У цьому виразі маємо дві залежні величини: 𝑛^2 та log(𝑛). Оскільки логарифмічна функція зазвичай зростає повільніше, ніж поліноміальна функція, то можемо припустити, що квадратична функція 𝑛^2 буде преобладати у цьому виразі. Таким чином, асимптотична складність у 𝑂()-нотації для даного виразу становить 𝑂(𝑛^2).

1. 3𝑛^2 + 10𝑛 − 6:

У цьому виразі також маємо лише одну залежну величину 𝑛, зведену до степені 2. Таким чином, асимптотична складність у 𝑂()-нотації для даного виразу становить 𝑂(𝑛^2).

**Отже, асимптотична складність у 𝑂()-нотації для всіх заданих виразів становить 𝑂(𝑛^2).**

**Завдання № 2:**

𝑓(𝑛) = 150𝑛^2 + 11 та 𝑔(𝑛) = 𝑛^2.

Для доведення, що 𝑓(𝑛) = 𝑂(𝑔(𝑛)), ми повинні знайти такі позитивні константи 𝑐 та 𝑛₀, для яких 𝑓(𝑛) ≤ 𝑐⋅𝑔(𝑛) для всіх 𝑛 ≥ 𝑛₀.  
Можемо вибрати 𝑐 = 151 і 𝑛₀ = 1.  
Тоді для всіх 𝑛 ≥ 1 маємо: 𝑓(𝑛) = 150𝑛^2 + 11 ≤ 151𝑛^2 = 𝑐⋅𝑔(𝑛).  
Отже, позитивні константи 𝑐 = 151 та 𝑛₀ = 1, для яких 𝑓(𝑛) ≤ 𝑐⋅𝑔(𝑛) для всіх   
𝑛 ≥ 𝑛₀.

**Це доводить, що 𝑓(𝑛) = 𝑂(𝑔(𝑛)).**

**Практичне заняття №2. Асимптотична складність алгоритмів. Інші нотації.**

**Завдання 1:**

lim(𝑛→∞) 𝑓(𝑛)/𝑔(𝑛)

= lim(𝑛→∞) (𝑛^3 + 2𝑛^2 − 5𝑛 + 8)/(𝑛^4)

Застосуємо правило Лопіталя, диференціюючи чисельник та знаменник окремо:

= lim(𝑛→∞) (3𝑛^2 + 4𝑛 − 5)/(4𝑛^3)

= lim(𝑛→∞) (6𝑛 + 4)/(12𝑛^2)

= lim(𝑛→∞) (6/12𝑛 + 4/12)/(1/12𝑛^2)

При 𝑛, що прямує до нескінченності, дріб (6/12𝑛 + 4/12)/(1/12𝑛^2) зводиться до нуля, оскільки степінь 𝑛 у знаменнику зростає швидше, ніж у чисельнику.

**Таким чином, границя дорівнює нулю, що означає, що 𝑓(𝑛) є O(𝑔(𝑛)).**

**Завдання 2:**

Щоб показати, що 𝑓(𝑛) є Θ(𝑔(𝑛)), ми повинні довести дві частини:

1. 𝑓(𝑛) є O(𝑔(𝑛)): Ми повинні знайти константу 𝑐₁ та натуральне число 𝑛₀, такі що для всіх натуральних чисел 𝑛 > 𝑛₀ виконується нерівність |𝑓(𝑛)| ≤ 𝑐₁|𝑔(𝑛)|.
2. 𝑓(𝑛) є Ω(𝑔(𝑛)): Ми повинні знайти константу 𝑐₂ та натуральне число 𝑛₀, такі що для всіх натуральних чисел 𝑛 > 𝑛₀ виконується нерівність |𝑓(𝑛)| ≥ 𝑐₂|𝑔(𝑛)|.

1. Застосуємо межу і обчислимо границю відношення 𝑓(𝑛)/𝑔(𝑛) при 𝑛, що прямує до нескінченності:  
lim(𝑛→∞) ((𝑛^4/8) - 12)/(𝑛^4)

lim(𝑛→∞) (1/8)

Отже, за означенням O-нотації, ми маємо |𝑓(𝑛)| ≤ (1/8)|𝑔(𝑛)| для всіх 𝑛 > 𝑛₀, де 𝑛₀ може бути будь-яким натуральним числом.

2. Знову обчислимо границю відношення 𝑓(𝑛)/𝑔(𝑛) при 𝑛, що прямує до нескінченності:

lim(𝑛→∞) ((𝑛^4/8) - 12)/(𝑛^4)

lim(𝑛→∞) (1/8)

При 𝑛, що прямує до нескінченності, перший дріб 1/8 залишається постійним. За означенням Ω-нотації, ми маємо |𝑓(𝑛)| ≥ (1/8)|𝑔(𝑛)| для всіх 𝑛 > 𝑛₀, де 𝑛₀ може бути будь-яким натуральним числом.

**Отже, доведено, що 𝑓(𝑛) є Θ(𝑔(𝑛)), оскільки одночасно виконуються обидві умови: 𝑓(𝑛) є O(𝑔(𝑛)) та 𝑓(𝑛) є Ω(𝑔(𝑛)).**

**Завдання 3:**

𝑓(𝑛) = 5(log2(𝑛))^2

𝑔(𝑛) = log2(𝑛)

Обчислимо границю відношення f(n)/g(n):

lim(𝑛→∞) (5(log2(𝑛))^2)/(log2(𝑛))

lim(𝑛→∞) (10log2(𝑛))/(1/𝑛)

lim(𝑛→∞) 10𝑛log2(𝑛)

Тепер застосуємо іншу межу, замінюючи log2(𝑛) на 𝑥:

lim(𝑥→∞) 10𝑥

При 𝑥, що прямує до нескінченності, вираз 10𝑥 зростає нескінченно.

Отже, отримуємо:

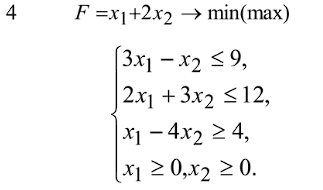
lim(𝑛→∞) 𝑓(𝑛)/𝑔(𝑛) = ∞

**Це означає, що для доведення 𝑓(𝑛) є Ω(𝑔(𝑛)), ми можемо вибрати будь-яку константу 𝑐 > 0 і знайти таке натуральне число 𝑛₀, що |𝑓(𝑛)| ≥ 𝑐|𝑔(𝑛)| для всіх 𝑛 > 𝑛₀.**

**Отже, ми довели, що 𝑓(𝑛) є Ω(𝑔(𝑛)).**

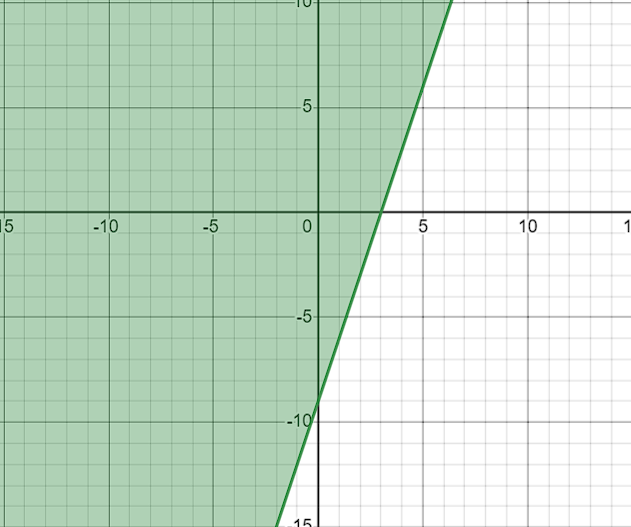
**МОДУЛЬ 2**

**Практичне завдання № 1. Геометричний метод розв'язку задачі лінійного програмування**

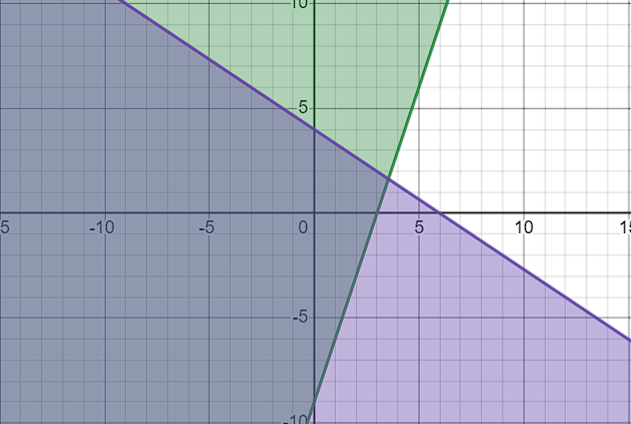


Необхідно знайти мінімальне значення цільової функції F = x1+2x2+5 → min, за системою обмежень:  
**1)** 3x1-x2≤9, (1)

1. Знаходимо його перетин з віссю X1, коли X2 = 0: 3X1 - 0 = 9 => X1 = 3.
2. Знаходимо його перетин з віссю X2, коли X1 = 0: 3(0) - X2 = 9 => X2 = -9.
3. З'єднаємо ці дві точки прямою.

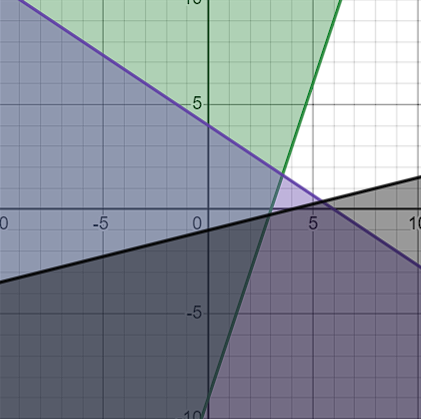
  
**2)** 2x1+3x2≤12, (2)

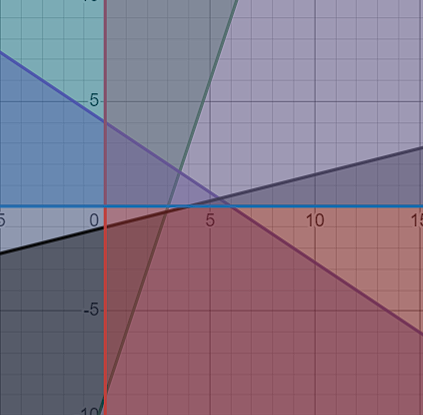
1. Знаходимо його перетин з віссю X1, коли X2 = 0: 2X1 + 3(0) = 12 => X1 = 6.
2. Знаходимо його перетин з віссю X2, коли X1 = 0: 2(0) + 3X2 = 12 => X2 = 4.
3. З'єднаємо ці дві точки прямою.



**3)** x1-4x2≥4, (3)

1. Знаходимо його перетин з віссю X1, коли X2 = 0: X1 - 4(0) = 4 => X1 = 4.
2. Знаходимо його перетин з віссю X2, коли X1 = 0: 0 - 4X2 = 4 => X2 = -1.
3. З'єднаємо ці дві точки прямою.

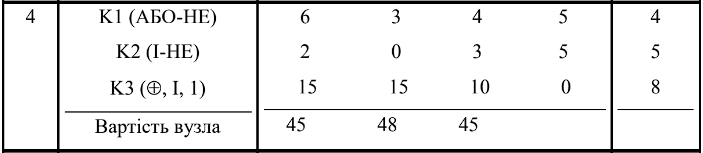
  
x1≥0, (4)  
x2≥0, (5)

  
Побудуємо область допустимих рішень, тобто графічно розв'яжемо систему нерівностей. Для цього побудуємо кожну пряму і визначимо напівплощини, задані нерівностями (напівплощини позначаються штрихом).

Проблема не має валідних рішень. ОДЗ - це порожній набір.

**Практичне заняття №2. Симплекс-метод розв'язання задачі лінійного програмування**

****



45·x1 + 48·x2 + 45·x3 → max  
6·x1 + 3·x2 + 4·x3 + 5·x4 ≤ 4  
2·x1 + 3·x3 + 4·x4 ≤ 5  
15·x1 + 15·x2 + 10·x3 ≤ 8

Перепишемо обмеження у канонічному вигляді:

6·x1 + 3·x2 + 4·x3 + 5·x4 + x5 = 4  
2·x1 + 3·x3 + 4·x4 + x6 = 5  
15·x1 + 15·x2 + 10·x3 + x7 = 8

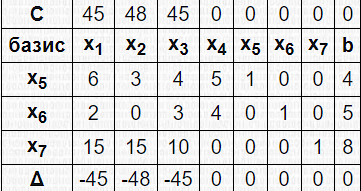
Початкова симплекс-таблиця:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С | 45 | 48 | 45 | 0 | 0 | 0 | 0 | 0 |
| базис | Х1 | Х2 | Х3 | Х4 | Х5 | Х6 | Х7 | b |
| Х5 | 6 | 3 | 4 | 5 | 1 | 0 | 0 | 4 |
| Х6 | 2 | 0 | 3 | 4 | 0 | 1 | 0 | 5 |
| Х7 | 15 | 15 | 0 | 0 | 0 | 0 | 1 | 8 |

Шукаємо дельти:

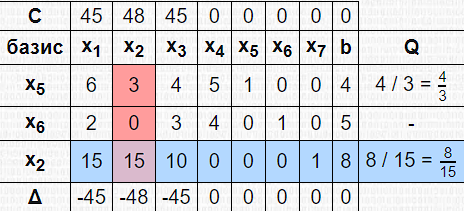
Δ1 = C5·a11 + C6·a21 + C7·a31 - C1 = 0·6 + 0·2 + 0·15 - 45 = -45  
Δ2 = C5·a12 + C6·a22 + C7·a32 - C2 = 0·3 + 0·0 + 0·15 - 48 = -48  
Δ3 = C5·a13 + C6·a23 + C7·a33 - C3 = 0·4 + 0·3 + 0·10 - 45 = -45  
Δ4 = C5·a14 + C6·a24 + C7·a34 - C4 = 0·5 + 0·4 + 0·0 - 0 = 0  
Δ5 = C5·a15 + C6·a25 + C7·a35 - C5 = 0·1 + 0·0 + 0·0 - 0 = 0  
Δ6 = C5·a16 + C6·a26 + C7·a36 - C6 = 0·0 + 0·1 + 0·0 - 0 = 0  
Δ7 = C5·a17 + C6·a27 + C7·a37 - C7 = 0·0 + 0·0 + 0·1 - 0 = 0  
Δb = C5·b1 + C6·b2 + C7·b3 - C8 = 0·4 + 0·5 + 0·8 - 0 = 0

Симплекс-таблиця з дельтами:

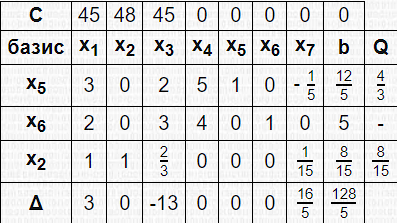


План не оптимальний.

Ітерація 1:

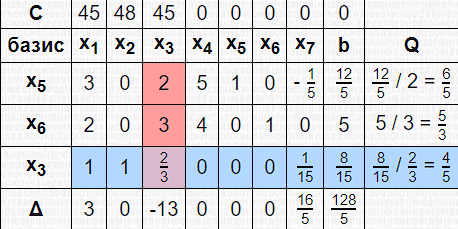


Симплекс-таблиця з оновленими дельтами:

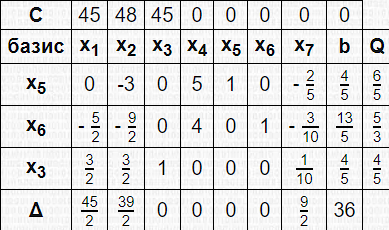


План не є оптимальним, оскільки Δ3 = -13 негативний.

Ітерація 2:



Симплекс-таблиця з оновленими дельтами:



Поточний план Х: [0, 0, 4/5, 0, 4/5, 13/5, 0]

Цільова ф-ція: 45\*0 + 48\*0 + 45\*4/5 + 0\*0 + 0\*4/5 + 0\*13/5 + 0\*0 = 36;

Відповідь: х1 = 0, х2 = 0, х3 = 4/5, х4 = 0, F = 36.

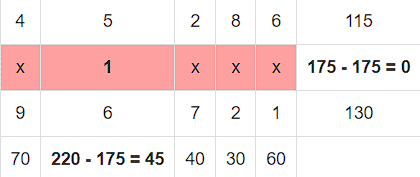
**Практична робота № 3**

**Тема. Транспортна задача**

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | B1 | B2 | | B3 | B4 | B5 | Запасы |
| A1 | 4 | 5 | | 2 | 8 | 6 | 115 |
| A2 | 3 | 1 | | 9 | 7 | 3 | 175 |
| A3 | 9 | 6 | | 7 | 2 | 1 | 130 |
| Потребности | 70 | 220 | | 40 | 30 | 60 |  |  |

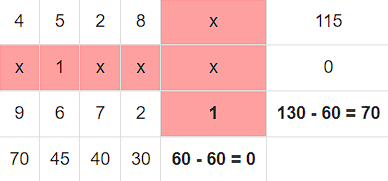
Потрібний елемент c22=1. Для цього елемента запасів 175, вимог - 220. Оскільки мінімум дорівнює 175, віднімемо його.

x22 = min(175,220) = 175.



Потрібний елемент c35=1. Для цього елемента запасів 130, вимог - 60. Оскільки мінімум дорівнює 60, віднімемо його.

х35 = min(130,60) = 60.



Потрібний елемент c13=2. Для цього елемента запасів 115, вимог - 40. Оскільки мінімум дорівнює 40, віднімемо його.

х13 = min(115,40) = 40.

|  |
| --- |
|  |
|  |
| 4 | 5 | **2** | 8 | x | **115 - 40 = 75** |
| x | 1 | x | x | x | 0 |
| 9 | 6 | x | 2 | 1 | 70 |
| 70 | 45 | **40 - 40 = 0** | 30 | 0 |  |

Потрібний елемент c34=2. Для цього елемента запасів 70, вимог - 30. Оскільки мінімум дорівнює 30, віднімемо його.

х34 = min(70,30) = 30.

|  |
| --- |
|  |
| 4 | 5 | 2 | x | x | 75 |
| x | 1 | x | x | x | 0 |
| 9 | 6 | x | **2** | 1 | **70 - 30 = 40** |
| 70 | 45 | 0 | **30 - 30 = 0** | 0 |  |

Потрібний елемент c11=4. Для цього елемента запасів 75, вимог - 70. Оскільки мінімум дорівнює 70, віднімемо його.

х11 = min(75,70) = 70.

|  |
| --- |
|  |
| **4** | 5 | 2 | x | x | **75 - 70 = 5** |
| x | 1 | x | x | x | 0 |
| x | 6 | x | 2 | 1 | 40 |
| **70 - 70 = 0** | 45 | 0 | 0 | 0 |  |

Потрібний елемент c12=5. Для цього елемента запасів 5, вимог - 45. Оскільки мінімум дорівнює 5, віднімемо його.

х12 = min(5,45) = 5.

|  |
| --- |
|  |
| 4 | **5** | 2 | x | x | **5 - 5 = 0** |
| x | 1 | x | x | x | 0 |
| x | 6 | x | 2 | 1 | 40 |
| 0 | **45 - 5 = 40** | 0 | 0 | 0 |  |

Потрібний елемент c32=6. Для цього елемента запасів 40, вимог - 40. Оскільки мінімум дорівнює 40, віднімемо його.

х32 = min(40,40) = 40.

|  |  |  |
| --- | --- | --- |
|  |  | |
| 4 | 5 | | | 2 | x | x |  | 0 |
| x | 1 | | | x | x | x |  | 0 |
| x | **6** | | | x | 2 | 1 |  | **40 - 40 = 0** |
| 0 | **40 - 40 = 0** | | | 0 | 0 | 0 |  |  |
|  |  | |
|  | B1 | B2 | | B3 | B4 | B5 | Запасы |
| A1 | 4[70] | 5[5] | | 2[40] | 8 | 6 | 115 |
| A2 | 3 | 1[175] | | 9 | 7 | 3 | 175 |
| A3 | 9 | 6[40] | | 7 | 2[30] | 1[60] | 130 |
| Потребности | 70 | 220 | | 40 | 30 | 60 |  |  |

В результаті був отриманий перший довідковий план, що допустимо, так як всі товари з баз зняті, потреби магазинів задоволені, а план відповідає системі обмежень транспортного завдання.

2. Порахуємо кількість займаних клітинок таблиці, їх 7, і воно повинно бути m + n - 1 = 7. Тому опорний план є невиродженим.

Значення цільової функції для цієї базової лінії має такий вигляд:

F(x) = 4\*70 + 5\*5 + 2\*40 + 1\*175 + 6\*40 + 2\*30 + 1\*60 = 920

u1 + v1 = 4; 0 + v1 = 4; v1 = 4  
u1 + v2 = 5; 0 + v2 = 5; v2 = 5  
u2 + v2 = 1; 5 + u2 = 1; u2 = -4  
u3 + v2 = 6; 5 + u3 = 6; u3 = 1  
u3 + v4 = 2; 1 + v4 = 2; v4 = 1  
u3 + v5 = 1; 1 + v5 = 1; v5 = 0  
u1 + v3 = 2; 0 + v3 = 2; v3 = 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | v1=4 | v2=5 | v3=2 | v4=1 | v5=0 |
| u1=0 | 4[70] | 5[5] | 2[40] | 8 | 6 |
| u2=-4 | 3 | 1[175] | 9 | 7 | 3 |
| u3=1 | 9 | 6[40] | 7 | 2[30] | 1[60] |

Довідковий план оптимальний, тому всі оцінки вільних осередків задовольняють умові ui + vj ≤ cij.

Мінімальні витрати складуть:

F(x) = 4\*70 + 5\*5 + 2\*40 + 1\*175 + 6\*40 + 2\*30 + 1\*60 = 920

**Практична робота № 4**

**Тема. Задача вибору або задача про призначення**

**ВАРІАНТ 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 5 | 3 | 4 | 7 |
| 6 | 1 | 2 | 5 | 3 |
| 2 | 2 | 4 | 1 | 4 |
| 5 | 7 | 5 | 1 | 6 |
| 5 | 9 | 3 | 2 | 7 |

Cij =

Змінили матрицю, помноживши всі елементи на (-1), а потім додавши їх до максимального елемента матриці (9), щоб матриця не містила негативних елементів:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 4 | 6 | 5 | 2 |
| 3 | 8 | 7 | 4 | 6 |
| 7 | 7 | 5 | 8 | 5 |
| 4 | 2 | 4 | 8 | 3 |
| 4 | 0 | 6 | 7 | 2 |

1. Проводимо скорочення матриці по рядках. У зв'язку з цим в знову отриманої матриці в кожному рядку буде як мінімум один нуль.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 2 | 4 | 3 | 0 | 2 |
| 0 | 5 | 4 | 1 | 3 | 3 |
| 2 | 2 | 0 | 3 | 0 | 5 |
| 2 | 0 | 2 | 6 | 1 | 2 |
| 4 | 0 | 6 | 7 | 2 | 0 |

Потім таку ж операцію редукції проводимо над колонками, для чого знаходимо мінімальний елемент в кожній колонці.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 2 | 4 | 2 | 0 |
| 0 | 5 | 4 | 0 | 3 |
| 2 | 2 | 0 | 2 | 0 |
| 2 | 0 | 2 | 5 | 1 |
| 4 | 0 | 6 | 6 | 2 |
| 0 | 0 | 0 | 1 | 0 |

Віднявши мінімальні елементи, отримаємо повністю приведену матрицю.

2. Методом проб і помилок шукаємо прийнятне рішення, для якого всі призначення мають нульову вартість.

Зафіксуйте нульове значення в осередку (1, 5). Інші нулі в рядку 1 і стовпці 5 закреслюються. Для цієї комірки викресліть нулі в клітинках (3; 5)  
Зафіксуйте нульове значення в осередку (2, 4). Інші нулі в рядку 2 і стовпці 4 закреслюються. Для цієї комірки викресліть нулі в осередках (2; 1).

Зафіксуйте нульове значення в осередку (3, 3). Інші нулі в рядку 3 і стовпці 3 закреслюються.

Зафіксуйте нульове значення в осередку (4, 2). Інші нулі в рядку 4 і стовпці 2 закреслюються. Для цієї комірки викресліть нулі в клітинках (5; 2)

В результаті отримаємо наступну матрицю:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 2 | 4 | 2 | [0] |
| [-0-] | 5 | 4 | [0] | 3 |
| 2 | 2 | [0] | 2 | [-0-] |
| 2 | [0] | 2 | 5 | 1 |
| 4 | [-0-] | 6 | 6 | 2 |

Оскільки розташування нульових елементів в матриці не дозволяє сформувати систему з 5 незалежних нулів (в матриці їх всього 4), рішення неприпустимо.

3. Проводимо модифікацію матриці. Викресліть рядки і стовпці з якомога більшою кількістю нульових елементів:

Рядок 2, Стовпець 2, Рядок 3, Стовпець 5

Отримуємо скорочену матрицю (елементи виділяються):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **5** | 2 | **4** | **2** | 0 |
| 0 | 5 | 4 | 0 | 3 |
| 2 | 2 | 0 | 2 | 0 |
| **2** | 0 | **2** | **5** | 1 |
| **4** | 0 | **6** | **6** | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3** | 2 | **2** | **0** | 0 |
| 0 | 5 | 4 | 0 | 3 |
| 2 | 2 | 0 | 2 | 0 |
| **0** | 0 | **0** | **3** | 1 |
| **2** | 0 | **4** | **4** | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3** | 2 | **2** | **0** | 0 |
| 0 | 5 | 4 | 0 | 3 |
| 2 | 2 | 0 | 2 | 0 |
| **0** | 0 | **0** | **3** | 1 |
| **2** | 0 | **4** | **4** | 2 |

Ми шукаємо прийнятне рішення, для якого всі завдання мають нульову вартість.

Зафіксуйте нульове значення в осередку (1, 5). Інші нулі в рядку 1 і стовпці 5 закреслюються. Для цієї комірки викресліть нулі в осередках (1; 4).

Зафіксуйте нульове значення в осередку (2, 4). Інші нулі в рядку 2 і стовпці 4 закреслюються. Для цієї комірки викресліть нулі в осередках (2; 1).

Зафіксуйте нульове значення в осередку (3, 3).

Інші нулі в рядку 3 і стовпці 3 закреслюються. Для цієї комірки викресліть нулі в клітинках (4; 3)

Зафіксуйте нульове значення в осередку (4, 1). Інші нулі в рядку 4 і стовпці 1 закреслюються. Для цієї комірки закресліть нулі в осередках (4; 2).

Зафіксуйте нульове значення в осередку (5, 2). Інші нулі в рядку 5 і стовпці 2 закреслюються.

В результаті отримаємо наступну матрицю:

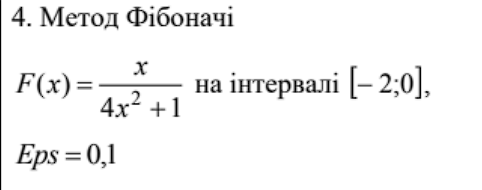
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 2 | 2 | [-0-] | **[0]** |
| [-0-] | 7 | 4 | **[0]** | 5 |
| 2 | 4 | **[0]** | 2 | 2 |
| **[0]** | [-0-] | [-0-] | 3 | 1 |
| 2 | **[0]** | 4 | 4 | 2 |

**Cmax = 5 + 7 + 4 + 5 + 9 = 30  
Путь: (2;4), (1;5), (3;3), (4;1), (5;2)**

**Практичне заняття № 5**

**Тема. Методи одновимірної пошукової оптимізації**

**ВАРІАНТ 4**



Знайдемо стаціонарні точки функції F(x) = x/(4x^2 + 1) шляхом обчислення похідної та встановлення її рівнянням нуль:

F'(x) = (1\*(4x^2 + 1) - x\*(8x))/(4x^2 + 1)^2 = (4x^2 + 1 - 8x^2)/(4x^2 + 1)^2 = (3 - 4x^2)/(4x^2 + 1)^2

Тепер прирівняємо похідну до нуля та вирішимо рівняння:

(3 - 4x^2)/(4x^2 + 1)^2 = 0

3 - 4x^2 = 0

4x^2 = 3

x^2 = 3/4

x = ±√(3/4)

Таким чином, стаціонарні точки функції F(x) знаходяться при x = √(3/4) та x = -√(3/4).

Ці значення x є потенційними точками максимуму або мінімуму функції F(x). Щоб визначити, яка з цих точок є максимумом, а яка - мінімумом, потрібно дослідити поведінку функції навколо цих точок, включаючи відомі значення функції на кінцях інтервалу [ - 2; 0].

Застосуємо метод Фібоначчі для встановлення точки максимуму або мінімуму на інтервалі [ - 2; 0] з вказаним Eps = 0,1.

Для застосування методу Фібоначчі для знаходження точки максимуму або мінімуму на інтервалі [-2; 0] з вказаним Eps = 0,1, спочатку потрібно визначити кількість ітерацій методу на основі значень Eps і довжини вхідного інтервалу.

Знаходження кількості ітерацій: Кількість ітерацій, N, визначається за формулою: N = (log(Δ) - log(Eps)) / log(φ), де Δ - довжина вхідного інтервалу, Eps - задана точність, φ - число Фібоначчі (приблизно 1.61803).

У нашому випадку: Δ = 2 - (-2) = 4.

Підставляючи ці значення в формулу, ми отримуємо: N = (log(4) - log(0.1)) / log(1.61803) ≈ 10.7477.

Оскільки N повинно бути цілим числом, ми округлюємо його до 11.

Тепер ми можемо застосувати метод Фібоначчі з 11 ітераціями.

Оскільки наша функція є монотонно спадною на цьому інтервалі, ми шукаємо максимум.

Застосуємо формулу для методу Фібоначчі:

φ = (1 + sqrt(5)) / 2 # число Фібоначчі L = b - a # довжина вхідного інтервалу

x1 = a + (Fib[N-M+2] / Fib[N-M+3]) \* L x2 = a + (Fib[N-M+1] / Fib[N-M+3]) \* L

Підставимо значення:

a = -2 b = 0 N = 11 Eps = 0.1

φ = (1 + sqrt(5)) / 2 L = 0 - (-2) = 2

x1 = -2 + (Fib[11-2+2] / Fib[11-2+3]) \* 2 x2 = -2 + (Fib[11-2+1] / Fib[11-2+3]) \* 2

Тепер виконаємо обчислення:

φ = 1.618033988749895 L = 2

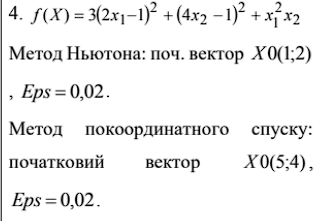
x1 = -2 + (Fib[11-2+2] / Fib[11-2+3]) \* 2 = -2 + (34 / 55) \* 2 ≈ -0.509090909090909

x2 = -2 + (Fib[11-2+1] / Fib[11-2+3]) \* 2 = -2 + (21 / 55) \* 2 ≈ -1.054545454545454

Таким чином, отримуємо x1 ≈ -0.50909 та x2 ≈ -1.05454. Значення, яке ближче до максимуму, є x2 ≈ -1.05454. Тому точка максимуму функції F(x) на інтервалі [-2; 0] з точністю Eps = 0.1 дорівнює x2 ≈ -1.05455.

**Практична робота № 6**

**Тема. Нелінійна багатовимірна безумовна оптимізація**



**Метод Ньютона:**

1. Обчислити градієнт функції: df/dx1 = 12x1 - 6 + 2x2 df/dx2 = 8x2 - 4 + x1^2
2. Обчислити гессіан функції: d^2f/dx1^2 = 12 + 2x2 d^2f/dx1dx2 = 2x1 d^2f/dx2dx1 = 2x1 d^2f/dx2^2 = 8
3. Задати початкову точку X0 = (1, 2)
4. Обчислити значення функції F(X0)
5. Повторювати наступні кроки, доки не досягнута необхідна точність Eps: a. Обчислити обернений гессіан: H^-1 = [[d^2f/dx1^2, d^2f/dx1dx2], [d^2f/dx2dx1, d^2f/dx2^2]]^-1 b. Обчислити вектор зміни X: delta\_X = -H^-1 \* gradient(X) c. Оновити значення X: X = X + delta\_X d. Обчислити значення функції F(X)
6. Вивести оптимальну точку X та мінімальне значення функції F(X)
7. import numpy as np
8. from scipy.optimize import minimize
9. # Функція
10. def objective(X):
11. x1, x2 = X
12. return 3\*(2\*x1 - 1)\*\*2 + (4\*x2 - 1)\*\*2 + x1\*\*2 \* x2
13. # Градієнт функції
14. def gradient(X):
15. x1, x2 = X
16. df\_dx1 = 12\*x1 - 6 + 2\*x2
17. df\_dx2 = 8\*x2 - 4 + x1\*\*2
18. return np.array([df\_dx1, df\_dx2])
19. # Гессіан функції
20. def hessian(X):
21. x1, x2 = X
22. h11 = 12 + 2\*x2
23. h12 = 2\*x1
24. h21 = 2\*x1
25. h22 = 8
26. return np.array([[h11, h12], [h21, h22]])
27. # Початковий вектор
28. X0 = np.array([1, 2])
29. # Розв'язання задачі мінімізації
30. result = minimize(objective, X0, method='Newton-CG', jac=gradient, hess=hessian, tol=0.02)
31. # Виведення результату
32. print('Метод Ньютона:')
33. print('Оптимальний вектор X:', result.x)

print('Мінімальне значення функції:', result.fun)

**Метод покоординатного спуску:**

1. Задати початкову точку X0 = (5, 4)
2. Обчислити значення функції F(X0)
3. Повторювати наступні кроки, доки не досягнута необхідна точність Eps: a. Оновити значення x1: x1 = x1 - learning\_rate \* (dF/dx1) b. Оновити значення x2: x2 = x2 - learning\_rate \* (dF/dx2) c. Обчислити значення функції F(X)
4. Вивести оптимальну точку X та мінімальне значення функції F(X)
5. from scipy.optimize import minimize
6. # Функція
7. def objective(X):
8. x1, x2 = X
9. return 3\*(2\*x1 - 1)\*\*2 + (4\*x2 - 1)\*\*2 + x1\*\*2 \* x2
10. # Початковий вектор
11. X0 = np.array([5, 4])
12. # Розв'язання задачі мінімізації
13. result = minimize(objective, X0, method='BFGS', tol=0.02)
14. # Виведення результату
15. print('Метод покоординатного спуску:')
16. print('Оптимальний вектор X:', result.x)

print('Мінімальне значення функції:', result.fun)

В обох методах Eps використовується для визначення точності досягнення оптимальної точки. Можна обчислити значення функції та оновлювати значення X за допомогою цих алгоритмів до тих пір, поки не досягнете заданої точності Eps.