

Практична робота

Тема роботи : *Задача про призначення (задачі вибору)*

Мета: ознайомитися з угорським методом та його застосуванням, набути практичні уміння та навички розв'язування задачі про призначення за допомогою алгоритму угорського методу.

Практичне завдання:

1. Розв'язати задачу про призначення як на максимум, так і на мінімум з заданою матрицею продуктивності праці/вартості робіт $C=(c_{ij})$
 - за допомогою угорського методу;
 - за допомогою процедури «Поиск решения» MS Excel/

Теоретичні питання.

1. Сформулювати змістовну постановку задачі про призначення.
2. Привести математичну модель задачі про призначення.
3. Сформулювати основні положення, на яких ґрунтується угорський метод розв'язування задачі про призначення.
4. Угорський метод призначень для розв'язання задачі про призначення на мінімум цільової функції. Обґрунтувати процедуру переходу від задачі максимізації цільової функції до задачі мінімізації.
5. Сформулювати ознаку оптимальності плану задачі.
6. Скільки розв'язків може мати задача про призначення?
7. За яке число ітерацій алгоритму отримано розв'язок задачі? Яка максимальна кількість ітерацій алгоритму можлива при розв'язанні задачі заданої Вам розмірності?

Постановка задачі

Необхідно виконати n робіт ($i = 1, \dots, n$). Для цього залучаються n виконавців ($j = 1, \dots, n$), кожен з яких здатен виконати будь-яку з робіт. Відомі продуктивності праці/витрати c_{ij} на виконання i -ї роботи j -м виконавцем. Потрібно призначити кожного виконавця на одну роботу так, щоб загальна продуктивність/ загальні витрати була/були максимальною/мінімальними.

Математична модель задачі

Змінні моделі: $x=(x_{ij})_{n \times n}$ – матриця призначень, де

$$x_{ij} = \begin{cases} 1, & \text{якщо на } i\text{-у роботу призначено } j\text{-ого виконавця;} \\ 0, & \text{у супротивному випадку.} \end{cases}$$

Цільова функція:

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max / \min \quad (1)$$

Система обмежень:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad i = \overline{1, n}, \quad j = \overline{1, n} \quad (4)$$

Основні положення, що лежать в основі угорського методу

1. Розв'язок задачі не зміниться, якщо до будь-якого стовпця або рядка додати (або відняти) певну сталу величину. Тобто, якщо план x^* — оптимальний план задачі (1)–(4), то він також буде оптимальним для цільової функції Z' з матрицею $C' = \|c'_{ij}\|$, де $c'_{ij} = c_{ij} \pm u_i \pm v_j$, $i=1, \dots, n$, $j=1, \dots, n$, $x \in \Omega$
2. Якщо всі $c'_{ij} \geq 0$, і знайдено план x^* , такий що

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^* = 0,$$

то x^* — оптимальний план.

Суть методу

Шляхом додавання певним чином знайдених чисел до окремих стовпців та віднімання з них інших чисел, формують так звану **систему незалежних нулів**. Набір нулів називається системою незалежних нулів, якщо жодні два (чи більше) нулі **не знаходяться в одному рядку або стовпці**. Якщо кількість незалежних нулів дорівнює **n**, то, присвоївши відповідним змінним $x_{ij} = 1$, а всім іншим — $x_{ij} = 0$, згідно з твердженням 2, отримаємо **оптимальний план призначення**.

Опис АЛГОРИТМУ УГОРСЬКОГО МЕТОДУ

Алгоритм складається з початкового кроку та **не більше ніж (n–2)** послідовно повторюваних ітерацій.

На **попередньому етапі**, якщо задача сформульована як задача **на максимум**, її перетворюють на еквівалентну задачу **на мінімум**. На цьому ж етапі виділяється система незалежних нулів.

Кожна наступна ітерація спрямована на збільшення (хоча б на 1) кількості незалежних нулів.

Як тільки кількість незалежних нулів **k** дорівнює **розмірності матриці (k = n)**, задача вважається розв'язаною.

Оптимальний план призначення визначається положенням незалежних нулів на останній ітерації.

Приклад: Нехай необхідно розв'язати задачу про призначення на $\max$

$$(\max Z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij}),$$

де

$$C = \begin{vmatrix} 10 & 7 & 6 & 9 \\ 3 & 7 & 5 & 1 \\ 5 & 6 & 8 & 3 \\ 6 & 11 & 7 & 5 \end{vmatrix}$$

Етап підготовки. Відшукуємо максимальний елемент у кожному стовпці матриці і обчислюємо кожний елемент матриці C з максимального елементу відповідного стовпця. В отриманій матриці C' відшукуємо мінімальний елемент в кожній зі строк і вираховуємо їх з відповідних елементів C' :

$$\begin{vmatrix} 10 & 7 & 6 & 9 \\ 3 & 7 & 5 & 1 \\ 5 & 6 & 8 & 3 \\ 6 & 11 & 7 & 5 \end{vmatrix} \quad C' = \begin{vmatrix} 0^* & 4 & 2 & 0 \\ 7 & 4 & 3 & 8 \\ 5 & 5 & 0 & 6 \\ 4 & 0 & 1 & 4 \end{vmatrix} \quad Co = \begin{vmatrix} 0^* & 4 & 2 & 0 \\ 4 & 1 & 0 & 5 \\ 5 & 5 & 0 & 6 \\ 4 & 0 & 1 & 4 \end{vmatrix}$$

Утворимо первісну систему незалежних нулів, відмічаючи їх зірочкою. Маємо матрицю Co . Число незалежних нулів в ній $k=3$. Так як $k < n$ ($n=4$), то переходимо до основного шагу.

Перша ітерація. В матриці 4 над стовпцями, які містять нулі з зірочками, ставимо знаки виділення (+). Знаходимо невиділений стовпець $j=4$. Шукаємо в ньому нульовий елемент. Це $C_{14} = 0$.

$$\begin{array}{c} \boxed{+} + + \\ \left| \begin{array}{cccc} 0^* & 4 & 2 & 0 \\ 4 & 1 & 0 & 5 \\ 5 & 5 & 0 & 6 \\ 4 & 0 & 1 & 4 \end{array} \right| \end{array}$$

Помічаємо його штрихом $C_{14} = 0'$. Продивляємося строку $i=1$. Вона містить помічений нуль $C_{11} = 0^*$. Помічаємо перший стовпець $\boxed{+}$.

Якщо тепер викреслити $j=2,3$ і помічену строку, то отримаємо матрицю, в якій нема нулів.

Позначимо множину непомічених строк через I_1 . Вочевидь, $I_1 = \{2,3,4\}$, а множина непомічених стовпців через I_1 , де $I_1 = \{1,4\}$. Знаходимо $\min\{C_{ij}\} = 4, i \in I_1, j \in I_1$.

Додамо число 4 до помічених стовпців і віднімемо його з непомічених строк. Отримаємо

$$\begin{array}{c} \boxed{+} + + \\ \left| \begin{array}{cccc} 0^* & 8 & 6 & 0' \\ 0 & 1 & 0^* & 1 \\ 1 & 5 & 0 & 2 \\ 0 & 0^* & 1 & 0 \end{array} \right| + \end{array}$$

В результаті такого перетворення отримаємо нові непомічені нулі $C_{21} = 0, C_{41} = 0$. Беремо будь-який з них і помічаємо '. Наприклад, $C_{21} = 0'$. Так як в рядку 2 маємо 0^* , то знімаємо помітку зі стовпця $j=3$.

$$\begin{array}{c} \boxed{+} + \boxed{+} \\ \left| \begin{array}{cccc} 0^* & 8 & 6 & 0' \\ 0' & 1 & 0^* & 1 \\ 1 & 5 & 0 & 2 \\ 0 & 0^* & 1 & 0 \end{array} \right| + \end{array}$$

Серед непомічених рядків матриці є непомічений нуль. Це $C_{41} = 0'$. Штрихуємо цей нуль.

В рядку $i=4$ маємо 0^* , знімаємо помітку зі стовпця $j=2$. Помічаємо строку $i=4$. Отримуємо:

$$\begin{array}{c} \boxed{+} \boxed{+} \boxed{+} \\ \left| \begin{array}{cccc} 0^* & 8 & 6 & 0' \\ 0' & 1 & 0^* & 1 \\ 1 & 5 & 0 & 2 \\ 0' & 0^* & 1 & 0 \end{array} \right| + \end{array}$$

Серед непомічених рядків знаходимо непомічений нуль

$C_{33} = 0$. Помічаємо цей нуль: $C_{33} = 0'$. В рядку $i=3$ немає нулів з зірочками. Тому треба будувати низку. Від останнього нуля зі штрихом $C_{33} = 0'$ по стовпцю переходимо до нуля з зірочкою $C_{23} = 0^*$. Від $C_{23} = 0^*$ по рядку $i=2$ переходимо до нуля зі штрихом $C_{21} = 0'$. Від $C_{21} = 0'$ по стовпцю $j=1$ переходимо до нуля з зірочкою $C_{11} = 0^*$. Від $C_{11} = 0^*$ переходимо по рядку $i=1$ до $C_{14} = 0'$. В стовпці $j=4$ нема нулів з зірочкою. Відповідно, будовання низки завершено. Отримаємо побудовану низку:

$$\begin{array}{c} \boxed{+} \boxed{+} \boxed{+} \\ \left| \begin{array}{cccc} 0^* & 8 & 6 & 0' \\ 0' & 1 & 0^* & 1 \\ 1 & 5 & 0' & 2 \\ 0' & 0^* & 1 & 0 \end{array} \right| + \end{array}$$

На низці $C_{33}C_{23}C_{21}C_{11}C_{14}$ нулі зі штрихами міняємо на нулі з зірочками. Зірочки над нулями, які не знаходяться на низці, зберігаємо. Всі інші знищуємо. Переходимо далі, де кількість незалежних нулів збільшується на одиницю, на цьому перша ітерація закінчується.

$$\begin{vmatrix} 0^* & 8 & 6 & 0^* \\ 0^* & 1 & 0^* & 1 \\ 1 & 5 & 0^* & 2 \\ 0^* & 0^* & 1 & 0 \end{vmatrix}$$

Друга ітерація: Підраховуємо кількість незалежних нулів: $k=4$, Так як $n=4$, то $k=n$. Задача розв'язана. Отримано оптимальний план задачі:

$$\begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

Відповідь. Потрібно призначити
для виконання роботи 1 працівника 4;
для виконання роботи 2 працівника 1;
для роботи 3 – працівника 3;
для роботи 4 – працівника 2.
Тоді продуктивність буде максимальною і становитиме 31 од.

РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРО ПРИЗНАЧЕННЯ ЗА ДОПОМОГОЮ ПРОЦЕДУРИ «ПОИСК РЕШЕНИЯ» MICROSOFT EXCEL FOR WINDOWS

	A	B	C	D	E	F	G	H	I
1	Матрица стоимостей								
2		10	7	6	9				
3		3	7	5	1				
4		5	6	8	3				
5		6	11	7	5				
6									
7	Матрица назначений								
8								=	1
9								=	1
10								=	1
11								=	1
12									
13									
14		=	=	=	=				
15		1	1	1	1			ЦФ	

Рис. 1. Подання вхідних даних на аркуші MS EXCEL

115	=СУММПРОИЗВ(B2:E5;B8:E11)								
	A	B	C	D	E	F	G	H	I
1	Матрица стоимостей								
2		10	7	6	9				
3		3	7	5	1				
4		5	6	8	3				
5		6	11	7	5				
6									
7	Матрица назначений								
8						=СУММ(B8:E8)	=		1
9						=СУММ(B9:E9)	=		1
10						=СУММ(B10:E10)	=		1
11						=СУММ(B11:E11)	=		1
12									
13		=СУММ(B8:B11)	=СУММ(C8:C11)	=СУММ(D8:D11)	=СУММ(E8:E11)				
14		=	=	=	=				
15		1	1	1	1			ЦФ	=СУММПРОИЗВ(B2:E5;B8:E11)

Рис. 2. Програмування комірок, що відповідають цільовій функції та лівим частинам обмежень

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению: ☐ минимальному значению

Изменяя ячейки:

Ограничения:

Параметры поиска решения

Максимальное время: секунд

Предельное число итераций:

Относительная погрешность:

Допустимое отклонение: %

Сходимость:

☒ Линейная модель ☐ Автоматическое масштабирование

☒ Неотрицательные значения ☐ Показывать результаты итераций

Оценки: ☒ линейная ☐ квадратичная

Разности: ☒ прямые ☐ центральные

Метод поиска: ☒ Ньютона ☐ сопряженных градиентов

Рис. 3. Экранна форма «Поиск решения» та «Параметры поиска решений»

	A	B	C	D	E	F	G	H	I
1		Матрица стоимостей							
2		10	7	6	9				
3		3	7	5	1				
4		5	6	8	3				
5		6	11	7	5				
6									
7		Матрица назначений							
8		0	0	0	1		1 =		1
9		1	0	0	0		1 =		1
10		0	0	1	0		1 =		1
11		0	1	0	0		1 =		1
12									
13		1	1	1	1				
14		=	=	=	=				
15		1	1	1	1			ЦФ	31

Рис. 4. Результаты работы процедуры «Поиска решения»

1.

3	10	5	9	16	8	17
6	8	11	8	18	19	20
7	13	10	3	4	14	18
5	9	6	21	12	17	22
5	4	11	6	13	14	11
17	7	12	13	16	17	9
13	0	8	8	10	12	17

2.

5	13	6	10	13	8	9
10	9	7	11	8	12	11
11	5	8	12	4	18	4
12	6	9	8	5	8	4
9	4	4	5	6	6	5
11	8	7	4	7	3	7
6	5	8	12	13	9	14

3.

6	5	9	10	7	12	8
9	7	11	6	8	11	10
8	10	7	8	10	7	4
5	6	10	5	6	11	12
4	9	8	9	4	1	2
5	6	10	11	10	12	5
4	11	5	4	5	12	13

4.

7	10	8	11	7	15	12
	2	5	6	10	18	4
8	11	9	2	16	3	6
5	2	5	14	3	10	5
8	7	6	7	13	8	14
6	8	17	10	11	9	5
15	18	5	9	12	6	10

5.

18	4	6	7	8	11	5
13	5	12	13	5	6	8
10	13	14	17	3	4	2
5	6	5	6	4	15	3
19	20	10	7	8	6	7
12	13	1	13	15	8	7
4	1	2	2	4	16	9

6.

4	5	9	5	6	14	6
8	12	4	13	16	15	16
2	15	8	10	17	7	9
14	8	4	9	5	6	7
3	5	4	12	10	11	13
10	9	11	5	6	12	8
7	13	8	12	8	11	10

7.

8	4	5	18	6	1	9
9	5	7	2	4	8	4
1	10	5	6	12	9	6
2	4	7	13	10	8	5
12	3	11	9	12	10	11
5	13	8	2	13	12	13
7	6	4	18	5	6	7

8.

13	4	5	12	3	6	14
7	1	9	4	11	2	10
12	4	7	6	8	7	4
5	4	6	1	7	8	6
8	9	9	7	10	3	5
7	4	4	5	4	3	8
15	3	3	13	5	6	16

9.

1	17	1	4	5	18	2
21	5	6	7	8	6	16
9	10	9	14	10	15	8
14	2	3	13	6	7	15
10	8	12	1	11	2	4
6	7	1	10	3	8	6
3	19	4	12	13	20	4

10.

1	4	5	8	9	4	5
5	6	7	8	10	11	12
4	18	4	7	6	7	8
5	4	3	6	10	4	5
9	10	8	9	5	13	6
6	8	11	12	7	8	9
12	4	5	6	2	5	4

11.

5	1	4	2	10	6	7
4	5	10	4	5	8	10
15	12	14	15	4	5	7
4	8	9	10	12	13	14
5	4	7	8	9	10	5
7	8	4	3	5	6	7
9	10	5	8	11	4	3

12.

12	13	8	5	5	16	17
1	2	6	7	8	3	4
6	7	2	16	3	9	10
4	5	15	20	19	11	4
10	1	2	18	17	3	5
5	6	4	10	5	7	8
18	19	11	12	14	14	15

13.

1	5	7	10	2	3	4
8	2	5	4	7	10	1
8	3	10	17	8	2	3
5	6	7	10	1	3	7
4	8	12	5	4	5	6
10	15	1	2	5	6	7
8	7	12	6	18	5	4

14.

3	5	10	7	8	10	12
4	6	7	4	5	6	7
12	13	11	6	7	8	9
10	4	5	8	9	4	5
8	7	9	5	6	7	11
1	3	12	1	4	5	6
4	10	11	13	15	16	8

15.

20	5	12	13	4	3	8
9	10	11	12	13	14	15
8	4	5	4	6	7	8
10	5	7	3	4	5	4
3	12	13	4	6	7	8
9	4	8	9	8	5	4
11	4	3	15	4	5	14

16.

1	5	2	10	3	12	4
6	7	8	9	10	12	5
8	3	4	5	6	8	9
1	2	13	4	5	6	7
8	9	4	5	6	8	10
3	4	8	7	12	13	1
4	5	1	2	3	8	10

17.

7	8	4	3	5	6	1
3	2	5	6	7	2	13
12	4	5	7	8	2	4
2	1	10	7	6	5	12
5	6	12	13	15	16	3
4	18	2	5	7	8	12
4	5	6	2	3	12	1

18.

8	4	3	1	12	13	5
4	2	5	3	4	5	6
1	4	2	5	6	7	8
9	4	5	6	7	8	9
10	11	12	13	14	15	16
5	6	7	8	9	10	4
6	1	2	13	3	4	5

19.

4	18	17	7	3	2	1
5	6	1	2	14	3	12
6	8	7	12	10	8	6
7	12	5	7	6	7	15
3	1	2	4	17	3	4
6	12	4	5	6	8	9
11	2	1	4	6	7	12

20.

20	5	1	8	9	10	18
16	12	4	5	6	7	8
13	15	16	8	9	10	6
7	8	12	5	7	8	9
5	7	9	10	11	12	6
12	5	7	8	9	4	5
7	4	3	4	5	1	2

21.

5	4	3	1	2	5	4
8	16	12	3	5	4	2
6	7	8	9	10	6	7
2	3	7	8	4	3	12
6	7	1	2	7	8	5
3	4	7	6	8	9	7
5	8	6	2	1	4	5

22.

1	5	3	4	7	6	10
12	2	4	7	8	9	10
13	3	5	6	4	3	2
1	4	5	8	9	12	15
25	5	1	3	8	10	6
7	3	6	1	6	5	1
12	5	8	5	10	4	26

23.

1	4	5	6	7	15	2
3	5	7	9	5	12	10
4	7	8	10	7	3	4
6	8	12	12	9	6	1
10	2	3	4	12	7	4
1	9	4	7	15	2	8
2	11	6	10	4	5	10

24.

12	4	5	6	7	2	1
5	7	8	9	10	11	8
6	3	4	5	6	7	9
8	7	2	11	12	13	2
6	5	1	5	6	7	6
1	4	5	7	12	4	5
8	9	14	5	6	7	4

25.

3	2	4	7	5	8	5
4	8	5	6	6	9	6
5	10	7	4	7	4	7
10	4	13	8	3	8	3
12	3	4	9	2	10	14
5	7	6	12	1	7	5
1	10	11	8	4	2	8

26.

4	5	6	7	9	4	1
3	7	10	6	8	5	7
4	8	5	7	16	8	9
5	5	3	5	4	2	3
3	6	4	2	5	4	5
1	7	2	6	8	1	2
4	8	1	7	9	6	7

27.

0	4	8	5	7	5	2
4	8	6	7	8	4	6
1	5	24	8	8	11	6
2	2	6	14	6	7	2
6	2	14	4	3	8	7
4	2	4	8	9	13	10
7	21	2	9	14	16	3

28.

3	4	5	9	7	8	13
5	6	7	8	12	14	8
9	7	8	9	10	11	12
6	12	15	16	5	4	6
7	8	9	10	4	5	7
8	10	12	7	8	9	10
15	16	17	18	19	15	4

29.

5	4	8	5	4	8	9
4	7	9	12	6	7	8
6	8	12	7	8	9	12
7	10	13	13	4	5	6
8	4	5	4	5	6	7
9	5	6	8	9	10	15
10	12	7	4	5	6	7

30.

5	4	3	6	12	4	5
8	9	10	7	8	13	4
6	7	8	9	12	6	7
8	10	4	8	13	4	5
6	12	5	4	7	8	12
7	8	9	12	5	6	7
6	7	8	9	13	14	15