

МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ О НАЗНАЧЕНИЯХ. ВЕНГЕРСКИЙ МЕТОД РЕШЕНИЯ ЗАДАЧИ

Необходимо выполнить n ($i=1, n$) работ. Для этого используются n ($j=1, n$) исполнителей, каждый из которых в состоянии выполнять любую работу. Известны затраты c_{ij} на выполнение i -той работы j -тым исполнителем. Требуется назначить каждого исполнителя на одну работу так, чтобы минимизировать суммарные затраты.

Пусть $x=(x_{ij})_{n \times n}$ – матрица назначений, где

x_{ij} – переменная величина, отражающая факт назначения

$$x_{ij} = \begin{cases} 1, & \text{если на } i\text{-ю работу назначен } j\text{-й исполнитель} \\ 0, & \text{в противном случае} \end{cases}$$

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

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

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

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

Основные утверждения, лежащие в основе венгерского метода:

1. Решение задачи не изменится, если к любому столбцу или строке прибавить (или вычесть) некоторую компоненту, т.е. если план x^* -- оптимальный план задачи (1)-(4), то он также оптимален для функции цели Z' с матрицей

$$C' = \|c'_{ij}\|, \text{ где}$$

$$c'_{ij} = c_{ij} \pm u_i \pm v_j, \quad i = \overline{1, n}, \quad j = \overline{1, n}, \quad x \in \Omega$$

2. Если все $c_{ij} > 0$ и найден план x^* , такой, что

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

то x^* - оптимальный план.

Суть метода: Путем прибавления определенным образом найденных чисел к некоторым столбцам и вычитания из них некоторых чисел находят систему так называемых *независимых нулей*. Набор нулей называется *системой независимых нулей*, если никакие два (или больше) нуля не лежат на одной линии (в строке или столбце). Если число независимых нулей равно n , то, приняв соответствующие им переменные x_{ij} равными 1, а все остальные – равными 0, согласно утверждению 2, получим оптимальный план назначения.

Описание АЛГОРИТМА ВЕНГЕРСКОГО МЕТОДА

Алгоритм состоит из предварительного шага и не более, чем $(n-2)$ последовательно повторяющихся итераций. На предварительном этапе в случае решения задачи на максимум, ее преобразуют в эквивалентную задачу на минимум. На этом же этапе выделяется система независимых нулей. Каждая последующая итерация направлена на

увеличение хотя бы на 1 числа независимых нулей. Как только число независимых нулей k станет равным размерности матрицы ($k=n$), задача решена.

Оптимальный план назначения определится положением независимых нулей на последней итерации.

Подготовительный этап.

Пусть исходная задача имеет вид

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

Находим $l_i = \max\{c_{ij}\}$. Заменяем задачу на максимум задачей на минимум:

$$Z'(x) = \sum_{i=1}^n \sum_{j=1}^n c'_{ij} x_{ij} \rightarrow \min,$$

где $c'_{ij} = l_j - c_{ij}$. Далее в каждой строке матрицы C' находим минимальный элемент и вычитаем его из элементов соответствующей строки: $c'' = \|c'_{ij} - \min_{j..} c'_{ij}\| = \|c''_{ij}\|$. В результате преобразования в каждой строке (и столбце) матрицы C' появится хотя бы один 0.

Подготовительный этап заканчивается выделением системы независимых нулей. Для этого в первом столбце матрицы выделяется произвольный 0 и отмечается *. Затем рассматриваются нули второго столбца. Если среди них есть такие, которые находятся не в одной строке с нулем первого столбца, уже отмеченного *, то один из них отмечается *. Аналогично рассматриваются нули для других столбцов. Отмеченные * нули независимы (по определению). Если число независимых нулей равно n , то согласно утверждению 2, задача решена.

Последующие итерации

Цель каждого последующего шага (итерации) – увеличение числа независимых нулей. Перед началом итерации выделяются знаком «+» столбцы матрицы, содержащие 0^* . Т.к. $k < n$, то не все столбцы окажутся выделенными. Устанавливаем, имеется ли среди невыделенных элементов матрицы хотя бы один 0.

Возможны следующие случаи:

1. Среди невыделенных элементов имеется хотя бы один 0. Проверяем, содержит ли строка с невыделенным нулем также 0^* .

а) если да, то невыделенный ноль отмечается $0'$, содержащая этот ноль строка отмечается справа знаком «+» и снимается знак выделения «+» над столбцом, в котором расположен 0^* , лежащий в только что выделенной строке. Снятие «+» производится знаком

$\oplus$.

б) если в строке с невыделенным нулем нулей со * нет, то невыделенный ноль также отмечается $0'$.

2. Среди невыделенных элементов нет нулей. Среди невыделенных элементов матрицы выбираем минимальный. Он вычитается из элементов, расположенных в невыделенных строках, и прибавляется к элементам, лежащим в выделенных столбцах. При этом новая матрица, согласно утверждению 1, эквивалентна исходной по отношению к оптимальному назначению.

Итерация оканчивается, если в строке с невыделенным нулем нет 0^* . Невыделенный ноль, как говорилось, отмечается $0'$, и строится цепочка элементов по правилу: движение начинается от $0'$ по столбцу к 0^* , от 0^* по строке переходим к $0'$, вновь от $0'$ к 0^* по столбцу и т.д. начало и конец цепочки – нули с ' '. если случится, что в одном столбце с исходным $0'$ нет 0^* , то цепочка вырождена.

После построения цепочки * над нулями уничтожаются, а ' заменяются на *. Все остальные * над нулями, не находящимися на цепочке, сохраняются. Т.к. в цепочке 0' на один больше, чем 0*, то в результате общее число 0* увеличится на единицу. На этом одна итерация заканчивается.

Подсчитываем количество 0* (независимых нулей). Если $k=n$, то найден оптимальный план назначения. Иначе переходим к следующей итерации.

ПРИМЕР

Пусть надо решить задачу о назначениях на максимум

$$\max: Z = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij},$$

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

Подготовительный этап. Отыскиваем максимальный элемент в каждом из столбцов матрицы и вычитаем каждый элемент матрицы C из максимального элемента соответствующего столбца. В полученной матрице C' отыскиваем минимальный элемент в каждой из строк и вычитаем их из соответствующих элементов матрицы C' :

$$1). C = \begin{pmatrix} 10 & 7 & 6 & 9 \\ 3 & 7 & 5 & 1 \\ 5 & 6 & 8 & 3 \\ 6 & 11 & 7 & 5 \\ \hline 10 & 11 & 8 & 9 \end{pmatrix} \quad 2). C' = \begin{pmatrix} 0 & 4 & 2 & 0 \\ 7 & 4 & 3 & 8 \\ 5 & 5 & 0 & 6 \\ 4 & 0 & 1 & 4 \end{pmatrix} \left| \begin{array}{l} 0 \\ 3 \\ 0 \\ 0 \end{array} \right.$$

$$3). C_0 = \begin{pmatrix} 0^* & 4 & 2 & 0 \\ 4 & 1 & 0^* & 5 \\ 5 & 5 & 0 & 6 \\ 4 & 0^* & 1 & 4 \end{pmatrix}$$

Образуем первоначальную систему независимых нулей, отмечая их звездочкой. Имеем матрицу C_0 . Число независимых нулей $k=3$. Так как $k < n=4$, то переходим к основному повторяющемуся шагу.

Первая итерация. В матрице 4 над столбцами, содержащими 0*, ставим знак «+». Находим невыделенный столбец $j=4$. Ищем в невыделенном столбце нулевой элемент. Это $c_{14}=0$. Помечаем его штрихом $c_{14}=0'$. Просматриваем строку $i=1$. В ней есть выделенный ноль $c_{11}=0^*$. Снимаем выделение первого столбца знаком $\oplus$. выделяем первую строку знаком «+». Получаем матрицу 5.

$$4). \quad \begin{array}{cccc} + & + & + & \\ \begin{pmatrix} 0^* & 4 & 2 & 0 \\ 4 & 1 & 0^* & 5 \\ 5 & 5 & 0 & 6 \\ 4 & 0^* & 1 & 4 \end{pmatrix} \end{array} \quad 5). \quad \begin{array}{cccc} \oplus & + & + & \\ \begin{pmatrix} 0^* & 4 & 2 & 0' \\ 4 & 1 & 0^* & 5 \\ 5 & 5 & 0 & 6 \\ 4 & 0^* & 1 & 4 \end{pmatrix} \end{array} +$$

Если в матрице 5 вычеркнуть выделенные столбцы $j=2,3$ и выделенную строку, то получим матрицу, в которой нет нулей. Обозначим множество невыделенных строк через I_1 . Очевидно, что $I_1=\{2,3,4\}$, а множество невыделенных столбцов через J_1 , здесь $J_1=\{1,4\}$. Находим $\min\{c_{ij}\}=4$, $i \in I_1, j \in J_1$.

Прибавим число 4 к выделенным столбцам и вычтем его из невыделенных строк, получим матрицу 6:

$$6). \quad \begin{array}{cccc} \oplus & + & + & \\ \begin{pmatrix} 0^* & 8 & 6 & 0' \\ 0 & 1 & 0^* & 1 \\ 1 & 5 & 0 & 2 \\ 0 & 0^* & 1 & 0 \end{pmatrix} \end{array} +$$

В результате такого преобразования получим новые невыделенные нули $C_{21}=0$, $C_{41}=0$, $C_{44}=0$. Берем любой из них и отмечаем его штрихом. Например, $C_{21}=0'$. Так как в строке $i=2$ имеется 0^* , то снимаем выделение столбца $j=3$. Перейдем к матрице 7:

$$7). \quad \begin{array}{cccc} \oplus & + & \oplus & \\ \begin{pmatrix} 0^* & 8 & 6 & 0' \\ 0' & 1 & 0^* & 1 \\ 1 & 5 & 0 & 2 \\ 0 & 0^* & 1 & 0 \end{pmatrix} \end{array} +$$

Среди невыделенных рядов матрицы есть невыделенный ноль. Это $C_{41}=0$. Штрихуем этот ноль $C_{41}=0'$. В строке $i=4$ имеется 0^* , снимаем выделение столбца $j=2$. Выделяем строку $i=4$. Получаем матрицу 8:

$$8). \quad \begin{array}{cccc} \oplus & \oplus & \oplus & \\ \begin{pmatrix} 0^* & 8 & 6 & 0' \\ 0' & 1 & 0^* & 1 \\ 1 & 5 & 0' & 2 \\ 0' & 0^* & 1 & 0 \end{pmatrix} \end{array} +$$

Среди невыделенных рядов находим невыделенный ноль $C_{33}=0$. Штрихуем этот ноль $C_{33}=0'$. В строке $i=3$ нет 0^* , поэтому нужно строить цепочку. От последнего нуля со штрихом $C_{33}=0'$ по столбцу переходим к нулю со звездочкой $C_{23}=0^*$. От $C_{23}=0^*$ по строке $i=2$ переходим к нулю со штрихом $C_{21}=0'$. От $C_{21}=0'$ по столбцу $j=1$ переходим к нулю со

звездочкой $C_{11}=0^*$. От $C_{11}=0^*$ переходим по строке $i=1$ к $C_{14}=0'$. В столбце $j=4$ нет 0^* . Следовательно, построение цепочки закончено. Построенная цепочка показана в матрице 9:

$$9). \quad \oplus \quad \oplus \quad \oplus$$

$$C = \begin{pmatrix} 0^* & 8 & 6 & 0' \\ 0' & 1 & 0^* & 1 \\ 1 & 5 & 0' & 2 \\ 0' & 0^* & 1 & 0 \end{pmatrix}^+$$

На цепочке $C_{33}C_{23}C_{21}C_{11}C_{14}$ нули со штрихом заменяем на нули со звездочкой. Звездочки над нулями, не находящимися на цепочке, сохраняем. Все остальные выделения уничтожаем. Переходим к матрице 10, в которой число независимых нулей увеличивается на единицу. На этом первая итерация заканчивается:

$$10) \quad \begin{pmatrix} 0 & 8 & 6 & 0^* \\ 0^* & 1 & 0 & 1 \\ 1 & 5 & 0^* & 2 \\ 0 & 0^* & 1 & 0 \end{pmatrix}$$

Вторая итерация. Подсчитываем число независимых нулей $k=4$. Так как $n=4$, то $k=n$. Задача решена. Оптимальный план представлен в матрице 11:

$$11). \quad x^* = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

РЕШЕНИЕ ЗАДАЧИ О НАЗНАЧЕНИЯХ С ПОМОЩЬЮ ПРОЦЕДУРЫ «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. Представление исходных данных на листе EXEL

115									
	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. Программирование ячеек, соответствующих целевой функции и левым частям системы ограничений

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению: 0

☐ минимальному значению

Изменяя ячейки:

Ограничения:

- \$B\$13:\$E\$13 = \$B\$15:\$E\$15
- \$B\$8:\$E\$11 = целое
- \$G\$8:\$G\$11 = \$I\$8:\$I\$11

Дополнительные кнопки: Предположить, Параметры, Добавить, Изменить, Удалить, Восстановить, Справка, Выполнить, Закрыть.

Параметры поиска решения

Максимальное время: 100 секунд

Предельное число итераций: 100

Относительная погрешность: 0,000001

Допустимое отклонение: 5 %

Сходимость: 0,0001

☒ Линейная модель ☐ Автоматическое масштабирование

☒ Неотрицательные значения ☐ Показывать результаты итераций

Оценки: ☒ линейная ☐ квадратичная

Разности: ☒ прямые ☐ центральные

Метод поиска: ☒ Ньютона ☐ сопряженных градиентов

Кнопки: ОК, Отмена, Загрузить модель..., Сохранить модель..., Справка.

Рис. 3. Экранная 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»

ВАРИАНТЫ ЗАДАНИЙ

1.

3	10	5	9
6	8	11	8
7	13	10	3
5	9	6	21

2.

5	13	6	10
10	9	7	11
11	5	8	12
12	6	9	8

3.

6	5	9	10
9	7	11	6
8	10	7	8
5	6	10	5

4.

7	10	8	11
	2	5	6
8	11	9	2
5	2	5	14

5.

18	4	6	7
13	5	12	13
10	13	14	17
5	6	5	6

6.

4	5	9	5
8	12	4	13
2	15	8	10
14	8	4	9

7.

8	4	5	18
9	5	7	2
1	10	5	6
2	4	7	13

8.

13	4	5	12
7	1	9	4
12	4	7	6
5	4	6	1

9.

1	17	1	4
21	5	6	7
9	10	9	14
14	2	3	13

10.

1	4	5	8
5	6	7	8
4	18	4	7
5	4	3	6

11.

5	1	4	2
4	5	10	4
15	12	14	15
4	8	9	10

12.

12	13	8	5
1	2	6	7
6	7	2	16
4	5	15	20

13.

1	5	7	10
8	2	5	4
8	3	10	17
5	6	7	10

14.

3	5	10	7
4	6	7	4
12	13	11	6
10	4	5	8

15.

20	5	12	13
9	10	11	12
8	4	5	4
10	5	7	3

16.

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

17.

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

18.

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

19.

4	18	17	7
5	6	1	2
6	8	7	12
7	12	5	7

20.

20	5	1	8
16	12	4	5
13	15	16	8
7	8	12	5

21.

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

22.

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

23.

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

24.

12	4	5	6
5	7	8	9
6	3	4	5
8	7	2	11

25.

3	2	4	7
4	8	5	6
5	10	7	4
10	4	13	8

26.

4	5	6	7
3	7	10	6
4	8	5	7
5	5	3	5

27.

0	4	8	5
6	2	14	4
4	2	4	8
7	21	2	9

28.

3	4	5	9
5	6	7	8
9	7	8	9
6	12	15	16

29.

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

30.

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