Лекція 5. Задачі призначення. Угорський метод.

1.1 Змістовна постановка

Задано n різних робіт, кожну з яких може виконувати будь-який з n виконавців. Ефективність при виконанні роботи i виконавцем j дорівнює c_{ij}.

Потрібно розподілити виконавців по роботах, тобто призначити одного виконавця на кожну роботу таким чином, щоб максимізувати сумарну ефективність.

Формально задача про призначення може бути сформульована так. Необхідно вибрати з кожного рядка і кожного стовпця матриці

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \cdots & \cdots & \cdots \\ c_{11} & \cdots & c_{nn} \end{pmatrix}$$

рівно по одному елементу (всього n елементів) так, щоб їх сума була найбільшою. Таке завдання називається завданням вибору.

2.2 Математична модель

Для кожної i-ї роботи (i = 1, ..., n) і для кожного j-го виконавця (j = 1, ..., n) введемо змінну x_{ij}, яка може приймати всього два значення (0 або 1): $x_{ij} = \begin{cases} 1, & \text{якщо } i-\text{а робота виконується } j-\text{м виконавцем,} \\ 0, & \text{в іншому випадку.} \end{cases}$

Тоді сумарна ефективність виконання всіх робіт виражається функцією:

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} .$$

Обмеження задачі:

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, \ldots, n, \tag{2.1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, \ldots, n \tag{2.2}$$

інтерпретуються наступним чином. Рівняння (2.1) означають, що кожна i-я робота виконується рівно один раз. Рівняння (2.2) пред'являють вимоги до кожного j-му виконавцю: кожен j-й виконавець виконує рівно одну роботу.

Таким чином, математична модель задачі про призначення є завданням целочисленного лінійного програмування виду

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \to \max \tag{2.3}$$

при обмеженнях (2.1), (2.2) і

$$0 \le x_{ij} \le 1,\ i, j = 1, \ldots, n, \tag{2.4}$$

$$x_{ij} - \text{цілочислове},\ i, j = 1, \ldots, n. \tag{2.5}$$

Неважко бачити, що модель (2.1) - (2.4) є окремим випадком класичної транспортної задачі, в якій число постачальників збігається з числом споживачів (n), а запаси і потреби всіх пунктів збігаються і дорівнюють одиниці $\left(a_i = b_j = 1,\ i, j = 1, \ldots, n\right)$.

2.3 Венгерський метод для задачі про призначення

По суті угорський метод є уточненням двоїстого симплекс-методу стосовно транспортної задачі. Ідею методу висловив угорський математик Егерварі в 1931, в 1953 р Інший угорський математик - Г.Кун розвинув його ідею і назвав метод угорським.

Введемо наступні поняття:

1. Дві матриці $C = \{c_{ij}\}$ и $D = \{d_{ij}\}$ називаються *еквівалентними* $(C \approx D)$, якщо $c_{ij} = d_{ij} + \alpha_i + \beta_j$ для довільних i і j.

Завдання вибору, що визначаються еквівалентними матрицями, є еквівалентними.

2. Нульові елементи $z_1, z_2, ..., z_k$ матриці C називаються *незалежними* нулями, якщо для будь-якого нуля z_i, $0 \le i \le 1$, рядок і стовпець, на перетині якого лежить цей нуль z_i, не містять інші нульові елементи $z_L, L \ne i$

3. *Виділені елементи матриці* C - це елементи рядків або стовпців, позначених знаком +. Всі інші елементи матриці С - *невиділені елементи*.

Алгоритм складається з підготовчого етапу і не більше ніж (n-2) послідовно проведених ітерацій. Кожна ітерація складається з еквівалентних перетворень матриці і вибору максимального числа незалежних нулів. Остаточним результатом кожної ітерації є збільшення числа незалежних нулів, наявних на початку ітерації, на одиницю. Як тільки кількість незалежних нулів стає рівним n, завдання вибору виявляється вирішеною: оптимальний варіант визначається позиціями незалежних нулів в останній з матриць, еквівалентних вихідній матриці.

2.4 Алгоритм венгерського методу

Підготовчий етап

1. Знаходимо максимальний елемент в кожному стовпці матриці C.

2. Для кожного стовпця всі елементи цього стовпця послідовно віднімаємо з максимального, а результат залишаємо у відповідній позиції. В результаті отримуємо матрицю з неотрицательными елементами, у кожному стовпці якої є, принаймні, один 0.

3. З усіх елементів кожного рядка віднімаємо мінімальний елемент цього рядка. В результаті отримуємо матрицю C_0 з невід'ємними елементами, в кожному рядку і кожному стовпці якої є, щонайменше, один нуль.

4. Позначаємо незалежні нулі символом «*» за схемою:

а) в першому стовпці помічаємо довільний нуль;

б) у другому стовпці помічаємо (якщо знайдеться) той нуль, в рядку якого немає нуля, позначеного «*»;

в) аналогічно переглядаємо один за одним всі стовпці матриці C_0.

На цьому підготовчий етап закінчується.

Основний етап

Крок 0. Отримана матриця C_0. Якщо в матриці відзначено рівно n нулів, процес вирішення закінчується і оптимальне рішення визначається позиціями незалежних нулів (0*) матриці C_0.

Якщо ж число незалежних нулів <n, то помічаємо знаком + стовпці матриці C_0, що містять 0*. Відповідні елементи стовпців є виділеними елементами.

Крок 1. Якщо серед невиділених елементів матриці C_0 немає нульових, переходимо до кроку 3. Якщо невиділений нуль є, то може бути один з випадків:

а) рядок, що містить невиділений нуль, містить також 0*;

б) рядок, що містить невиділений нуль, не містить 0*.

У разі а) помічаємо знайдений нуль штрихом, помічаємо знаком + рядок, що містить цей 0', і прибираємо знак виділення + на колонку, який перетинається з виділеної рядком по 0*. Повертаємося на початок кроку 1.

У випадку б) помічаємо знайдений нуль штрихом і переходимо до кроку 2.

Таким чином, крок 1 закінчиться, коли:

1) або всі нулі будуть виділені - при цьому переходимо до кроку 3;

2) або виявиться невиділений нуль в рядку, де немає нуля із зірочкою - при цьому переходимо до кроку 2.

Крок 2. Будуємо послідовність з елементів 0' і 0* матриці за правилом:

а) послідовність починається з вихідного 0' (останнього нуля, позначеного штрихом);

б) від 0 'по стовпцю йдемо до 0* (якщо такий знайдеться);

в) від 0 * по рядку йдемо до 0';

г) повторюємо пункт б).

Отже, послідовність утворюється пересуванням від 0' до 0* по стовпці, від 0* до 0' по рядку і т.д. При цьому послідовність завжди починається з нуля зі штрихом (пункт а)) і закінчується нулем зі штрихом (завершення побудови послідовності можливо в пункті б) на елементі 0 ').

Отриману послідовність перетворимо так. Все 0* в послідовності замінюємо на звичайні нулі, все 0' в послідовності замінюємо на 0*. Потім чистимо матрицю C_0: прибираємо всі штрихи, все знаки виділення +. В результаті отримуємо матрицю C_0, в якій число незалежних нулів (0*) збільшено на одиницю. На цьому ітерація алгоритму завершується. Переходимо до кроку 0.

Крок 3. До цього кроку в матриці C_0 серед невиділених елементів немає нульових елементів. Тому:

а) вибираємо серед невиділених елементів мінімальний і позначаємо його h;

б) величину $h > 0$ віднімаємо з усіх елементів матриці C_0, розташованих в *виділених рядках*;

в) величину h прибавляємо до всіх елементів матриці C_0, розташованих в *виділених стовбчиках*.

В результаті отримаємо еквівалентну матрицю C_0' з невиділеними нульовими елементами. Покладаємо $C_0 = C_0'$ і переходимо до кроку 1.

2.5 Приклад розв'язку задачі про призначення венгерським методом

Задача. Розв'язати задачу про призначення, яка визначається матрицею

$$C = \begin{pmatrix} 3 & 4 & 2 & 2 & 1 \\ 4 & 5 & 3 & 1 & 3 \\ 4 & 3 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 & 2 \\ 1 & 3 & 1 & 2 & 1 \end{pmatrix}.$$

Розв'язок

При вирішенні задачі знак виділення +, підлягає знищенню, обводимо кружечком. Послідовність елементів 0' 0* на другому кроці вказуємо стрілками. 1. Підготовчий етап. Максимальний елемент першого стовпця матриці С дорівнює 4. Тому для отримання першого стовпця необхідно з 4 відняти елементи першого стовпця матриці. Аналогічно для отримання другого, третього, четвертого і п'ятого стовпців віднімаємо елементи цих стовпців з максимальних елементів 5, 3, 2 і 3 відповідно. В отриманій матриці мінімальний елемент кожного рядка дорівнює нулю, тому матриця C_0 знайдено. Помічаємо незалежні нулі: у першому стовпці вибираємо довільний шлях, тоді в другому і третьому стовпцях незалежних нулів немає, у четвертому стовпці помічаємо незалежний нуль, у п'ятому стовпці незалежного нуля немає. Слід зауважити, що в розглянутій задачі можливі й інші варіанти вибору незалежних нулів

$$
C=\begin{pmatrix} 3 & 4 & 2 & 2 & 1 \\ 4 & 5 & 3 & 1 & 3 \\ 4 & 3 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 & 2 \\ 1 & 3 & 1 & 2 & 1 \end{pmatrix} \quad \sim \quad C_0=\left(\begin{array}{c|cc|c|c} 1 & 1 & 1 & 0^* & 2 \\ 0^* & 0 & 0 & 1 & 0 \\ 0 & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \\ + & & & + & \end{array}\right)
$$

2. Перша ітерація

Крок 0. Виділяємо знаком + перший і четвертий стовпчики - вони містять 0*.

Крок 1. Переглядаємо невиділені нулі матриці C_0. Помічаємо штрихом нуль, розташований у другому рядку і в другому стовпці. Оскільки в цьому рядку міститься 0*, то маємо випадок а) і ставимо + на другий рядок, а знак + над першим стовпчиком обводимо кружком (прибираємо). Помічаємо штрихом нуль цього стовпця, який лежить в третьому рядку, що не містить 0*. Маємо випадок б), тому переходимо до кроку 2.

$$C_0 = \left(\begin{array}{c|cc|c|c} 1 & 1 & 1 & 0^* & 2 \\ 0^* & 0 & 0 & 1 & 0 \\ 0 & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \\ + & & & + & \end{array}\right) \sim C_0 = \begin{array}{c} \\ + \\ \\ \\ \\ \\ \end{array}\left(\begin{array}{ccccc} 1 & 1 & 1 & 0^* & 2 \\ 0^* & 0' & 0 & 1 & 0 \\ 0' & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \\ \oplus & & & + & \end{array}\right)$$

Крок 0 Крок 1

Крок 2. Будуємо послідовність: від останнього позначеного 0' рухаємося до 0* (перший стовпець, другий рядок), потім від 0* з другого рядка переходимо до 0', розташованому в тому ж рядку і в другому стовпці. Оскільки другий стовпець не містить 0*, то процес побудови послідовності закінчений. Знайдена послідовність складається з елементів: $c_{31} = 0', c_{21} = 0^*, c_{22} = 0'$. Преобразована послідовність набуває вигляду $c_{31} = 0^*, c_{21} = 0, c_{22} = 0^*$. Вносимо ці зміни в матрицю C_0, чистимо її (прибираємо знаки + і штрих) і отримуємо в результаті матрицю C_0, в якій число незалежних нулів (0*) збільшено на одиницю. На цьому перша ітерація алгоритму завершується.

$$C_0 = \begin{array}{c}\\ + \\ \\ \\ \\ \\ \end{array}\begin{pmatrix} 1 & 1 & 1 & 0^* & 2 \\ 0^* & 0' & 0 & 1 & 0 \\ 0' & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \\ & & & + & \end{pmatrix} \sim C_0 = \begin{pmatrix} 1 & 1 & 1 & 0^* & 2 \\ 0 & 0^* & 0 & 1 & 0 \\ 0^* & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \end{pmatrix} \sim$$

Крок 2

Оскільки подальші ітерації проводяться аналогічно, нижче наводяться результати обчислень без додаткових пояснень.

3. Друга ітерація

$$C_0 = \begin{pmatrix} 1 & 1 & 1 & 0^* & 2 \\ 0 & 0^* & 0 & 1 & 0 \\ 0^* & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \\ + & + & & + & \end{pmatrix} \sim C_0 = \begin{array}{c} \\ + \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} 1 & 1 & 1 & 0^* & 2 \\ 0 & 0^* & 0' & 1 & 0 \\ 0^* & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \\ + & \oplus & & + & \end{pmatrix} \sim$$

Крок 0 Крок 1

$$C_0' = \begin{array}{c} \\ + \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} 0 & 0 & 0 & -1^* & 1 \\ 0 & 0^* & 0' & 1 & 0 \\ -1^* & 1 & 1 & 0 & 1 \\ 0 & 4 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 & 0 \\ + & & + & & \end{pmatrix} \sim C_0' = \begin{pmatrix} 1 & 0 & 0 & 0^* & 1 \\ 1 & 0^* & 0' & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 & 1 \\ + & & & + & \end{pmatrix} \sim$$

Крок 3 а), б), h=1 Крок 3 в)

$$C_0 = \begin{array}{cc} \begin{matrix} + \\ + \\ \\ \\ \end{matrix} & \begin{pmatrix} 1 & 0' & 0 & 0^* & 1 \\ 1 & 0^* & 0' & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0' & 1 \end{pmatrix} \\ & \begin{matrix} + & & & \oplus & \end{matrix} \end{array} \sim C_0 = \begin{array}{c} \begin{pmatrix} 1 & 0' & 0 & 0^* & 1 \\ 1 & 0^* & 0' & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0' & 1 \end{pmatrix} \\ \begin{matrix} + & & & & \end{matrix} \end{array} \sim$$

Крок 1 Крок 2

$$C_0 = \begin{pmatrix} 1 & 0^* & 0 & 0 & 1 \\ 1 & 0 & 0^* & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0^* & 1 \end{pmatrix}$$

4. Третя ітерація

$$C_0 = \left(\begin{array}{cccc|c} 1 & 0^* & 0 & 0 & 1 \\ 1 & 0 & 0^* & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0' \\ 3 & 1 & 1 & 0^* & 1 \\ + & + & + & + & \end{array}\right) \sim C_0 = \begin{pmatrix} 1 & 0^* & 0 & 0 & 1 \\ 1 & 0 & 0^* & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0 & 0 & 0^* \\ 3 & 1 & 1 & 0^* & 1 \end{pmatrix}$$

Крок 0, Крок 1 Крок 2

Після третьої ітерації кількість незалежних нулів (0*) стало рівним розмірності n матриці і тому процес вибору закінчений. Шукані елементи матриці С відповідають позиціям незалежних нулів матриці C_0.

Значення цільової функції

$$L(x)=c_{12}+c_{23}+c_{31}+c_{45}+c_{54}=4+3+4+2+2=15.$$

Оптимальне рішення задачі про призначення з відповідною матрицею С визначається набором $\{x_{ij}\},\ i,j=1,\dots,5$ зі значеннями: $x_{12}=x_{23}=x_{31}=x_{45}=$ $=x_{54}=1$. Змістовно це означає, що першу роботу виконує другий виконавець, другий - третій виконавець, третю роботу - перший виконавець, четверту - п'ятий виконавець і п'яту роботу виконує четвертий виконавець.