

**Міністерство освіти України**  
**Національний технічний університет України**  
**“Київський політехнічний інститут імені Ігоря Сікорського”**  
**Факультет інформатики та обчислювальної техніки**

**ЗВІТ**

до практикуму № 5

з дисципліни

“Інформаційні технології підтримки прийняття рішень”

на тему: “Методи TOPSIS, VIKOR”

Варіант 11

Виконала:

Студентка групи ІІІ-71

Каспрук Анастасія Андріївна

Київ 2021

## **1. Результати.**

**(Нумерація альтернатив починається з нуля).**

### **Частина 1.**

**А) Найкраща альтернатива – 14.**

**Б) Найкраща альтернатива – 14.**

### **Частина 2.**

**2.1. Компромісний розв'язок – 14.**

**2.2. Для всіх значень  $\vartheta$   $[0; 1]$  компромісний розв'язок – 14.**

## 2. Постановка задачі.

### Завдання

Задано множину альтернатив  $\{A_1, \dots, A_{15}\}$ , що оцінені за критеріями  $k_1 \dots k_{12}$ . Оцінки альтернатив за усіма критеріями дано в таблиці.

Кожен критерій має ваговий коефіцієнт  $w_i$  (Таблиця із вхідними даними співпадає з даними для практикуму 4)

В методах TOPSIS і VIKOR застосовуються нормовані вагові коефіцієнти критеріїв, тому потрібно виконати нормування цих коефіцієнтів (згідно співвідношення (3) в описі методу VIKOR)

### Необхідно:

#### Частина 1.

За допомогою метода TOPSIS встановити ранжування на множині альтернатив (згідно спадання важливості) та вибрати найкращу альтернативу для випадків:

А) всі критерії потрібно максимізувати;

Б) критерії  $k_1-k_7$  підлягають максимізації, а критерії  $k_8-k_{12}$  - мінімізації (тобто  $k_1-k_7 \in K^+$ , а  $k_8-k_{12} \in K^-$ ).

#### Частина 2.

1. За допомогою метода VIKOR встановити ранжування на множині альтернатив (згідно спаданню важливості) та вибрати найкращу альтернативу (компромісний розв'язок), якщо параметр  $\vartheta=0,5$ . Вважати, що всі критерії потрібно максимізувати.
2. Дослідити в методі VIKOR вплив зміни параметра  $\vartheta$  на результуючий компромісний розв'язок. Дослідити випадки  $0 \leq \vartheta \leq 1$  з кроком 0,1. В кожному випадку обґрунтувати результат – ранжування і компромісний розв'язок.

Навести **Висновки** проведеного аналізу, а також виконати порівняння обох методів за стратегією та результатами.

### Завдання для варіанту 11:

Таблиця оцінок альтернатив  $A_1-A_{15}$  за критеріями  $k_1-k_{12}$

3	8	9	7	7	10	7	4	5	4	2	6
3	6	7	4	6	9	9	10	7	3	3	4
5	3	3	3	10	10	1	9	7	6	1	1
2	3	4	9	7	7	2	4	10	10	3	2
1	8	1	10	4	7	6	1	1	6	3	2
8	7	4	6	6	2	1	10	6	10	3	8
10	8	6	5	5	3	1	9	9	1	5	6
6	8	3	1	1	4	10	5	2	3	5	3
5	1	8	8	5	2	5	6	9	8	8	4
5	3	4	3	10	2	4	6	9	7	4	2
4	2	5	3	7	7	6	5	1	5	7	3
4	1	4	8	2	8	2	7	3	7	6	4
4	10	2	3	1	5	3	9	1	10	4	2
4	8	6	5	2	6	2	8	2	6	3	8
4	9	7	9	7	9	2	9	8	10	2	6

Вагові коефіцієнти критеріїв  $k_1-k_{12}$ :

1 10 5 2 2 6 2 5 8 2 5 8

### 3. Розв'язок.

#### Частина 1 А).

На наступному скріншоті відображено поетапний розв'язок задачі багатокритеріальної оптимізації методом TOPSIS для випадку, коли всі критерії потрібно максимізувати.

```
Microsoft Visual Studio Debug Console

Поетапний розв'язок задачі методом TOPSIS

Нормалізовані оцінки альтернатив:
0 0.154 0.322 0.436 0.291 0.300 0.386 0.363 0.142 0.207 0.148 0.118 0.336
1 0.154 0.241 0.339 0.166 0.257 0.347 0.467 0.355 0.289 0.111 0.178 0.224
2 0.257 0.121 0.145 0.125 0.429 0.386 0.052 0.320 0.289 0.222 0.059 0.056
3 0.103 0.121 0.194 0.374 0.300 0.270 0.104 0.142 0.413 0.370 0.178 0.112
4 0.051 0.322 0.048 0.416 0.171 0.270 0.312 0.036 0.041 0.222 0.178 0.112
5 0.411 0.281 0.194 0.250 0.257 0.077 0.052 0.355 0.248 0.370 0.178 0.448
6 0.514 0.322 0.290 0.208 0.214 0.116 0.052 0.320 0.372 0.037 0.296 0.336
7 0.309 0.322 0.145 0.042 0.043 0.154 0.519 0.178 0.083 0.111 0.296 0.168
8 0.257 0.040 0.387 0.333 0.214 0.077 0.260 0.213 0.372 0.296 0.474 0.224
9 0.257 0.121 0.194 0.125 0.429 0.077 0.208 0.213 0.372 0.259 0.237 0.112
10 0.206 0.080 0.242 0.125 0.300 0.270 0.312 0.178 0.041 0.185 0.415 0.168
11 0.206 0.040 0.194 0.333 0.086 0.309 0.104 0.249 0.124 0.259 0.355 0.224
12 0.206 0.402 0.097 0.125 0.043 0.193 0.156 0.320 0.041 0.370 0.237 0.112
13 0.206 0.322 0.290 0.208 0.086 0.232 0.104 0.284 0.083 0.222 0.178 0.448
14 0.206 0.362 0.339 0.374 0.300 0.347 0.104 0.320 0.330 0.370 0.118 0.336

Зважені нормалізовані оцінки альтернатив:
0 0.003 0.057 0.039 0.010 0.011 0.041 0.013 0.013 0.030 0.005 0.011 0.048
1 0.003 0.043 0.030 0.006 0.009 0.037 0.017 0.032 0.041 0.004 0.016 0.032
2 0.005 0.022 0.013 0.004 0.015 0.041 0.002 0.029 0.041 0.008 0.005 0.008
3 0.002 0.022 0.017 0.013 0.011 0.029 0.004 0.013 0.059 0.013 0.016 0.016
4 0.001 0.057 0.004 0.015 0.006 0.029 0.011 0.003 0.006 0.008 0.016 0.016
5 0.007 0.050 0.017 0.009 0.009 0.008 0.002 0.032 0.035 0.013 0.016 0.064
6 0.009 0.057 0.026 0.007 0.008 0.012 0.002 0.029 0.053 0.001 0.026 0.048
7 0.006 0.057 0.013 0.001 0.002 0.017 0.019 0.016 0.012 0.004 0.026 0.024
8 0.005 0.007 0.035 0.012 0.008 0.008 0.009 0.019 0.053 0.011 0.042 0.032
9 0.005 0.022 0.017 0.004 0.015 0.008 0.007 0.019 0.053 0.009 0.021 0.016
10 0.004 0.014 0.022 0.004 0.011 0.029 0.011 0.016 0.006 0.007 0.037 0.024
11 0.004 0.007 0.017 0.012 0.003 0.033 0.004 0.022 0.018 0.009 0.032 0.032
12 0.004 0.072 0.009 0.004 0.002 0.021 0.006 0.029 0.006 0.013 0.021 0.016
13 0.004 0.057 0.026 0.007 0.003 0.025 0.004 0.025 0.012 0.008 0.016 0.064
14 0.004 0.065 0.030 0.013 0.011 0.037 0.004 0.029 0.047 0.013 0.011 0.048

PIS <утопічна точка>: 0.009 0.072 0.039 0.015 0.015 0.041 0.019 0.032 0.059 0.013 0.042 0.064
NIS <антиутопічна точка>: 0.001 0.007 0.004 0.001 0.002 0.008 0.002 0.003 0.006 0.001 0.005 0.008
D+ <відстань до PIS>: 0.054 0.057 0.092 0.083 0.092 0.060 0.048 0.079 0.082 0.085 0.093 0.090 0.086 0.064 0.043
D- <відстань до NIS>: 0.086 0.077 0.059 0.065 0.059 0.085 0.090 0.062 0.075 0.059 0.049 0.052 0.074 0.084 0.096

C <наближеність до PIS>:
<альтернатива: 14, наближеність: 0.691>
<альтернатива: 6, наближеність: 0.652>
<альтернатива: 0, наближеність: 0.616>
<альтернатива: 5, наближеність: 0.585>
<альтернатива: 1, наближеність: 0.575>
<альтернатива: 13, наближеність: 0.569>
<альтернатива: 8, наближеність: 0.479>
<альтернатива: 12, наближеність: 0.462>
<альтернатива: 3, наближеність: 0.442>
<альтернатива: 7, наближеність: 0.440>
<альтернатива: 9, наближеність: 0.408>
<альтернатива: 2, наближеність: 0.392>
<альтернатива: 4, наближеність: 0.390>
<альтернатива: 11, наближеність: 0.368>
<альтернатива: 10, наближеність: 0.345>

Найкраща альтернатива: 14
```

Спочатку були обчислені нормалізовані оцінки альтернатив за наступним співвідношенням.

$$r_{kj}(x) = \frac{x_{kj}}{\sqrt{\sum_{k=1}^n x_{kj}^2}}, \quad k = 1, \dots, n; j = 1, \dots, m$$

Потім кожна нормалізована оцінка була домножена на вагу відповідного критерію.

$$v_{kj}(x) = w_j r_{kj}(x), k = 1, \dots, n; j = 1, \dots, m$$

Далі були обчислені координати до утопічної та антиутопічної точок PIS та NIS.

$$PIS = A^+ = \{v_1^+(x), v_2^+(x), \dots, v_j^+(x), \dots, v_m^+(x)\} \quad (5)$$

$$= \left\{ \left( \max_k v_{kj}(x) \mid j \in K^+ \right), \left( \min_k v_{kj}(x) \mid j \in K^- \right) \mid k = 1, \dots, n \right\}$$

$$NIS = A^- = \{v_1^-(x), v_2^-(x), \dots, v_j^-(x), \dots, v_m^-(x)\} \quad (6)$$

$$= \left\{ \left( \min_k v_{kj}(x) \mid j \in K^+ \right), \left( \max_k v_{kj}(x) \mid j \in K^- \right) \mid k = 1, \dots, n \right\}$$

Наступним кроком стало обчислення відстаней кожної альтернативи до позитивної ідеальної точки PIS та негативної ідеальної точки NIS.

$$D_k^* = \sqrt{\sum_{j=1}^m [v_{kj}(x) - v_j^+(x)]^2}, \quad k = 1, \dots, n$$

$$D_k^- = \sqrt{\sum_{j=1}^m [v_{kj}(x) - v_j^-(x)]^2}, \quad k = 1, \dots, n$$

На базі значень  $D_k^*$  та  $D_k^-$  були встановлені наближеності кожної альтернативи до позитивної ідеальної точки PIS.

$$C_k^* = D_k^- / (D_k^* + D_k^-), \quad k = 1, \dots, n$$

$$\text{де } C_k^* \in [0, 1] \quad \forall k = 1, \dots, n$$

Впорядкувавши значення  $C_k^*$  у порядку спадання, на першій позиції у множині отримали найбільш наближену до позитивної ідеальної точки PIS (тобто найкращу) альтернативу.

### **Частина 1 Б).**

На наступному скріншоті відображено поетапний розв'язок задачі багатокритеріальної оптимізації методом TOPSIS для випадку, коли критерії k1-k7 підлягають максимізації, а критерії k8-k12 - мінімізації (тобто k1-k7  $\in K^+$ , а k8-k12  $\in K^-$ ).

```

Microsoft Visual Studio Debug Console

Поетапний розв'язок задачі методом TOPSIS

Нормалізовані оцінки альтернатив:
0 0.222 0.778 1.000 0.667 0.667 1.000 0.667 0.667 0.556 0.667 0.857 0.286
1 0.222 0.556 0.750 0.333 0.556 0.875 0.889 0.000 0.333 0.778 0.714 0.571
2 0.444 0.222 0.250 0.222 1.000 1.000 0.000 0.111 0.333 0.444 1.000 1.000
3 0.111 0.222 0.375 0.889 0.667 0.625 0.111 0.667 0.000 0.000 0.714 0.857
4 0.000 0.778 0.000 1.000 0.333 0.625 0.556 1.000 1.000 0.444 0.714 0.857
5 0.778 0.667 0.375 0.556 0.556 0.000 0.000 0.000 0.444 0.000 0.714 0.000
6 1.000 0.778 0.625 0.444 0.444 0.125 0.000 0.111 0.111 1.000 0.429 0.286
7 0.556 0.778 0.250 0.000 0.000 0.250 1.000 0.556 0.889 0.778 0.429 0.714
8 0.444 0.000 0.875 0.778 0.444 0.000 0.444 0.444 0.111 0.222 0.000 0.571
9 0.444 0.222 0.375 0.222 1.000 0.000 0.333 0.444 0.111 0.333 0.571 0.857
10 0.333 0.111 0.500 0.222 0.667 0.625 0.556 1.000 0.556 0.143 0.714
11 0.333 0.000 0.375 0.778 0.111 0.750 0.111 0.333 0.778 0.333 0.286 0.571
12 0.333 1.000 0.125 0.222 0.000 0.375 0.222 0.111 1.000 0.000 0.571 0.857
13 0.333 0.778 0.625 0.444 0.111 0.500 0.111 0.222 0.889 0.444 0.714 0.000
14 0.333 0.889 0.750 0.889 0.667 0.875 0.111 0.111 0.222 0.000 0.857 0.286

Зважені нормалізовані оцінки альтернатив:
0 0.004 0.139 0.089 0.024 0.024 0.107 0.024 0.060 0.079 0.024 0.077 0.041
1 0.004 0.099 0.067 0.012 0.020 0.094 0.032 0.000 0.048 0.028 0.064 0.082
2 0.008 0.040 0.022 0.008 0.036 0.107 0.000 0.010 0.048 0.016 0.089 0.143
3 0.002 0.040 0.033 0.032 0.024 0.067 0.004 0.060 0.000 0.000 0.064 0.122
4 0.000 0.139 0.000 0.036 0.012 0.067 0.020 0.089 0.143 0.016 0.064 0.122
5 0.014 0.119 0.033 0.020 0.020 0.000 0.000 0.000 0.063 0.000 0.064 0.000
6 0.018 0.139 0.056 0.016 0.016 0.013 0.000 0.010 0.016 0.036 0.038 0.041
7 0.010 0.139 0.022 0.000 0.000 0.027 0.036 0.050 0.127 0.028 0.038 0.102
8 0.008 0.000 0.078 0.028 0.016 0.000 0.016 0.040 0.016 0.008 0.000 0.082
9 0.008 0.040 0.033 0.008 0.036 0.000 0.012 0.040 0.016 0.012 0.051 0.122
10 0.006 0.020 0.045 0.008 0.024 0.067 0.020 0.050 0.143 0.020 0.013 0.102
11 0.006 0.000 0.033 0.028 0.004 0.080 0.004 0.030 0.111 0.012 0.026 0.082
12 0.006 0.179 0.011 0.008 0.000 0.040 0.008 0.010 0.143 0.000 0.051 0.122
13 0.006 0.139 0.056 0.016 0.004 0.054 0.004 0.020 0.127 0.016 0.064 0.000
14 0.006 0.159 0.067 0.032 0.024 0.094 0.004 0.010 0.032 0.000 0.077 0.041

PIS <утопічна точка>: 0.018 0.179 0.089 0.036 0.036 0.107 0.036 0.000 0.000 0.000 0.000 0.000
NIS <антиутопічна точка>: 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.089 0.143 0.036 0.089 0.143
D+ <відстань до PIS>: 0.142 0.148 0.239 0.219 0.245 0.167 0.135 0.215 0.230 0.234 0.253 0.242 0.227 0.170 0.105
D- <відстань до NIS>: 0.237 0.215 0.175 0.180 0.164 0.230 0.243 0.167 0.196 0.159 0.131 0.146 0.208 0.228 0.266
C <наближеність до PIS>:
<альтернатива: 14, наближеність: 0.717>
<альтернатива: 6, наближеність: 0.642>
<альтернатива: 0, наближеність: 0.625>
<альтернатива: 1, наближеність: 0.593>
<альтернатива: 5, наближеність: 0.579>
<альтернатива: 13, наближеність: 0.573>
<альтернатива: 12, наближеність: 0.478>
<альтернатива: 8, наближеність: 0.460>
<альтернатива: 3, наближеність: 0.451>
<альтернатива: 7, наближеність: 0.437>
<альтернатива: 2, наближеність: 0.424>
<альтернатива: 9, наближеність: 0.404>
<альтернатива: 4, наближеність: 0.402>
<альтернатива: 11, наближеність: 0.376>
<альтернатива: 10, наближеність: 0.342>
Найкраща альтернатива: 14

```

Для цього випадку нормалізовані оцінки альтернатив обчислюються за наступними співвідношеннями.

### 1.1 Для критеріїв прибутку ( $K^+$ )

$$r_{kj}(x) = (x_{kj} - x_j^-) / (x_j^* - x_j^-),$$

де  $x_j^* = \max_k x_{kj}$ ,  $x_j^- = \min_k x_{kj}$ ;

або  $x_j^*$  – бажане значення,

$x_j^-$  – найгірше значення

### 1.2 Для критеріїв витрат ( $K^-$ )

$$r_{kj}(x) = (x_j^* - x_{kj}) / (x_j^* - x_j^-),$$

Всі інші співвідношення співпадають з тими, що наведені для розв'язку частини 1 А.

## Частина 2.1.

На наступних скріншотах відображено поетапний розв'язок задачі багатокритеріальної оптимізації методом VIKOR для параметру  $\vartheta=0,5$ .

```
Microsoft Visual Studio Debug Console

Поетапний розв'язок задачі методом VIKOR

Нормалізовані оцінки альтернатив:
0 0.778 0.222 0.000 0.333 0.333 0.000 0.333 0.667 0.556 0.667 0.857 0.286
1 0.778 0.444 0.250 0.667 0.444 0.125 0.111 0.000 0.333 0.778 0.714 0.571
2 0.556 0.778 0.750 0.778 0.000 0.000 1.000 0.111 0.333 0.444 1.000 1.000
3 0.889 0.778 0.625 0.111 0.333 0.375 0.889 0.667 0.000 0.000 0.714 0.857
4 1.000 0.222 1.000 0.000 0.667 0.375 0.444 1.000 1.000 0.444 0.714 0.857
5 0.222 0.333 0.625 0.444 0.444 1.000 1.000 0.000 0.444 0.000 0.714 0.000
6 0.000 0.222 0.375 0.556 0.556 0.875 1.000 0.111 0.111 1.000 0.429 0.286
7 0.444 0.222 0.750 1.000 0.750 0.000 0.556 0.889 0.778 0.429 0.714
8 0.556 1.000 0.125 0.222 0.556 1.000 0.556 0.444 0.111 0.222 0.000 0.571
9 0.556 0.778 0.625 0.778 0.000 1.000 0.667 0.444 0.111 0.333 0.571 0.857
10 0.667 0.889 0.500 0.778 0.333 0.375 0.444 0.556 1.000 0.556 0.143 0.714
11 0.667 1.000 0.625 0.222 0.889 0.250 0.889 0.333 0.778 0.333 0.286 0.571
12 0.667 0.000 0.875 0.778 1.000 0.625 0.778 0.111 1.000 0.000 0.571 0.857
13 0.667 0.222 0.375 0.556 0.889 0.500 0.889 0.222 0.889 0.444 0.714 0.000
14 0.667 0.111 0.250 0.111 0.333 0.125 0.889 0.111 0.222 0.000 0.857 0.286

Зважені нормалізовані оцінки альтернатив:
0 0.014 0.040 0.000 0.012 0.012 0.000 0.012 0.060 0.079 0.024 0.077 0.041
1 0.014 0.079 0.022 0.024 0.016 0.013 0.004 0.000 0.048 0.028 0.064 0.082
2 0.010 0.139 0.067 0.028 0.000 0.000 0.036 0.010 0.048 0.016 0.089 0.143
3 0.016 0.139 0.056 0.004 0.012 0.040 0.032 0.060 0.000 0.000 0.064 0.122
4 0.018 0.040 0.089 0.000 0.024 0.040 0.016 0.089 0.143 0.016 0.064 0.122
5 0.004 0.060 0.056 0.016 0.016 0.107 0.036 0.000 0.063 0.000 0.064 0.000
6 0.000 0.040 0.033 0.020 0.020 0.094 0.036 0.010 0.016 0.036 0.038 0.041
7 0.008 0.040 0.067 0.036 0.036 0.080 0.000 0.050 0.127 0.028 0.038 0.102
8 0.010 0.179 0.011 0.008 0.020 0.107 0.020 0.040 0.016 0.008 0.000 0.082
9 0.010 0.139 0.056 0.028 0.000 0.107 0.024 0.040 0.016 0.012 0.051 0.122
10 0.012 0.159 0.045 0.028 0.012 0.040 0.016 0.050 0.143 0.020 0.013 0.102
11 0.012 0.179 0.056 0.008 0.032 0.027 0.032 0.030 0.111 0.012 0.026 0.082
12 0.012 0.000 0.078 0.028 0.036 0.067 0.028 0.010 0.143 0.000 0.051 0.122
13 0.012 0.040 0.033 0.020 0.032 0.054 0.032 0.020 0.127 0.016 0.064 0.000
14 0.012 0.020 0.022 0.004 0.012 0.013 0.032 0.010 0.032 0.000 0.077 0.041
```

```
Microsoft Visual Studio Debug Co...

S:
<альтернатива: 4, наближеність: 0.661>
<альтернатива: 10, наближеність: 0.638>
<альтернатива: 7, наближеність: 0.611>
<альтернатива: 11, наближеність: 0.604>
<альтернатива: 9, наближеність: 0.604>
<альтернатива: 2, наближеність: 0.585>
<альтернатива: 12, наближеність: 0.575>
<альтернатива: 3, наближеність: 0.544>
<альтернатива: 8, наближеність: 0.500>
<альтернатива: 13, наближеність: 0.448>
<альтернатива: 5, наближеність: 0.421>
<альтернатива: 1, наближеність: 0.393>
<альтернатива: 6, наближеність: 0.383>
<альтернатива: 0, наближеність: 0.369>
<альтернатива: 14, наближеність: 0.274>

R:
<альтернатива: 8, наближеність: 0.179>
<альтернатива: 11, наближеність: 0.179>
<альтернатива: 10, наближеність: 0.159>
<альтернатива: 2, наближеність: 0.143>
<альтернатива: 4, наближеність: 0.143>
<альтернатива: 12, наближеність: 0.143>
<альтернатива: 3, наближеність: 0.139>
<альтернатива: 9, наближеність: 0.139>
<альтернатива: 7, наближеність: 0.127>
<альтернатива: 13, наближеність: 0.127>
<альтернатива: 5, наближеність: 0.107>
<альтернатива: 6, наближеність: 0.094>
<альтернатива: 1, наближеність: 0.082>
<альтернатива: 0, наближеність: 0.079>
<альтернатива: 14, наближеність: 0.077>

Q:
<альтернатива: 11, наближеність: 0.927>
<альтернатива: 10, наближеність: 0.873>
<альтернатива: 4, наближеність: 0.825>
<альтернатива: 8, наближеність: 0.791>
<альтернатива: 9, наближеність: 0.732>
<альтернатива: 2, наближеність: 0.727>
<альтернатива: 12, наближеність: 0.713>
<альтернатива: 7, наближеність: 0.683>
<альтернатива: 3, наближеність: 0.655>
<альтернатива: 13, наближеність: 0.473>
<альтернатива: 5, наближеність: 0.340>
<альтернатива: 6, наближеність: 0.225>
<альтернатива: 1, наближеність: 0.179>
<альтернатива: 0, наближеність: 0.137>
<альтернатива: 14, наближеність: 0.000>

C1: 14
C2: 14

Компромісний розв'язок: 14
```

Спочатку були обчислені нормалізовані оцінки альтернатив за наступним співвідношенням.

$$|f_j^* - f_{kj}| / |f_j^* - f_j^-|$$

Потім кожна нормалізована оцінка була домножена на вагу відповідного критерію.

$$w_j |f_j^* - f_{kj}| / |f_j^* - f_j^-|$$

Далі для кожної альтернативи були обчислені значення S та R за наступними співвідношеннями.

$$S_k = \sum_{j=1}^m w_j |f_j^* - f_{kj}| / |f_j^* - f_j^-|$$

$$R_k = \max_j \left\{ w_j |f_j^* - f_{kj}| / |f_j^* - f_j^-| \right. \\ \left. j = 1, 2, \dots, m \right\}$$

$$k = 1, 2, \dots, n$$

$S_k$  – визначає відстань від альтернативи  $A_k$  до компромісного розв’язку  $A^*$ , визначену за манхеттенською метрикою (чим менша відстань, тим ближче  $A_k$  до  $A^*$ ).

$R_k$  – визначає відстань від альтернативи  $A_k$  до компромісного розв’язку  $A^*$ , визначену за метрикою Чебишева (чим менша відстань, тим ближче  $A_k$  до  $A^*$ ).

На базі значень S та R були обчислені значення Q.

$$Q_k = v(S_k - S^*) / (S^- - S^*) + \\ + (1 - v)(R_k - R^*) / (R^- - R^*)$$

де  $S^* = \min_k S_k$  (або  $S^* = 0$  - означає досягнення бажаного рівня)

$S^- = \max_k S_k$  (або  $S^- = 1$  - найгірший рівень),

$R^* = \min_j R_j$  (або  $R^* = 0$  і  $R^- = 1$  аналогічно)  
 $R^- = \max_j R_j$

На даному етапі маємо 3 ранжування альтернатив за значеннями S, R, Q. Впорядовуємо їх у порядку спадання.

Далі для відбору компромісного розв’язку необхідно перевірити 2 умови. Пропонується вважати компромісним розв’язком альтернативу  $a^*$ , яка в ранжуванні за Q є найкращою (мінімальне значення  $Q_k$ ), якщо виконуються дві наступні умови:

**C1.** «Прийнятна перевага»

**C2.** «Прийнятна стабільність».



### C1. «Прийнятна перевага»:

$$Q(a'') - Q(a') \geq DQ,$$

де  $a''$  – наступна найкраща альтернатива в  
ранжуванні за  $Q$  після  $a'$ ,  
 $DQ = 1/(n-1)$ ,  $n$  – кількість альтернатив

## C2. «Прийнятна стабільність»:

Альтернатива  $a'$  повинна також мати і найкращі значення (мінімальні) в ранжуваннях за  $S$  та/або  $R$

У компромісний розв'язок входять ті альтернативи, які задовольняють обидві умови.

## Частина 2.2.

Дослідимо в методі VIKOR вплив зміни параметра  $\vartheta$  на результуючий компромісний розв'язок. Розглянемо випадки  $0 \leq \vartheta \leq 1$  з кроком 0.1.

The figure displays three screenshots of the Microsoft Visual Studio Debug Console, showing the results of a program for different values of  $v$  (0.0, 0.1, and 0.2). Each screenshot displays a list of 14 alternatives with their corresponding 'наближеність' (approximation) values.

**Скріншот 1:  $v = 0.0$**

Альтернатива	наближеність
4	0.661
10	0.638
7	0.611
11	0.604
1	0.604
2	0.585
12	0.575
3	0.544
8	0.500
13	0.448
5	0.421
1	0.393
6	0.383
0	0.369
14	0.274

**Скріншот 2:  $v = 0.1$**

Альтернатива	наближеність
4	0.661
10	0.638
7	0.611
11	0.604
1	0.604
2	0.585
12	0.575
3	0.544
8	0.500
13	0.448
5	0.421
1	0.393
6	0.383
0	0.369
14	0.274

**Скріншот 3:  $v = 0.2$**

Альтернатива	наближеність
4	0.661
10	0.638
7	0.611
11	0.604
1	0.604
2	0.585
12	0.575
3	0.544
8	0.500
13	0.448
5	0.421
1	0.393
6	0.383
0	0.369
14	0.274



```
Microsoft Visual Studio Debug Co...
Для значення  $\vartheta = 0.9$ 
S:
<альтернатива: 4, наближеність: 0.661>
<альтернатива: 10, наближеність: 0.638>
<альтернатива: 7, наближеність: 0.611>
<альтернатива: 11, наближеність: 0.604>
<альтернатива: 9, наближеність: 0.604>
<альтернатива: 2, наближеність: 0.585>
<альтернатива: 12, наближеність: 0.575>
<альтернатива: 3, наближеність: 0.544>
<альтернатива: 8, наближеність: 0.500>
<альтернатива: 13, наближеність: 0.448>
<альтернатива: 5, наближеність: 0.421>
<альтернатива: 1, наближеність: 0.393>
<альтернатива: 6, наближеність: 0.383>
<альтернатива: 0, наближеність: 0.369>
<альтернатива: 14, наближеність: 0.274>
R:
<альтернатива: 8, наближеність: 0.179>
<альтернатива: 11, наближеність: 0.179>
<альтернатива: 10, наближеність: 0.159>
<альтернатива: 2, наближеність: 0.143>
<альтернатива: 4, наближеність: 0.143>
<альтернатива: 12, наближеність: 0.143>
<альтернатива: 3, наближеність: 0.139>
<альтернатива: 9, наближеність: 0.139>
<альтернатива: 7, наближеність: 0.127>
<альтернатива: 13, наближеність: 0.127>
<альтернатива: 5, наближеність: 0.107>
<альтернатива: 6, наближеність: 0.094>
<альтернатива: 1, наближеність: 0.082>
<альтернатива: 0, наближеність: 0.079>
<альтернатива: 14, наближеність: 0.077>
Q:
<альтернатива: 4, наближеність: 0.965>
<альтернатива: 10, наближеність: 0.927>
<альтернатива: 11, наближеність: 0.869>
<альтернатива: 7, наближеність: 0.833>
<альтернатива: 9, наближеність: 0.829>
<альтернатива: 2, наближеність: 0.788>
<альтернатива: 12, наближеність: 0.764>
<альтернатива: 3, наближеність: 0.689>
<альтернатива: 8, наближеність: 0.625>
<альтернатива: 13, наближеність: 0.455>
<альтернатива: 5, наближеність: 0.372>
<альтернатива: 1, наближеність: 0.283>
<альтернатива: 6, наближеність: 0.270>
<альтернатива: 0, наближеність: 0.224>
<альтернатива: 14, наближеність: 0.000>
C1: 14
C2: 14
Компронісний розв'язок: 14
```

```
Microsoft Visual Studio Debug Co...
Для значення  $\vartheta = 1.0$ 
S:
<альтернатива: 4, наближеність: 0.661>
<альтернатива: 10, наближеність: 0.638>
<альтернатива: 7, наближеність: 0.611>
<альтернатива: 11, наближеність: 0.604>
<альтернатива: 9, наближеність: 0.604>
<альтернатива: 2, наближеність: 0.585>
<альтернатива: 12, наближеність: 0.575>
<альтернатива: 3, наближеність: 0.544>
<альтернатива: 8, наближеність: 0.500>
<альтернатива: 13, наближеність: 0.448>
<альтернатива: 5, наближеність: 0.421>
<альтернатива: 1, наближеність: 0.393>
<альтернатива: 6, наближеність: 0.383>
<альтернатива: 0, наближеність: 0.369>
<альтернатива: 14, наближеність: 0.274>
R:
<альтернатива: 8, наближеність: 0.179>
<альтернатива: 11, наближеність: 0.179>
<альтернатива: 10, наближеність: 0.159>
<альтернатива: 2, наближеність: 0.143>
<альтернатива: 4, наближеність: 0.143>
<альтернатива: 12, наближеність: 0.143>
<альтернатива: 3, наближеність: 0.139>
<альтернатива: 9, наближеність: 0.139>
<альтернатива: 7, наближеність: 0.127>
<альтернатива: 13, наближеність: 0.127>
<альтернатива: 5, наближеність: 0.107>
<альтернатива: 6, наближеність: 0.094>
<альтернатива: 1, наближеність: 0.082>
<альтернатива: 0, наближеність: 0.079>
<альтернатива: 14, наближеність: 0.077>
Q:
<альтернатива: 4, наближеність: 1.000>
<альтернатива: 10, наближеність: 0.941>
<альтернатива: 7, наближеність: 0.871>
<альтернатива: 11, наближеність: 0.854>
<альтернатива: 9, наближеність: 0.854>
<альтернатива: 2, наближеність: 0.803>
<альтернатива: 12, наближеність: 0.777>
<альтернатива: 3, наближеність: 0.698>
<альтернатива: 8, наближеність: 0.583>
<альтернатива: 13, наближеність: 0.451>
<альтернатива: 5, наближеність: 0.380>
<альтернатива: 1, наближеність: 0.308>
<альтернатива: 6, наближеність: 0.281>
<альтернатива: 0, наближеність: 0.246>
<альтернатива: 14, наближеність: 0.000>
C1: 14
C2: 14
Компронісний розв'язок: 14
```

Бачимо, що зі зміною значення параметра  $\vartheta$  ранжування для Q також змінюється. Для малих значень  $\vartheta = 0$  та  $\vartheta = 0.1$  прийнятну перевагу задовольняє більша кількість альтернатив, а саме множини  $\{14, 0, 1\}$  та  $\{14, 0\}$ . Однак, для більших значень параметру  $\vartheta$  прийнятну умову задовольняє лише одна альтернатива. Для всіх значень до компромісного розв'язку входить лише одна альтернатива, а саме 14.

## 4. Лістинг програми.

Посилання на github-репозиторій з кодом:

<https://github.com/KasprukNastia/decisions/tree/master/Lab5>

Клас **TOPSIS**

```
/// <summary>
/// Клас, що описує метод TOPSIS
/// </summary>
public class TOPSIS
{
    /// <summary>
    /// Значення критеріїв для альтернатив
    /// </summary>
    public int[][] Evaluations { get; }

    /// <summary>
    /// Ваги критеріїв
    /// </summary>
    public IReadOnlyList<double> Weights { get; }

    /// <summary>
    /// Критерії, які потрібно максимізувати
    /// </summary>
    public IReadOnlyList<int> WeightsToMaximize { get; }

    /// <summary>
    /// Критерії, які потрібно мінімізувати
    /// </summary>
    public IReadOnlyList<int> WeightsToMinimize { get; }

    /// <summary>
    /// К-сть критеріїв
    /// </summary>
    public int CriteriasCount { get; }

    /// <summary>
    /// К-сть альтернатив
    /// </summary>
    public int AlternativesCount { get; }

    /// <summary>
    /// Нормалізовані оцінки альтернатив
    /// </summary>
    private double[][] _normalizedEvaluations;
    /// <summary>
    /// Нормалізовані оцінки альтернатив
    /// </summary>
    public double[][] NormalizedEvaluations
    {
        get
        {
            if (_normalizedEvaluations == null)
            {
                if (WeightsToMaximize == null && WeightsToMinimize == null)
                    _normalizedEvaluations = CalcNormalizedAllMax();
                else
                    _normalizedEvaluations = CalcNormalizedMinAndMax();
            }
            return _normalizedEvaluations;
        }
    }
}
```

```

/// <summary>
/// Зважені нормалізовані оцінки альтернатив
/// </summary>
private double[][] _weightedNormalizedEvaluations;
/// <summary>
/// Зважені нормалізовані оцінки альтернатив
/// </summary>
public double[][] WeightedNormalizedEvaluations
{
    get
    {
        if (_weightedNormalizedEvaluations == null)
        {
            _weightedNormalizedEvaluations = new double[AlternativesCount][];
            for (int i = 0; i < AlternativesCount; i++)
            {
                _weightedNormalizedEvaluations[i] = new double[CriteriaCount];
                for (int j = 0; j < CriteriaCount; j++)
                {
                    _weightedNormalizedEvaluations[i][j] = NormalizedEvaluations[i][j] * Weights[j];
                }
            }
        }
        return _weightedNormalizedEvaluations;
    }
}

/// <summary>
/// Утопічна точка
/// </summary>
private IReadOnlyList<double> _pis;
/// <summary>
/// Утопічна точка
/// </summary>
public IReadOnlyList<double> PIS
{
    get
    {
        if (_pis == null)
        {
            if (WeightsToMaximize == null && WeightsToMinimize == null)
                _pis = Enumerable.Range(0, CriteriaCount)
                    .Select(criteria => GetMinAndMaxForCriteriaInWeighted(criteria))
                    .Select(minMax => minMax.max).ToList();
            else
                _pis = Enumerable.Range(0, CriteriaCount)
                    .Select(criteria => GetMinAndMaxForCriteriaInWeighted(criteria))
                    .Select((minMax, criteria) => WeightsToMaximize.Contains(criteria) ? minMax.max : minMax.min)
                    .ToList();
        }
        return _pis;
    }
}

/// <summary>
/// Антиутопічна точка
/// </summary>
private IReadOnlyList<double> _nis;
/// <summary>
/// Антиутопічна точка
/// </summary>
public IReadOnlyList<double> NIS
{
    get
    {
        if (_nis == null)

```

```

    {
        if (WeightsToMaximize == null && WeightsToMinimize == null)
            _nis = Enumerable.Range(0, CriteriasCount)
                .Select(criteria => GetMinAndMaxForCriteriaInWeighted(criteria))
                .Select(minMax => minMax.min).ToList();
        else
            _nis = Enumerable.Range(0, CriteriasCount)
                .Select(criteria => GetMinAndMaxForCriteriaInWeighted(criteria))
                .Select((minMax, criteria) => WeightsToMaximize.Contains(criteria) ? minMax.min : minMax.max)
                .ToList();
    }
    return _nis;
}
}

```

```

/// <summary>
/// Відстані до утопічної точки
/// </summary>
private IReadOnlyList<double> _dToPis;
/// <summary>
/// Відстані до утопічної точки
/// </summary>
public IReadOnlyList<double> DToPIS
{
    get
    {
        if(_dToPis == null)
        {
            double[] dToPis = new double[AlternativesCount];
            for(int alternative = 0; alternative < AlternativesCount; alternative++)
            {
                dToPis[alternative] = Math.Sqrt(
                    WeightedNormalizedEvaluations[alternative]
                        .Select((elem, criteria) => Math.Pow(elem - PIS[criteria], 2))
                        .Sum());
            }
            _dToPis = dToPis.ToList();
        }
        return _dToPis;
    }
}

```

```

/// <summary>
/// Відстані до антиутопічної точки
/// </summary>
private IReadOnlyList<double> _dToNis;
/// <summary>
/// Відстані до антиутопічної точки
/// </summary>
public IReadOnlyList<double> DToNIS
{
    get
    {
        if (_dToNis == null)
        {
            double[] dToNis = new double[AlternativesCount];
            for (int alternative = 0; alternative < AlternativesCount; alternative++)
            {
                dToNis[alternative] = Math.Sqrt(
                    WeightedNormalizedEvaluations[alternative]
                        .Select((elem, criteria) => Math.Pow(elem - NIS[criteria], 2))
                        .Sum());
            }
            _dToNis = dToNis.ToList();
        }
        return _dToNis;
    }
}

```

```

    }
}

/// <summary>
/// Наближеності до утопічної точки
/// </summary>
private IReadOnlyList<(int alternative, double closeness)> _c;
/// <summary>
/// Наближеності до утопічної точки
/// </summary>
public IReadOnlyList<(int alternative, double closeness)> C
{
    get
    {
        if(_c == null)
            _c = DToNIS.Select((dToNis, alternative) => (alternative, dToNis / (DToPIS[alternative] + dToNis)))
                .OrderByDescending(elem => elem.Item2)
                .ToList();
        return _c;
    }
}

public TOPSIS(int[][] evaluations,
    IReadOnlyList<double> weights,
    IReadOnlyList<int> weightsToMaximize = null,
    IReadOnlyList<int> weightsToMinimize = null)
{
    Evaluations = evaluations ?? throw new ArgumentNullException(nameof(evaluations));

    AlternativesCount = evaluations.Length;
    if (AlternativesCount > 0)
        CriteriasCount = evaluations[0].Length;
    for (int i = 1; i < AlternativesCount; i++)
    {
        if (evaluations[i].Length != CriteriasCount)
            throw new ArgumentException($"Evaluation must be provided only for {CriteriasCount} criterias");
    }

    Weights = weights ?? throw new ArgumentNullException(nameof(weights));
    if (Weights.Count != CriteriasCount)
        throw new ArgumentException("The number of weights does not meet the number of criterias");
    double weightsSum = Weights.Sum();
    Weights = Weights.Select(w => w / weightsSum).ToList();

    WeightsToMaximize = weightsToMaximize;
    WeightsToMinimize = weightsToMinimize;
}

/// <summary>
/// Обчислення нормалізованих оцінок альтернатив для випадку, коли всі критерії потрібно максимізувати
/// </summary>
private double[][] CalcNormalizedAllMax()
{
    double[][] normalized = new double[AlternativesCount][];

    List<double> geomMeansForCriterias = Enumerable.Range(0, CriteriasCount)
        .Select(criteria => CalcGeometricMeanForCriteria(criteria)).ToList();
    for (int i = 0; i < AlternativesCount; i++)
    {
        normalized[i] = new double[CriteriasCount];
        for(int j = 0; j < CriteriasCount; j++)
        {
            normalized[i][j] = Evaluations[i][j] / geomMeansForCriterias[j];
        }
    }
}

```

```

        return normalized;
    }

    /// <summary>
    /// Обчислення нормалізованих оцінок альтернатив для випадку,
    /// коли частину критеріїв потрібно максимізувати, а частину мінімізувати
    /// </summary>
    private double[][] CalcNormalizedMinAndMax()
    {
        double[][] normalized = new double[AlternativesCount][];

        List<(int min, int max)> minMaxForCriterias = Enumerable.Range(0, CriteriasCount)
            .Select(criteria => GetMinAndMaxForCriteria(criteria)).ToList();
        for (int i = 0; i < AlternativesCount; i++)
        {
            normalized[i] = new double[CriteriasCount];
            for (int j = 0; j < CriteriasCount; j++)
            {
                (int min, int max) = minMaxForCriterias[j];
                normalized[i][j] = WeightsToMaximize.Contains(j) ?
                    (Evaluations[i][j] - min) / (double)(max - min) :
                    (max - Evaluations[i][j]) / (double)(max - min);
            }
        }

        return normalized;
    }

    /// <summary>
    /// Обчислення середнього геометричного для значень критерію
    /// </summary>
    private double CalcGeometricMeanForCriteria(int criteria)
    {
        double sum = 0;
        for(int alternative = 0; alternative < AlternativesCount; alternative++)
        {
            sum += Math.Pow(Evaluations[alternative][criteria], 2);
        }
        return Math.Sqrt(sum);
    }

    /// <summary>
    /// Обчислення мінімального та максимального значень для критерію
    /// з початкових оцінок альтернатив
    /// </summary>
    private (int min, int max) GetMinAndMaxForCriteria(int criteria)
    {
        int[] criteriaValues = new int[AlternativesCount];
        for (int i = 0; i < AlternativesCount; i++)
        {
            criteriaValues[i] = Evaluations[i][criteria];
        }

        return (criteriaValues.Min(), criteriaValues.Max());
    }

    /// <summary>
    /// Обчислення мінімального та максимального значень для критерію
    /// зі зважених оцінок альтернатив
    /// </summary>
    private (double min, double max) GetMinAndMaxForCriteriaInWeighted(int criteria)
    {
        double[] criteriaValues = new double[AlternativesCount];
        for (int i = 0; i < AlternativesCount; i++)
        {
            criteriaValues[i] = WeightedNormalizedEvaluations[i][criteria];
        }
    }

```



```

    }

    return (criteriaValues.Min(), criteriaValues.Max());
}
}

```

## Клас **VIKOR**

```

/// <summary>
/// Клас, що описує метод VIKOR
/// </summary>
public class VIKOR
{
    /// <summary>
    /// Значення критеріїв для альтернатив
    /// </summary>
    public int[][] Evaluations { get; }

    /// <summary>
    /// Значення коефіцієнту V
    /// </summary>
    public double VCoef { get; }

    /// <summary>
    /// Ваги критеріїв
    /// </summary>
    public IReadOnlyList<double> Weights { get; }

    /// <summary>
    /// К-сть критеріїв
    /// </summary>
    public int CriteriasCount { get; }

    /// <summary>
    /// К-сть альтернатив
    /// </summary>
    public int AlternativesCount { get; }

    /// <summary>
    /// Нормалізовані оцінки альтернатив
    /// </summary>
    private double[][] _normalizedEvaluations;

    /// <summary>
    /// Нормалізовані оцінки альтернатив
    /// </summary>
    public double[][] NormalizedEvaluations
    {
        get
        {
            if (_normalizedEvaluations == null)
            {
                _normalizedEvaluations = new double[AlternativesCount][];

                IReadOnlyList<(int min, int max)> minMaxForCriterias = Enumerable.Range(0, CriteriasCount)
                    .Select(criteria => GetMinAndMaxForCriteria(criteria)).ToList();
                for (int i = 0; i < AlternativesCount; i++)
                {
                    _normalizedEvaluations[i] = new double[CriteriasCount];
                    for (int j = 0; j < CriteriasCount; j++)
                    {
                        (int min, int max) = minMaxForCriterias[j];
                        _normalizedEvaluations[i][j] = (max - Evaluations[i][j]) / (double)(max - min);
                    }
                }
            }
        }
    }
}

```

```

        return _normalizedEvaluations;
    }
}

/// <summary>
/// Зважені нормалізовані оцінки альтернатив
/// </summary>
private double[][] _weightedNormalizedEvaluations;
/// <summary>
/// Зважені нормалізовані оцінки альтернатив
/// </summary>
public double[][] WeightedNormalizedEvaluations
{
    get
    {
        if (_weightedNormalizedEvaluations == null)
        {
            _weightedNormalizedEvaluations = new double[AlternativesCount][];
            for (int i = 0; i < AlternativesCount; i++)
            {
                _weightedNormalizedEvaluations[i] = new double[CriteriaCount];
                for (int j = 0; j < CriteriaCount; j++)
                {
                    _weightedNormalizedEvaluations[i][j] = NormalizedEvaluations[i][j] * Weights[j];
                }
            }
        }
        return _weightedNormalizedEvaluations;
    }
}

/// <summary>
/// Значення S - середні інтервали покращення
/// </summary>
private IReadOnlyList<(int alternative, double sValue)> _s;
/// <summary>
/// Значення S - середні інтервали покращення
/// </summary>
public IReadOnlyList<(int alternative, double sValue)> S
{
    get
    {
        if (_s == null)
            _s = WeightedNormalizedEvaluations.Select((wne, alternative) => (alternative, wne.Sum()))
                .ToList();
        return _s;
    }
}

/// <summary>
/// Значення R - максимальні інтервали покращення
/// </summary>
private IReadOnlyList<(int alternative, double rValue)> _r;
/// <summary>
/// Значення R - максимальні інтервали покращення
/// </summary>
public IReadOnlyList<(int alternative, double rValue)> R
{
    get
    {
        if (_r == null)
            _r = WeightedNormalizedEvaluations.Select((wne, alternative) => (alternative, wne.Max()))
                .ToList();
        return _r;
    }
}

```

```

/// <summary>
/// Значення Q
/// </summary>
private IReadOnlyList<(int alternative, double qValue)> _q;
/// <summary>
/// Значення Q
/// </summary>
public IReadOnlyList<(int alternative, double qValue)> Q
{
    get
    {
        if(_q == null)
        {
            var sValues = S.Select(pair => pair.sValue);
            var rValues = R.Select(pair => pair.rValue);
            double sMin = sValues.Min(), sMax = sValues.Max(), rMin = rValues.Min(), rMax = rValues.Max();
            _q = Enumerable.Range(0, AlternativesCount)
                .Select(alternative =>
                    (alternative, (VCoef * (S[alternative].sValue - sMin) / (sMax - sMin)) + ((1 - VCoef) * (R[alternative].rValue - rMin)
/ (rMax - rMin))))
                .ToList();
        }
        return _q;
    }
}

/// <summary>
/// Значення C1 - прийнята перевага
/// </summary>
private List<int> _c1;
/// <summary>
/// Значення C1 - прийнята перевага
/// </summary>
public IReadOnlyList<int> C1
{
    get
    {
        if(_c1 == null)
        {
            IReadOnlyList<(int alternative, double qValue)> orderedQ = Q.OrderBy(elem => elem.qValue).ToList();

            _c1 = new List<int>();
            _c1.Add(orderedQ.First().alternative);
            for(int i = 1; i < AlternativesCount; i++)
            {
                if ((orderedQ[i].qValue - orderedQ.First().qValue) < (1 / (double)(AlternativesCount - 1)))
                    _c1.Add(orderedQ[i].alternative);
                else break;
            }
        }
        return _c1;
    }
}

/// <summary>
/// Значення C2 - прийнята стабільність
/// </summary>
private List<int> _c2;
/// <summary>
/// Значення C2 - прийнята стабільність
/// </summary>
public IReadOnlyList<int> C2
{
    get
    {

```

```

        if (_c2 == null)
        {
            IReadOnlyList<int alternative, double sValue> orderedS = S.OrderBy(elem => elem.sValue).ToList();
            IReadOnlyList<int alternative, double rValue> orderedR = R.OrderBy(elem => elem.rValue).ToList();

            _c2 = new List<int>();
            foreach(int alternative in C1)
            {
                if (S[alternative].sValue == orderedS.First().sValue ||
                    R[alternative].rValue == orderedR.First().rValue)
                    _c2.Add(alternative);
            }
        }
        return _c2;
    }
}

```

```

/// <summary>
/// Фінальний результат, що задовольняє обидві умови C1 і C2
/// </summary>

```

```
private IReadOnlyList<int> _finalResult;
```

```

/// <summary>
/// Фінальний результат, що задовольняє обидві умови C1 і C2
/// </summary>

```

```
public IReadOnlyList<int> FinalResult
```

```

{
    get
    {
        if(_finalResult == null)
            _finalResult = C1.Intersect(C2).ToList();
        return _finalResult;
    }
}

```

```

public VIKOR(int[][] evaluations,
    double vCoef,
    IReadOnlyList<double> weights)

```

```

{
    Evaluations = evaluations ?? throw new ArgumentNullException(nameof(evaluations));

```

```

    if (vCoef < 0 || vCoef > 1)
        throw new ArgumentException($"{nameof(vCoef)} value must be value between 0 and 1");
    vCoef = vCoef;

```

```

    AlternativesCount = evaluations.Length;
    if (AlternativesCount > 0)
        CriteriasCount = evaluations[0].Length;
    for (int i = 1; i < AlternativesCount; i++)
    {
        if (evaluations[i].Length != CriteriasCount)
            throw new ArgumentException($"Evaluation must be provided only for {CriteriasCount} criterias");
    }

```

```

    Weights = weights ?? throw new ArgumentNullException(nameof(weights));
    if (Weights.Count != CriteriasCount)
        throw new ArgumentException("The number of weights does not meet the number of criterias");
    double weightsSum = Weights.Sum();
    Weights = Weights.Select(w => w / weightsSum).ToList();
}

```

```

/// <summary>
/// Обчислення мінімального та максимального значень для критерію
/// з початкових оцінок альтернатив
/// </summary>
private (int min, int max) GetMinAndMaxForCriteria(int criteria)
{

```

```
int[] criteriaValues = new int[AlternativesCount];
for (int i = 0; i < AlternativesCount; i++)
{
    criteriaValues[i] = Evaluations[i][criteria];
}

return (criteriaValues.Min(), criteriaValues.Max());
}
}
```

## 5. Опис класів. Перелік розроблених функцій на методів.

Клас	Властивість	Опис	Тип значення, що повертає
TOPSIS	Evaluations	Значення критеріїв для альтернатив	int[][]
	Weights	Ваги критеріїв	ICollection<double>
	WeightsToMaximize	Критерії, які потрібно максимізувати	ICollection<int>
	WeightsToMinimize	Критерії, які потрібно мінімізувати	ICollection<int>
	CriteriaCount	К-сть критеріїв	int
	AlternativesCount	К-сть альтернатив	int
	NormalizedEvaluations	Нормалізовані оцінки альтернатив	double[][]
	WeightedNormalizedEvaluations	Зважені нормалізовані оцінки альтернатив	double[][]
	PIS	Утопічна точка	ICollection<double>
	NIS	Антиутопічна точка	ICollection<double>
	DToPIS	Відстані до утопічної точки	ICollection<double>
	DToNIS	Відстані до антиутопічної точки	ICollection<double>
	C	Наближеності до утопічної точки	ICollection<(int alternative, double closeness)>
VIKOR	Evaluations	Значення критеріїв для альтернатив	int[][]
	VCoef	Значення коефіцієнту V	double
	Weights	Ваги критеріїв	ICollection<double>
	CriteriaCount	К-сть критеріїв	int
	AlternativesCount	К-сть альтернатив	int
	NormalizedEvaluations	Нормалізовані оцінки альтернатив	double[][]
	WeightedNormalizedEvaluations	Зважені нормалізовані оцінки альтернатив	double[][]
	S	Значення S - середні інтервали покращення	ICollection<(int alternative, double sValue)>
	R	Значення R - максимальні інтервали покращення	ICollection<(int alternative, double rValue)>
	Q	Значення Q	ICollection<(int alternative, double qValue)>
	C1	Значення C1 - прийнята перевага	ICollection<int>
	C2	Значення C2 - прийнята стабільність	ICollection<int>
	FinalResult	Фінальний результат, що задовольняє обидві умови C1 і C2	ICollection<int>

Клас	Ф-ція/Метод	Параметри	Опис	Значення, що повертає
TOPSIS	CalcNormalizedAllMax	-	Обчислення нормалізованих оцінок альтернатив для випадку, коли всі критерії потрібно максимізувати	double[][] – нормалізовані оцінки альтернатив
	CalcNormalizedMinAndMax	-	Обчислення нормалізованих оцінок альтернатив для випадку, коли частину критеріїв потрібно максимізувати, а частину мінімізувати	double[][] – нормалізовані оцінки альтернатив
	CalcGeometricMeanForCriteria	int criteria – номер критерію	Обчислення середнього геометричного для значень критерію	double – середнє геометричне для значень критерію
	GetMinAndMaxForCriteria	int criteria – номер критерію	Обчислення мінімального та максимального значень для критерію з початкових оцінок альтернатив	int min – мінімальне значення критерію; int max – максимальне значення критерію
	GetMinAndMaxForCriteriaInWeighted	int criteria – номер критерію	Обчислення мінімального та максимального значень для критерію зі зважених оцінок альтернатив	int min – мінімальне значення критерію; int max – максимальне значення критерію
VIKOR	GetMinAndMaxForCriteria	int criteria – номер критерію	Обчислення мінімального та максимального значень для критерію з початкових оцінок альтернатив	int min – мінімальне значення критерію; int max – максимальне значення критерію

## 6. Висновки.

У даній лабораторній роботі ми мали можливість навчитися розв'язувати задачі багатокритеріальної оптимізації застосовуючи методи TOPSIS і VIKOR.

Основна ідея методу TOPSIS дуже проста: після визначення «ідеального» і «ідеально-негативного» очікуваного станів проводиться спроба пошуку такого рішення, яке б дозволяло максимально наблизитися до «ідеального» і залишатися максимально віддаленим від «ідеально-негативного».

Процес прийняття рішення починається з раунду оцінювання, коли експерт оцінює всі альтернативні рішення за всіма критеріями. В результаті формується матриця рішень.

Метод TOPSIS складається з 6 послідовних кроків:

- 1) розрахунок нормалізованої матриці рішення;
- 2) розрахунок зваженої нормалізованої матриці рішення;
- 3) визначення «ідеального» і «ідеально-негативного» очікуваного станів;
- 4) розрахунок віддаленості альтернатив до «ідеального» і «ідеально-негативного» станів;
- 5) розрахунок відносній близькості до «ідеального» стану;
- 6) ранжування критеріїв.

Метод VIKOR також базується на визначенні віддаленості альтернативи до компромісного розв'язку  $A^*$ . Отримане цим методом рішення може бути прийнято за найкраще, оскільки воно забезпечує максимальну корисність більшості (представлену  $\min S$ ) і мінімальні індивідуальні втрати опонента (представлене  $\min R$ ). Параметри  $S$  і  $R$  інтегровані в  $Q$  для визначення компромісного рішення, що базується на взаємних поступках. Коефіцієнт  $\vartheta$  дозволяє впливати на це компромісне рішення, з наданням переваги  $S$  або  $R$ .

Обидва методи дозволяють визначити найкращу альтернативу. Метод TOPSIS зрозуміліший для реалізації, однак VIKOR гнучкіший.

У результаті ми навчилися розв'язувати задачі багатокритеріальної оптимізації застосовуючи методи TOPSIS і VIKOR та змогли знайти рішення задачі у відповідності з варіантом.