Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем і технологій

Практикум №2
З дисципліни «Теорія прийняття рішень»

На тему

«Оптимізація за бінарним відношенням»

Виконала: студентка гр. ІС-03
Козюк Ю.О.
Перевірила: Жураковська О. С.

Київ-2023

Варіант 64

Завдання 1

Для кожного з бінарних відношень R1-R8 (із завдання 1 практикуму 1) визначити множину найкращих альтернатив за принципами домінування та блокування.

Таблиця 1 – Результати виконання завдання 1

Відношення	**Клас, до якого належить БВ**	**Опт. альтернативи за принципом домінування**	**Опт. альтернативи за принципом блокування**
R1	Квазіпорядок	$X^*_R = \{1\}$ $X^{**}_R = \{1\}$	$X^0_R = \{1\}$ $X^{00}_R = \{1\}$
R2	Строгий порядок	$X^*_P = \{5\}$	$X^0_P = \{5\}$
R3	Слабке впорядкування	$X^*_P = \{\emptyset\}$	$X^0_P = \{4, 7, 8\}$
R4	Класу немає	$X^*_R = \{\emptyset\}$ $X^{**}_R = \{\emptyset\}$	$X^0_R = \{2, 8\}$ $X^{00}_R = \{\emptyset\}$
R5	Еквівалетність	$X^*_R = \{\emptyset\}$ $X^{**}_R = \{\emptyset\}$	$X^0_R = \{1, 2, 3, 4, 5, 6, 7, 8\}$ $X^{00}_R = \{\emptyset\}$
R6	Нестрогий порядок	$X^*_R = \{8\}$ $X^{**}_R = \{8\}$	$X^0_R = \{8\}$ $X^{00}_R = \{8\}$
R7	Класу немає	$X^*_R = \{\emptyset\}$ $X^{**}_R = \{\emptyset\}$	$X^0_R = \{\emptyset\}$ $X^{00}_R = \{\emptyset\}$
R8	Квазіпорядок	$X^*_R = \{5\}$ $X^{**}_R = \{5\}$	$X^0_R = \{5\}$ $X^{00}_R = \{5\}$

Завдання 2

На множині із 15 альтернатив задано 10 бінарних відношень R1-R10 матрицями відношень. Для кожного із БВ Ri необхідно:

1) перевірити наявність властивості ациклічності;

2) а) якщо відношення Ri є ациклічним - знайти множину Неймана Моргенштерна (отриманий результат обгрунтувати – показати, що отримана множина відповідає означенню розв'язка НейманаМоргенштерна);

б) якщо відношення Ri не ациклічне - знайти множини оптимальних альтернатив за принципом К-оптимізації (k=1, k=2, k=3, k=4)

Таблиця 2 – Результати виконання завдання 2

Відношення	**Ациклічне/ неациклічне**	**Розв’язок Неймана - Моргенштерна**	**Опт. альтернативи за принципом К-оптимізації**
R1	+	X^{HM} = {1, 2, 7}	
R2	–		1-max: {3, 6, 8, 12, 13} 1-opt: {3, 6, 8, 12, 13} 2-max: {3, 6, 13} 2-opt: {3, 6, 13} 3-max: {∅} 4-max: {3, 6, 13}
R3	+	X^{HM} = {3, 4, 6}	
R4	+	X^{HM} = {2, 5}	
R5	–		1-max: {1, 2, 3, 4, 6, 14} 2-opt: {1, 2, 3, 4, 6, 14} 2-max: {1, 2, 3, 4, 6, 14} 2-opt: {1, 2, 3, 4, 6, 14} 3-max: {1, 3, 4, 8, 11} 4-max: {1, 3, 4}
R6	+	X^{HM} = {2, 4, 6}	
R7	–		1-max: {1, 2, 3, 7, 15} 1-opt: {1, 2, 3, 7, 15} 2-max: {2, 7, 15} 2-opt: {2, 7, 15} 3-max: {∅} 4-max: {2, 7, 15}
R8	+	X^{HM} = {4, 5, 10, 13}	
R9	+	X^{HM} = {1, 5, 9}	
R10	+	X^{HM} = {6, 9, 15}	

Пояснення до завдання 1

1)

R	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1
3	0	1	1	1	1	1	1	1
4	0	0	0	1	1	1	1	0
5	0	0	0	1	1	1	1	0
6	0	0	0	1	1	1	1	0
7	0	0	0	1	1	1	1	0
8	0	1	1	1	1	1	1	1

R	1	2	3	4	5	6	7	8
1	I	P	P	P	P	P	P	P
2		I	I	P	P	P	P	I
3		I	I	P	P	P	P	I
4				I	I	I	I	
5				I	I	I	I	
6				I	I	I	I	
7				I	I	I	I	
8		I	I	I	I	I	I	I

Клас: квазіпорядок

У відношенні наявна симетрична частина

Оптимізація за домінуванням

$X^*_R = \{1\}$ – рядок містить всі одиниці

$X^{**}_R = \{1\}$ – стовпець містить усі нулі, крім головної діагоналі

Оптимізація за блокуванням

$X^0_R = \{1\}$ – стовпець містить лише I або 0

$X^{00}_R = \{1\}$ – стовпець містить усі 0, крім діагоналі

2)

R	1	2	3	4	5	6	7	8
1	0	1	1	1	0	1	1	1
2	0	0	1	0	0	1	0	0
3	0	0	0	0	0	0	0	0
4	0	1	1	0	0	1	1	1
5	1	1	1	1	0	1	1	1
6	0	0	1	0	0	0	0	0
7	0	1	1	0	0	1	0	1
8	0	1	1	0	0	1	0	0

R	1	2	3	4	5	6	7	8
1	N	P	P	P		P	P	P
2		N	P			P		
3			N					
4		P	P	N		P		P
5	P	P	P	P	N	P	P	P
6			P			N		
7		P	P			P	N	P
8		P	P			P		N

Клас: строгий порядок

Є асиметричність

Оптимізація за домінуванням

$X^*_P = \{5\}$ – в рядку всі одиниці, окрім головної діагоналі

Оптимізація за блокуванням

$X^0_P = \{5\}$ – стовпчик містить всі нулі

3)

R	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	1	0	1	0	1	0	0	0
3	0	0	0	0	0	0	0	0
4	1	1	1	0	1	1	0	0
5	0	0	0	0	0	0	0	0
6	1	0	1	0	1	0	0	0
7	1	1	1	0	1	1	0	0
8	1	1	1	0	1	1	0	0

R	1	2	3	4	5	6	7	8
1	N		N		N			
2	P	N	P		P	N		
3	N		N		N			
4	P	P	P	N	P	P	N	N
5	N		N		N			
6	P	N	P		P	N		
7	P	P	P	N	N	P	N	N
8	P	P	P	N	P	P	N	N

Клас: слабкого впорядкування

Є асиметричність

Оптимізація за домінуванням

$X^*_P = \{\emptyset\}$ – немає рядка, в якому всі одиниці, окрім головної діагоналі

Оптимізація за блокуванням

$X^0_P = \{4, 7, 8\}$ – стовпчики містять всі нулі

4)

R	1	2	3	4	5	6	7	8
1	0	0	1	0	0	0	0	0
2	0	1	1	0	1	0	1	1
3	1	0	0	1	1	0	0	1
4	1	0	0	0	1	0	1	0
5	1	0	1	1	0	1	1	0
6	1	0	1	1	1	1	1	0
7	0	0	0	0	0	0	1	0
8	0	1	1	1	0	1	0	0

R	1	2	3	4	5	6	7	8
1	N	N	I				N	N
2	N	I	P	N	P	N	P	I
3	I		N	N	I		N	I
4	P	N	N	N	I		P	
5	P		I	I	N	I	P	N
6	P	N	P	P	I	I	P	
7	N		N				I	N
8	N	I	I	P	N	P	N	N

Клас: Класу немає

У відношенні наявна симетрична частина

Оптимізація за домінуванням

$X^*_R = \{\emptyset\}$ – немає рядка, що містить всі одиниці

$X^{**}_R = \{\emptyset\}$ – відповідно немає сенсу шукати строго найбільший елемент

Оптимізація за блокуванням

$X^0_R = \{2, 8\}$ – стовпці містять лише I або 0

$X^{00}_R = \{\emptyset\}$ – немає стовпців, які містять усі 0, крім діагоналі

5)

R	1	2	3	4	5	6	7	8
1	1	1	0	0	0	0	0	1
2	1	1	0	0	0	0	0	1
3	0	0	1	0	1	0	1	0
4	0	0	0	1	0	1	0	0
5	0	0	1	0	1	0	1	0
6	0	0	0	1	0	1	0	0
7	0	0	1	0	1	0	1	0
8	1	1	0	0	0	0	0	1

R	1	2	3	4	5	6	7	8
1	I	I	N	N	N	N	N	I
2	I	I	N	N	N	N	N	I
3	N	N	I	N	I	N	N	N
4	N	N	N	I	N	I	N	N
5	N	N	I	N	I	N	I	N
6	N	N	N	I	N	I	N	N
7	N	N	N	N	I	N	I	N
8	I	I	N	N	N	N	N	I

Клас: еквівалентність

Відношення є симетричним

Оптимізація за домінуванням

$X^*_R = \{\emptyset\}$ – немає рядка, що містить всі одиниці

$X^{**}_R = \{\emptyset\}$ – відповідно немає сенсу шукати строго найбільший елемент

Оптимізація за блокуванням

$X^0_R = \{1, 2, 3, 4, 5, 6, 7, 8\}$ – усі стовпці містять лише I або 0

$X^{00}_R = \{\emptyset\}$ – немає стовпців, які містять усі 0, крім діагоналі

6)

R	1	2	3	4	5	6	7	8
1	1	0	0	0	0	1	0	0
2	1	1	1	1	1	1	1	0
3	1	0	1	0	0	1	1	0
4	1	0	1	1	1	1	1	0
5	1	0	1	0	1	1	1	0
6	0	0	0	0	0	1	0	0
7	1	0	0	0	0	1	1	0
8	1	1	1	1	1	1	1	1

R	1	2	3	4	5	6	7	8
1	I					P		
2	P	I	P	P	P	P	P	
3	P		I			P	P	
4	P		P	I	P	P	P	
5	P		P		I	P	P	
6						I		
7	P					P	I	
8	P	P	P	P	P	P	P	I

Клас: нестрогий порядок

У відношенні наявна симетрична частина

Оптимізація за домінуванням

$X^*_R = \{8\}$ – рядок містить всі одиниці

$X^{**}_R = \{8\}$ – стовпець містить усі нулі, крім головної діагоналі

Оптимізація за блокуванням

$X^0_R = \{8\}$ – стовпець містить лише I або 0

$X^{00}_R = \{8\}$ – стовпець містить усі 0, крім діагоналі

7)

R	1	2	3	4	5	6	7	8
1	1	1	1	1	1	0	1	0
2	0	1	1	1	1	1	1	0
3	0	0	1	0	1	1	0	0
4	0	1	0	1	1	1	0	1
5	0	0	1	1	0	1	0	1
6	1	1	0	0	0	1	1	0
7	0	0	0	0	1	1	0	1
8	0	0	1	1	0	1	0	1

R	1	2	3	4	5	6	7	8
1	I	P	P	P	P		P	N
2		I	P	I	P	I	P	N
3			I	N	I	P	N	
4		I	N	I	I	P	N	I
5			I	I	N	P		P
6	P	I				I	I	
7			N	N	P	I	N	P
8	N	N	P	I		P		I

Клас: класу немає

У відношенні наявна симетрична частина

Оптимізація за домінуванням

$X^*_R = \{\emptyset\}$ – немає рядка, що містить всі одиниці

$X^{**}_R = \{\emptyset\}$ – відповідно немає сенсу шукати строго найбільший елемент

Оптимізація за блокуванням

$X^0_R = \{\emptyset\}$ – немає стовпців, які містять лише I або 0

$X^{00}_R = \{\emptyset\}$ – немає стовпців, які містять усі 0, крім діагоналі

8)

R	1	2	3	4	5	6	7	8
1	1	1	1	1	0	1	1	1
2	0	1	0	1	0	1	1	0
3	1	1	1	1	0	1	1	1
4	0	1	0	1	0	1	1	0
5	1	1	1	1	1	1	1	1
6	0	1	0	1	0	1	1	0
7	0	1	0	1	0	1	1	0
8	1	1	1	1	0	1	1	1

R	1	2	3	4	5	6	7	8
1	I	P	I	P		P	P	I
2		I		I		I	I	
3	I	P	I	P		P	P	I
4		I		I		I	I	
5	P	P	P	P	I	P	P	P
6		I		I		I	I	
7		I		I		I	I	
8	I	P	I	P		P	P	I

Клас: квазіпорядок

У відношенні наявна симетрична частина

Оптимізація за домінуванням

$X^{*}_{R} = \{5\}$ – рядок містить усі одиниці

$X^{**}_{R} = \{5\}$ – стовпець містить усі нулі, крім головної діагоналі

Оптимізація за блокуванням

$X^{0}_{R} = \{5\}$ – стовпець містить лише I або 0

$X^{00}_{R} = \{5\}$ – стовпець містить усі 0, крім діагоналі

Пояснення до завдання 2

Для перевірки ациклічності відношення було застосовано алгоритм DFS (Пошук у глибину).

Алгоритм використовується для визначення, чи є бінарне відношення ациклічним з наступних причин:

- Глибинне вивчення графа: Алгоритм DFS дозволяє пройтися по графу вглиб від початкового вузла, перевіряючи всі можливі гілки до досягнення кінця. Це надає можливість досліджувати структуру графа в глибину та дійсно дізнатися, чи є в ньому цикли.
- Виявлення циклів: Під час виконання DFS ведеться відстеження відвіданих вузлів та стеку вузлів, які знаходяться на поточному шляху. Якщо під час обходу графа знайдено вузол, який вже є на стеку, це свідчить про наявність циклу в графі. Це дозволяє дізнатися, чи є бінарне відношення ациклічним.
- Ефективність: DFS є ефективним методом для визначення ациклічності графа. Для багатьох типів графів, DFS має оптимальну часову складність O(V + E), де V - це кількість вершин, а E - кількість ребер. Для бінарних відношень, де часто існує значно менше з'єднань, ніж всі можливі з'єднання, DFS може бути дуже ефективним.
- Простота реалізації: Алгоритм DFS відносно простий у реалізації та надає зрозумілий спосіб перевірки ациклічності графа. Він добре читається та легко розуміється.

R1. Розв'язок Неймана-Моргенштерна: $X^{НМ} = \{1, 2, 7\}$. Внутрішня та зовнішня стійкість виконуються.

1) Якщо відношення ациклічним, обираємо метод Неймана-Моргенштерна.

2) Необхідно сформувати множину S0, в яку будуть входити елементи, верхній переріз яких порожній (у матриці – у таких стовпчиках всі нулі).
3) Множина S1/S0 будується на основі S0 – включаємо елементи, для яких верхній переріз включається в множину S0 (у матриці у стовпчиках одиниці стоять лише в тих рядках, що входять до множини S0).

4) Операція повторюється для Sn+1/Sn, допоки Sn не буде дорівнювати всій множині альтернатив Ω.

```
Processing R1 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {1, 2}
S1\S0 = {5}
S2\S1 = {3}
S3\S2 = {4}
S4\S3 = {6}
S5\S4 = {7}
S6\S5 = {8}
S7\S6 = {11}
S8\S7 = {10}
S9\S8 = {9, 12}
S10\S9 = {13}
S11\S10 = {14}
S12\S11 = {15}
```

5) Шукаємо множини Q0..Qn. Q0 = S0. Для знаходження Q1 ми перевіряємо кандидатів до цієї множини з S1/S0. Умова полягає в тому, що верхній переріз кожного з цих кандидатів має бути порожнім у рядках Q0 (S0). Якщо ця умова для кандидата виконується, то ми додаємо його в множину Q1. Якщо множина Q1 виявиться порожньою, то Q1 = Q0. Потрібно повторити ці дії для всіх множин Q, і множина Qn буде розв'язком Неймана-Моргенштерна.

```
Q0 = [1, 2]
Q1 = [1, 2]
Q10 = [1, 2, 7]
Q11 = [1, 2, 7]
Q12 = [1, 2, 7]
Q2 = [1, 2]
Q3 = [1, 2]
Q4 = [1, 2]
Q5 = [1, 2, 7]
Q6 = [1, 2, 7]
Q7 = [1, 2, 7]
Q8 = [1, 2, 7]
Q9 = [1, 2, 7]

X_HM = {1, 2, 7}
```

6) Необхідно провести перевірку, отримана множина має мати дві властивості: внутрішня то зовнішня стійкість.

Внутрішня стійкість полягає в тому, що усі пари альтернатив в отриманій множині мають бути непорівнювальними (N).

```
Internal stability check:

(1, 1) = 0
(2, 1) = 0
(7, 1) = 0
(1, 2) = 0
(2, 2) = 0
(7, 2) = 0
(1, 7) = 0
(2, 7) = 0
(7, 7) = 0

Internal stability is achieved. The check has been passed.
```

Зовнішня стійкість полягає у тому, що кожна альтернатива, яка не потрапила у розв'язок, має мати вхідний зв'язок з будь-якої з вершин, що потрапила у розв'язок Неймана-Моргенштерна.

```
External stability check:

X_HM does not contain these vertexes = {3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15}
Element 3 has incoming connection from vertex 2.
Element 4 has incoming connection from vertex 1.
Element 5 has incoming connection from vertex 2.
Element 6 has incoming connection from vertex 2.
Element 8 has incoming connection from vertex 1.
Element 9 has incoming connection from vertex 1.
Element 10 has incoming connection from vertex 7.
Element 11 has incoming connection from vertex 1.
Element 12 has incoming connection from vertex 1.
Element 13 has incoming connection from vertex 1.
Element 14 has incoming connection from vertex 2.
Element 15 has incoming connection from vertex 1.

External stability is achieved. The check has been passed.
```

R2. K-оптимізація:
1-max: {3, 6, 8, 12, 13}
1-opt: {3, 6, 8, 12, 13}
2-max: {3, 6, 13}
2-opt: {3, 6, 13}
3-max: {∅}
4-max: {3, 6, 13}

1) Якщо відношення не є ациклічним, обираємо K-оптимізацію.
2) Представляємо відношення P, I, N на одній матриці.

```
Processing R2 matrix:
The graph contains cycles. K-optimization will be used.

Transformed matrix:
N N 0 N N 0 N N N N N N 0 N P
N I 0 N I 0 I N N I N N 0 I N
P P N P P N P N P P P N N P P
N N 0 N N 0 N N P N N N 0 N N
N I 0 N I 0 I N N I N N 0 I N
P P N P P N P N P P P N N P P
N I 0 N I 0 I N N I N N 0 I N
N N N N N N N N N N N I N N N
N N 0 0 N 0 N N N N N N 0 N N
N I 0 N I 0 I N N I N N 0 I N
N N 0 N N 0 N N N N N N 0 N N
N N N N N N N I N N N N N N N
P P N P P N P N P P P N N P P
N I 0 N I 0 I N N I N N 0 I N
0 N 0 N N 0 N N N N N N 0 N N
```

3) K1 - перевага альтернатив визначається строгою перевагою, рівноцінністю та непорівнюваністю за відношенням R (P, I, N).
Сім'я множин $S^1_R(x)$ представлена таким чином: кожен рядок матриці x визначає множину $S^1_R(x)$; якщо на перетині зі стовпчиком матриці стоїть «1», то цей елемент належить до множини $S^1_R(x)$.

```
Matrix S1:
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
```

4) Щоб обрати найкращу альтернативу, треба обрати множину, що є максимальною за включенням, тобто такий рядок включає в себе всі інші рядки і при цьому жоден інший рядок не включає його як власну підмножину. Таких альтернатив може бути декілька.

5) 1-оптимальні альтернативи – це такі альтернативи, які співпадають з усією множиною альтернатив Ω. Їх може не існувати.

```
S1(1) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(2) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(3) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(4) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(5) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(6) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(7) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(8) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(9) = [1, 2, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(10) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(11) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(12) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(13) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(14) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S1(15) = [2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]

K1_max = [3, 6, 8, 12, 13]
K1_opt = [3, 6, 8, 12, 13]
```

6) K2 - перевага альтернатив визначається строгою перевагою та непорівнюваністю за відношенням R (P, N).

Сім’я множин $S^2_R(x)$ представлена таким чином: кожен рядок матриці x визначає множину $S^2_R(x)$; якщо на перетині зі стовпчиком матриці стоїть «1», то цей елемент належить до множини $S^2_R(x)$.

```
Matrix S2:
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1]
[1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1]
[1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1]
[0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1]
```

7) Визначення k2-максимальних та k2-оптимальних множин відбувається так само, як і у випадку з K1.

```
S2(1) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S2(2) = [1, 4, 8, 9, 11, 12, 15]
S2(3) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(4) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S2(5) = [1, 4, 8, 9, 11, 12, 15]
S2(6) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(7) = [1, 4, 8, 9, 11, 12, 15]
S2(8) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15]
S2(9) = [1, 2, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S2(10) = [1, 4, 8, 9, 11, 12, 15]
S2(11) = [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]
S2(12) = [1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15]
S2(13) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(14) = [1, 4, 8, 9, 11, 12, 15]
S2(15) = [2, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15]

K2_max = [3, 6, 13]
K2_opt = [3, 6, 13]
```

8) К3 - перевага альтернатив визначається строгою перевагою та рівноцінністю за відношенням R (P, I).

Сім'я множин $S^3_R(x)$ представлена таким чином: кожен рядок матриці x визначає множину $S^3_R(x)$; якщо на перетині зі стовпчиком матриці стоїть «1», то цей елемент належить до множини $S^3_R(x)$.

```
Matrix S3:

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]
[1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]
[1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]
[0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]
[0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

9) Визначення k3-максимальних та k3-оптимальних множин відбувається так само, як і у випадку з К3.

```
S3(1) = [15]
S3(2) = [2, 5, 7, 10, 14]
S3(3) = [1, 2, 4, 5, 7, 9, 10, 11, 14, 15]
S3(4) = [9]
S3(5) = [2, 5, 7, 10, 14]
S3(6) = [1, 2, 4, 5, 7, 9, 10, 11, 14, 15]
S3(7) = [2, 5, 7, 10, 14]
S3(8) = [12]
S3(9) = []
S3(10) = [2, 5, 7, 10, 14]
S3(11) = []
S3(12) = [8]
S3(13) = [1, 2, 4, 5, 7, 9, 10, 11, 14, 15]
S3(14) = [2, 5, 7, 10, 14]
S3(15) = []

K3_max = []
K3_opt = []
```

10) K4 - перевага альтернатив визначається строгою перевагою за відношенням R (P).

Сім'я множин $S^3_R(x)$ представлена таким чином: кожен рядок матриці x визначає множину $S^3_R(x)$; якщо на перетині зі стовпчиком матриці стоїть «1», то цей елемент належить до множини $S^3_R(x)$.

```
S4(1) = [15]
S4(2) = []
S4(3) = [1, 2, 4, 5, 7, 9, 10, 11, 14, 15]
S4(4) = [9]
S4(5) = []
S4(6) = [1, 2, 4, 5, 7, 9, 10, 11, 14, 15]
S4(7) = []
S4(8) = []
S4(9) = []
S4(10) = []
S4(11) = []
S4(12) = []
S4(13) = [1, 2, 4, 5, 7, 9, 10, 11, 14, 15]
S4(14) = []
S4(15) = []

K4_max = [3, 6, 13]
K4_opt = []
```

11) Визначення k4-максимальних та k4-оптимальних множин відбувається так само, як і у випадку з K4.

```
Matrix S4:

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

R3**.** Розв’язок Неймана-Моргенштерна: $X^{HM} = \{3, 4, 6\}$. Внутрішня та зовнішня стійкість виконуються.

```
Processing R3 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {3}
S1\S0 = {2}
S2\S1 = {4, 6}
S3\S2 = {1}
S4\S3 = {5}
S5\S4 = {8, 7}
S6\S5 = {9, 12}
S7\S6 = {13}
S8\S7 = {14}
S9\S8 = {11}
S10\S9 = {10}
S11\S10 = {15}
```

```
Q0 = [3]
Q1 = [3]
Q10 = [3, 4, 6]
Q11 = [3, 4, 6]
Q2 = [3, 4, 6]
Q3 = [3, 4, 6]
Q4 = [3, 4, 6]
Q5 = [3, 4, 6]
Q6 = [3, 4, 6]
Q7 = [3, 4, 6]
Q8 = [3, 4, 6]
Q9 = [3, 4, 6]

X_HM = {3, 4, 6}
```

```
Internal stability check:

(3, 3) = 0
(4, 3) = 0
(6, 3) = 0
(3, 4) = 0
(4, 4) = 0
(6, 4) = 0
(3, 6) = 0
(4, 6) = 0
(6, 6) = 0

Internal stability is achieved. The check has been passed.
```

```
External stability check:

X_HM does not contain these vertexes = {1, 2, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15}
Element 1 has incoming connection from vertex 3.
Element 2 has incoming connection from vertex 3.
Element 5 has incoming connection from vertex 4.
Element 7 has incoming connection from vertex 3.
Element 8 has incoming connection from vertex 3.
Element 9 has incoming connection from vertex 3.
Element 10 has incoming connection from vertex 6.
Element 11 has incoming connection from vertex 3.
Element 12 has incoming connection from vertex 3.
Element 13 has incoming connection from vertex 4.
Element 14 has incoming connection from vertex 3.
Element 15 has incoming connection from vertex 4.

External stability is achieved. The check has been passed.
```

R4. Розв’язок Неймана-Моргенштерна: $X^{НМ} = \{2, 5\}$. Внутрішня та зовнішня стійкість виконуються.

```
Processing R4 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {5}
S1\S0 = {1}
S2\S1 = {2}
S3\S2 = {7}
S4\S3 = {3}
S5\S4 = {4}
S6\S5 = {9}
S7\S6 = {10}
S8\S7 = {8}
S9\S8 = {6}
S10\S9 = {11}
S11\S10 = {12}
S12\S11 = {13, 15}
S13\S12 = {14}
```

```
Q0 = [5]
Q1 = [5]
Q10 = [2, 5]
Q11 = [2, 5]
Q12 = [2, 5]
Q13 = [2, 5]
Q2 = [2, 5]
Q3 = [2, 5]
Q4 = [2, 5]
Q5 = [2, 5]
Q6 = [2, 5]
Q7 = [2, 5]
Q8 = [2, 5]
Q9 = [2, 5]

X_HM = {2, 5}
```

```
Internal stability check:

(2, 2) = 0
(5, 2) = 0
(2, 5) = 0
(5, 5) = 0

Internal stability is achieved. The check has been passed.
```

```
External stability check:

X_HM does not contain these vertexes = {1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
Element 1 has incoming connection from vertex 5.
Element 3 has incoming connection from vertex 2.
Element 4 has incoming connection from vertex 5.
Element 6 has incoming connection from vertex 5.
Element 7 has incoming connection from vertex 2.
Element 8 has incoming connection from vertex 5.
Element 9 has incoming connection from vertex 2.
Element 10 has incoming connection from vertex 2.
Element 11 has incoming connection from vertex 2.
Element 12 has incoming connection from vertex 2.
Element 13 has incoming connection from vertex 2.
Element 14 has incoming connection from vertex 5.
Element 15 has incoming connection from vertex 2.

External stability is achieved. The check has been passed.
```

R5. K-оптимізація:

1-max: {1, 2, 3, 4, 6, 14}2-opt: {1, 2, 3, 4, 6, 14}

2-max: {1, 2, 3, 4, 6, 14}
2-opt: {1, 2, 3, 4, 6, 14}
3-max: {1, 3, 4, 8, 11}
4-max: {1, 3, 4}

```
Processing R5 matrix:
The graph contains cycles. K-optimization will be used.

Transformed matrix:
N N N N P N P P P P P P P N P
N N N N N N N N N N N N N N N
N N N N P N P P P P P P P N P
N N N N P N P P P P P P P N P
0 N 0 0 N N N 0 N N 0 N N N N
N N N N N N N N N N N N N N N
0 N 0 0 N N N 0 N N 0 N N N N
0 N 0 0 P N P I P P I P P N P
0 N 0 0 N N N 0 N N 0 N N N N
0 N 0 0 N N N 0 N N 0 N N N N
0 N 0 0 P N P I P P I P P N P
0 N 0 0 N N N 0 N N 0 N N N N
0 N 0 0 N N N 0 N N 0 N N N N
N N N N N N N N N N N N N N N
0 N 0 0 N N N 0 N N 0 N N N N
```

```
S1(1) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(2) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(3) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(4) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(5) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S1(6) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(7) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S1(8) = [2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(9) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S1(10) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S1(11) = [2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(12) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S1(13) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S1(14) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(15) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]

K1_max = [1, 2, 3, 4, 6, 14]
K1_opt = [1, 2, 3, 4, 6, 14]
```

```
S2(1) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(2) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(3) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(4) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(5) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(6) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(7) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(8) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(9) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(10) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(11) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(12) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(13) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]
S2(14) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(15) = [2, 5, 6, 7, 9, 10, 12, 13, 14, 15]

K2_max = [1, 2, 3, 4, 6, 14]
K2_opt = [1, 2, 3, 4, 6, 14]
```

```
S3(1) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S3(2) = []
S3(3) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S3(4) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S3(5) = []
S3(6) = []
S3(7) = []
S3(8) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S3(9) = []
S3(10) = []
S3(11) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S3(12) = []
S3(13) = []
S3(14) = []
S3(15) = []

K3_max = [1, 3, 4, 8, 11]
K3_opt = []
```

```
S4(1) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S4(2) = []
S4(3) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S4(4) = [5, 7, 8, 9, 10, 11, 12, 13, 15]
S4(5) = []
S4(6) = []
S4(7) = []
S4(8) = [5, 7, 9, 10, 12, 13, 15]
S4(9) = []
S4(10) = []
S4(11) = [5, 7, 9, 10, 12, 13, 15]
S4(12) = []
S4(13) = []
S4(14) = []
S4(15) = []

K4_max = [1, 3, 4]
K4_opt = []
```

R6. Розв’язок Неймана-Моргенштерна: $X^{НМ} = \{2, 4, 6\}$. Внутрішня та зовнішня стійкість виконуються.

```
Processing R6 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {2}
S1\S0 = {1}
S2\S1 = {3}
S3\S2 = {4}
S4\S3 = {5}
S5\S4 = {6}
S6\S5 = {9}
S7\S6 = {7}
S8\S7 = {8}
S9\S8 = {10}
S10\S9 = {13}
S11\S10 = {11}
S12\S11 = {12}
S13\S12 = {14}
S14\S13 = {15}
```

```
Q0 = [2]
Q1 = [2]
Q10 = [2, 4, 6]
Q11 = [2, 4, 6]
Q12 = [2, 4, 6]
Q13 = [2, 4, 6]
Q14 = [2, 4, 6]
Q2 = [2]
Q3 = [2, 4]
Q4 = [2, 4]
Q5 = [2, 4, 6]
Q6 = [2, 4, 6]
Q7 = [2, 4, 6]
Q8 = [2, 4, 6]
Q9 = [2, 4, 6]

X_HM = {2, 4, 6}
```

```
Internal stability check:

(2, 2) = 0
(4, 2) = 0
(6, 2) = 0
(2, 4) = 0
(4, 4) = 0
(6, 4) = 0
(2, 6) = 0
(4, 6) = 0
(6, 6) = 0

Internal stability is achieved. The check has been passed.
```

```
External stability check:

X_HM does not contain these vertexes = {1, 3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15}
Element 1 has incoming connection from vertex 2.
Element 3 has incoming connection from vertex 2.
Element 5 has incoming connection from vertex 2.
Element 7 has incoming connection from vertex 4.
Element 8 has incoming connection from vertex 4.
Element 9 has incoming connection from vertex 2.
Element 10 has incoming connection from vertex 4.
Element 11 has incoming connection from vertex 4.
Element 12 has incoming connection from vertex 2.
Element 13 has incoming connection from vertex 2.
Element 14 has incoming connection from vertex 2.
Element 15 has incoming connection from vertex 2.

External stability is achieved. The check has been passed.
```

R7. K-оптимізація

1-max: {1, 2, 3, 7, 15}

1-opt: {1, 2, 3, 7, 15}

2-max: {2, 7, 15}

2-opt: {2, 7, 15}

3-max: {∅}

4-max: {2, 7, 15}

```
Processing R7 matrix:
The graph contains cycles. K-optimization will be used.

Transformed matrix:
N N I N N N N N N N N N N N N
N N N P P P N P P P P P P P N
I N N N N N N N N N N N N N N
N 0 N I I I 0 I N N N N N I 0
N 0 N I I I 0 I N N N N N I 0
N 0 N I I I 0 I N N N N N I 0
N N N P P P N P P P P P P P N
N 0 N I I I 0 I N N N N N I 0
N 0 N N N N 0 N N N N P N N 0
N 0 N N N N 0 N N N N N P N 0
N 0 N N N N 0 N N N N N N N 0
N 0 N N N N 0 N 0 N N N N N 0
N 0 N N N N 0 N N 0 N N N P 0
N 0 N I I I 0 I N N N N 0 I 0
N N N P P P N P P P P P P P N
```

```
S1(1) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(2) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(3) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(4) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(5) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(6) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(7) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S1(8) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(9) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(10) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(11) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S1(12) = [1, 3, 4, 5, 6, 8, 10, 11, 12, 13, 14]
S1(13) = [1, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14]
S1(14) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14]
S1(15) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]

K1_max = [1, 2, 3, 7, 15]
K1_opt = [1, 2, 3, 7, 15]
```

```
S2(1) = [1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(2) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(3) = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(4) = [1, 3, 9, 10, 11, 12, 13]
S2(5) = [1, 3, 9, 10, 11, 12, 13]
S2(6) = [1, 3, 9, 10, 11, 12, 13]
S2(7) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
S2(8) = [1, 3, 9, 10, 11, 12, 13]
S2(9) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S2(10) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S2(11) = [1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S2(12) = [1, 3, 4, 5, 6, 8, 10, 11, 12, 13, 14]
S2(13) = [1, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14]
S2(14) = [1, 3, 9, 10, 11, 12]
S2(15) = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]

K2_max = [2, 7, 15]
K2_opt = [2, 7, 15]
```

```
S3(1) = [3]
S3(2) = [4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S3(3) = [1]
S3(4) = [4, 5, 6, 8, 14]
S3(5) = [4, 5, 6, 8, 14]
S3(6) = [4, 5, 6, 8, 14]
S3(7) = [4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S3(8) = [4, 5, 6, 8, 14]
S3(9) = [12]
S3(10) = [13]
S3(11) = []
S3(12) = []
S3(13) = [14]
S3(14) = [4, 5, 6, 8, 14]
S3(15) = [4, 5, 6, 8, 9, 10, 11, 12, 13, 14]

K3_max = []
K3_opt = []
```

```
S4(1) = []
S4(2) = [4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S4(3) = []
S4(4) = []
S4(5) = []
S4(6) = []
S4(7) = [4, 5, 6, 8, 9, 10, 11, 12, 13, 14]
S4(8) = []
S4(9) = [12]
S4(10) = [13]
S4(11) = []
S4(12) = []
S4(13) = [14]
S4(14) = []
S4(15) = [4, 5, 6, 8, 9, 10, 11, 12, 13, 14]

K4_max = [2, 7, 15]
K4_opt = []
```

R8. Розв'язок Неймана-Моргенштерна: $X^{НМ} = \{4, 5, 10, 13\}$. Внутрішня та зовнішня стійкість виконуються.

```
Processing R8 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {4}
S1\S0 = {1}
S2\S1 = {3}
S3\S2 = {5}
S4\S3 = {6}
S5\S4 = {7}
S6\S5 = {10}
S7\S6 = {8}
S8\S7 = {2}
S9\S8 = {9}
S10\S9 = {12}
S11\S10 = {11}
S12\S11 = {13}
S13\S12 = {14}
S14\S13 = {15}
```

```
Q0 = [4]
Q1 = [4]
Q10 = [4, 5, 10]
Q11 = [4, 5, 10]
Q12 = [4, 5, 10, 13]
Q13 = [4, 5, 10, 13]
Q14 = [4, 5, 10, 13]
Q2 = [4]
Q3 = [4, 5]
Q4 = [4, 5]
Q5 = [4, 5]
Q6 = [4, 5, 10]
Q7 = [4, 5, 10]
Q8 = [4, 5, 10]
Q9 = [4, 5, 10]

X_HM = {10, 13, 4, 5}
```

```
Internal stability check:

(10, 10) = 0
(13, 10) = 0
(4, 10) = 0
(5, 10) = 0
(10, 13) = 0
(13, 13) = 0
(4, 13) = 0
(5, 13) = 0
(10, 4) = 0
(13, 4) = 0
(4, 4) = 0
(5, 4) = 0
(10, 5) = 0
(13, 5) = 0
(4, 5) = 0
(5, 5) = 0

Internal stability is achieved. The check has been passed.
```

```
External stability check:

X_HM does not contain these vertexes = {1, 2, 3, 6, 7, 8, 9, 11, 12, 14, 15}
Element 1 has incoming connection from vertex 4.
Element 2 has incoming connection from vertex 10.
Element 3 has incoming connection from vertex 4.
Element 6 has incoming connection from vertex 5.
Element 7 has incoming connection from vertex 4.
Element 8 has incoming connection from vertex 10.
Element 9 has incoming connection from vertex 10.
Element 11 has incoming connection from vertex 10.
Element 12 has incoming connection from vertex 4.
Element 14 has incoming connection from vertex 10.
Element 15 has incoming connection from vertex 4.

External stability is achieved. The check has been passed.
```

R9. Розв’язок Неймана-Моргенштерна: $X^{HM} = \{1, 5, 9\}$. Внутрішня та зовнішня стійкість виконуються.

```
Processing R9 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {1}
S1\S0 = {3}
S2\S1 = {2}
S3\S2 = {4}
S4\S3 = {5}
S5\S4 = {6}
S6\S5 = {7}
S7\S6 = {11}
S8\S7 = {8}
S9\S8 = {9}
S10\S9 = {10}
S11\S10 = {13}
S12\S11 = {14}
S13\S12 = {12}
S14\S13 = {15}
```

```
Q0 = [1]
Q1 = [1]
Q10 = [1, 5, 9]
Q11 = [1, 5, 9]
Q12 = [1, 5, 9]
Q13 = [1, 5, 9]
Q14 = [1, 5, 9]
Q2 = [1]
Q3 = [1]
Q4 = [1, 5]
Q5 = [1, 5]
Q6 = [1, 5]
Q7 = [1, 5]
Q8 = [1, 5]
Q9 = [1, 5, 9]

X_HM = {1, 5, 9}
```

```
Internal stability check:

(1, 1) = 0
(5, 1) = 0
(9, 1) = 0
(1, 5) = 0
(5, 5) = 0
(9, 5) = 0
(1, 9) = 0
(5, 9) = 0
(9, 9) = 0

Internal stability is achieved. The check has been passed.
```

```
External stability check:

X_HM does not contain these vertexes = {2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15}
Element 2 has incoming connection from vertex 1.
Element 3 has incoming connection from vertex 1.
Element 4 has incoming connection from vertex 1.
Element 6 has incoming connection from vertex 5.
Element 7 has incoming connection from vertex 1.
Element 8 has incoming connection from vertex 1.
Element 10 has incoming connection from vertex 9.
Element 11 has incoming connection from vertex 5.
Element 12 has incoming connection from vertex 5.
Element 13 has incoming connection from vertex 5.
Element 14 has incoming connection from vertex 1.
Element 15 has incoming connection from vertex 1.

External stability is achieved. The check has been passed.
```

R10. Розв’язок Неймана-Моргенштерна: $X^{HM} = \{6, 9, 15\}$. Внутрішня та зовнішня стійкість виконуються.

```
Processing R10 matrix:
The graph is acyclic. Neyman Morgenstern method will be used.

S0 = {6}
S1\S0 = {1}
S2\S1 = {2}
S3\S2 = {4}
S4\S3 = {5}
S5\S4 = {7}
S6\S5 = {3}
S7\S6 = {8}
S8\S7 = {10}
S9\S8 = {11, 12}
S10\S9 = {13}
S11\S10 = {14}
S12\S11 = {9, 15}
```

```
Q0 = [6]
Q1 = [6]
Q10 = [6]
Q11 = [6]
Q12 = [6, 9, 15]
Q2 = [6]
Q3 = [6]
Q4 = [6]
Q5 = [6]
Q6 = [6]
Q7 = [6]
Q8 = [6]
Q9 = [6]

X_HM = {9, 6, 15}
```

```
Internal stability check:

(9, 9) = 0
(6, 9) = 0
(15, 9) = 0
(9, 6) = 0
(6, 6) = 0
(15, 6) = 0
(9, 15) = 0
(6, 15) = 0
(15, 15) = 0

Internal stability is achieved. The check has been passed.
```

```
External stability check:

X_HM does not contain these vertexes = {1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 14}
Element 1 has incoming connection from vertex 6.
Element 2 has incoming connection from vertex 6.
Element 3 has incoming connection from vertex 6.
Element 4 has incoming connection from vertex 6.
Element 5 has incoming connection from vertex 6.
Element 7 has incoming connection from vertex 6.
Element 8 has incoming connection from vertex 6.
Element 10 has incoming connection from vertex 6.
Element 11 has incoming connection from vertex 6.
Element 12 has incoming connection from vertex 6.
Element 13 has incoming connection from vertex 6.
Element 14 has incoming connection from vertex 6.

External stability is achieved. The check has been passed.
```

Висновки:

У даній лабораторній роботі було застосовано 4 методи вирішення задачі оптимізації за бінарним відношенням:

1) Оптимізація за домінуванням – пошук множини максимальних альтернатив. Оптимізація за домінуванням може не дати потрібного результату, адже може не бути альтернатив, більш переважних за всі інші.
2) Оптимізація за блокуванням – пошук множини найбільших альтернатив. Це по суті пошук альтернатив, для яких не існує більш переважних альтернатив, ніж вони самі. Цей спосіб оптимізації також може не дати результатів.

3) Метод Неймана-Моргенштерна. Ми шукаємо множину, яка в сукупності є більш переважною за всі інші альтернативи. Це – множина найкращих альтернатив. Усі альтернативи, які потрапили в цю множину, мають бути непорівнювальні між собою. Даний алгоритм застосовується, тільки коли бінарне відношення є ациклічним.

4) K-оптимізація. Вибір найкращих альтернатив залежить від вибору аспекту порівняння альтернатив:

- K1 - перевага альтернатив визначається строгою перевагою, рівноцінністю та непорівнюваністю за відношенням R (P, I, N);
- K2 - перевага альтернатив визначається строгою перевагою та непорівнюваністю за відношенням R (P, N);
- K3 - перевага альтернатив визначається строгою перевагою та рівноцінністю за відношенням R (P, I);
- K4 - перевага альтернатив визначається строгою перевагою за відношенням R (P).

Даний алгоритм застосовується, коли бінарне відношення не є ациклічним.