3-я итерация.

Размечем матрицу C_1 и приступаем к поиску.

1-й этап, поисковый.

Выделим штрихом нуль с координатами (1, 4). Он стоит в строке с нулевой невязкой, поэтому строку выделяем плюсом. Просматриваем её пересечения с выделенными столбцами.

На пересечении с выделенным столбцом – существенный нуль, отмечаем его звёздочкой, снимаем выделение столбца, заключая его в скобки. Продолжаем поиск в третьем столбце, отмечаем нуль (3, 3) штрихом. Его невязка тако же нулевая, отмечаем строку плюсом, просматриваем пересечения с выделенными столбцами.

По результатам просмотра нуль (3, 2) получает звезду, а с третьего столбца – снимается выделение. Больше в невыделенной части нулей нет. На этом поиск заканчивается неудачей.

	+	(+)	(+)		
$C_1 =$	5	8	$\bar{0}^*$	$\bar{0}$`	+
	$\bar{0}$	3	6	4	
	5	$\bar{0}^*$	$\bar{0}$`	2	+

					δ_i
$X_2 =$			70	90	0
	120				20
		50	120		0
δ_j	0	0	0	20	

3-й этап, эквивалентные преобразования матрицы C_1.

В невыделенной части матрицы C_1 (выделена серым, извиняюсь за каламбур) отыскиваем положительный элемент для коррекции $h = 3$. В результате имеем матрицу C_2, показанную ниже.

	+				
$C_2 =$	8	8	$\bar{0}^* \rightarrow$	$\rightarrow \bar{0}$`	+
	$\bar{0}$	$\downarrow 0$`	$\uparrow 3$	1	
	8	$\bar{0}^* \rightarrow$	$\uparrow \bar{0}$`	2	+

					δ_i
$X_2 =$			50	110	0
	120	20			0
		30	140		0
δ_j	0	0	0	0	

Приступаем к поиску, 1-й этап.

Отмечаем штрихом нуль (2, 2). Его невязка по строке показывает, что этот нуль является искомым.

Этап 2-й.

Строим цепочку (2, 2)` $\Rightarrow$ (3, 2)$^*$ $\Rightarrow$ (3, 3)` $\Rightarrow$ (1, 3)* $\Rightarrow$ (1, 4)`, она показана фоном и стрелками. Корректирующий элемент

$$\Delta = \min\{\delta_2 = 20,\ \delta_4 = 20,\ x_{3,2} = 50,\ x_{13} = 70\} = 20.$$

Полученный план перевозок X_2 показан рядом с матрицей C_2 его невязка нулевая.

Следовательно, достигнут оптимум, можно рассчитать целевую функцию $L = 1330$.

Это столько же, как и в методе потенциалов. Планы перевозок, полученных обоими методами, совпадают.

Замечания по процессу решения.

1. Текущие матрицы C и X изменяются не синхронно. В принципе, возможны решения, когда в ходе нескольких итераций матрица C остаётся одним и тем же, а план X – неоднократно меняется.

2. Известны случаи, когда при равенстве целевых функций, при одном и том же условии, решённом венгерским методом и методом потенциалов, получаются разные планы перевозок. Это есть своеобразная иллюстрация теорем линейного программирования (смотри раздел ранее).

2.3.4.7. Алгоритм решения ТЗ с ограниченной пропускной способностью коммуникаций [20, 24, 28 – 30]

Общая постановка транспортной задачи, ранее нами рассмотренная, характерна тем, что возможность перевозки $x_{i,j}$ единиц товара от i-го поставщика j-му потребителю физически ничем не ограничивается. Однако повседневная практика перевозок может повлечь за собой ограничения на количество перевозимого груза.

Указанные ограничения могут иметь различную физическую или иную природу:

- быть связанными с затратами на горючее (лимитироваться);
- определяться грузоподъёмностью транспортных средств, их проходимостью или предельным числом рейсов;
- диктоваться соотношением полезного груза в функции от дистанции перевозки, как это бывает, например, в дальнебомбардировочной авиации;
- и т.п.

  В таких случаях говорят о транспортной задаче с ограниченными пропускными способностями коммуникаций. Её математическая модель несколько отличается от общеизвестной (2.32) – (2.34), в части дополнительных ограничений:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j};$$

$$\sum_{j=1}^{m} x_{i,j} = a_i; \sum_{i=1}^{n} x_{i,j} = b_j;$$

$$0 \le x_{i,j} \le d_{i,j}; \qquad i = \overline{1,m};\ j = \overline{1,n}.$$

Величина $d_{i,j}$ называется ограничением коммуникации, благодаря этому задачу в данной постановке называют ещё Td-задачей, т.к. подразумевается, что в дополнение к известным векторам и матрицам A, B и C добавляется еще и матрица ограничений D.

Td-задачи решаются с помощью модифицированного алгоритма венгерского метода, модификации касаются учёта ограничений в ходе решения.

При изложении метода будем использовать следующие определения.

1. Элемент $c_{i,j} = 0$ матрицы C называется ***X-неполным нулём*** (или просто ***неполным нулём***), если в плане X решаемой Td-задачи величина $x_{i,j} < d_{i,j}$ (меньше пропускной способности коммуникаций).

2. ***Полным нулём*** (***X-полным нулём***) называется элемент $c_{i,j} = 0$ матрицы C, для которого $x_{i,j} = d_{i,j}$ (равен пропускной способности коммуникации).

3. Элемент $c_{i,j} = 0$ матрицы C является ***существенным нулём***, если $x_{i,j} > 0$ в плане перевозок X.

4. Если $x_{i,j}$ в плане перевозок X нулевой, то соответствующий нуль матрицы $c_{i,j}$ называется ***несущественным нулём***.

5. Элемент, находящийся в матрице C на пересечении выделенной строки и выделенного столбца, называется ***дважды выделенным***.

Функционирование алгоритма

Венгерский алгоритм решения Td-задачи имеет такую же структуру, показанную на рисунке 2.9, как и при отсутствии ограничений с точностью до названия этапов. Поэтому алгоритм дадим в отличиях и дополнениях.

Предварительный этап.

Заполнение плана X осуществляется не на основании формулы (2.37), а с учётом ограничений:

$$X_{i,j} = \min\{a_i, b_j, d_{i,j}\}. \tag{2.46}$$

ИТЕРАЦИЯ.

Разметка.

Дополнительно отмечают ***существенные нули*** матрицы C: точкой сверху – x-неполные нули $\dot{0}$, двумя точками – х-полные $\ddot{0}$.

1-й этап, поисковый.

Цель поиска: найти ***неполный нуль*** матрицы C, независимо от того существенный он или несущественный.

2-й этап, построение цепочки и коррекция плана X.

Корректирующий элемент выбирается по правилу

$$\theta = \min\left\{\delta_i^{НАЧАЛА}, \delta_j^{КОНЦА}, x^*_{i,j}, r'_{i,j}\right\}, \tag{2.47}$$

где $\delta_i^{НАЧАЛА}$ – невязка строки начала цепочки; $\delta_j^{КОНЦА}$ – невязка столбца конца цепочки; $x^*_{i,j}$ – элементы, стоящие на чётных местах цепочки, $r'_{i,j} = d_{i,j} - x_{i,j}$ – величина насыщения, необходимая для приведения коммуникации, стоящей на нечётной позиции в цепочке, к x-полному нулю.

3-й этап, коррекция матрицы C.

1. Корректирующий элемент h определяется как минимальный среди ***невыделенных положительных*** элементов, и, взятых ***по модулю***, ***дважды выделенных отрицательных***.

$$h = \min\{c_{i,j} > 0, |c^{+}_{i,j} < 0|\}. \tag{2.48}$$

2. Если все x-неполные нули выделены при нулевых невязках, все невыделенные элементы C отрицательны, а дважды выделенные – положительны, то Td-***задача неразрешима***.

3. Если в процессе коррекции ***дважды выделенный отрицательный*** элемент матрицы C ***становится нулём***, то его ***помечают звёздочкой***, а ***знак выделения*** над столбцом ***уничтожают***, делая столбец доступным для выполнения поисковых операций.

Пример решения задачи

Используем условие уже известного примера, дополненное ограничениями.

	B_1	B_2	B_3	B_4	
A_1	7	8	1	2	160
A_2	4	5	9	8	140
A_3	9	2	3	6	170
	120	50	190	110	

$$D = \begin{array}{|c|c|c|c|} \hline 50 & 100 & 200 & 50 \\ \hline 150 & 50 & 100 & 50 \\ \hline 50 & 50 & 50 & 50 \\ \hline \end{array}$$

Условие сбалансировано, модель замкнутая.

Предварительный этап.

На предварительном этапе матрица C_0 будет такая же, как и в случае ТЗ без ограничений.

$$\begin{array}{ccccc} & + & + & & \\ & 3 & 6 & \dot{0}^{\nabla} & 0 \\ C_0 = & \dot{0} & 3 & 8 & 6 \\ & 5 & \ddot{0} & 2 & 4 \end{array} \quad + \quad \begin{array}{cccccc} & & & & & \delta_i \\ & & & 160 & & 0 \\ X_0 = & 120 & & & & 20 \\ & & 50 & & & 120 \\ \delta_j & 0 & 0 & 30 & 110 & \end{array}$$

Начальный план X_0 получился тоже идентичный, поскольку он (2.46) соответствует. Заполнение X_0 читателю рекомендуется проделать самостоятельно. Суммарная невязка плана $\Delta = 280$.

1-я итерация.

Этап разметки.

Нули (1, 3), (2, 1) являются неполными, нуль (3, 2) – полным, выделяем 1-й и 2-й столбцы.

1-й этап.

Нуль (1, 3) выделяем штрихом, а первую строку – плюсом, так как её соответствует нулевая невязка в плане X_0. На этом поиск закончен, необходимы эквивалентные преобразования.

3-й этап.

Корректирующий элемент в невыделенной, серой части матрицы, согласно (2.46), равен 2. Применение алгоритма коррекции даёт C_1.

$$\begin{array}{ccccc} & + & + & & \\ & 5 & 8 & \dot{0}^{\nabla} & 0 \\ C_1 = & \dot{0} & 3 & 6 & 4 \\ & 5 & \ddot{0} & 0^{\nabla} & 2 \end{array} \quad + \quad \begin{array}{cccccc} & & & & & \delta_i \\ & & & 160 & & 0 \\ X_1 = & 120 & & & & 20 \\ & & 50 & 30 & & 90 \\ \delta_j & 0 & 0 & 0 & 110 & \end{array}$$

1-й этап.

Выделяем элемент (3, 3) штрихом, на этом этап поиска заканчивается удачно.

2-й этап.

Цепочка состоит из одного элемента. Соответствующую позицию плана X заполняем элементом $\theta = \min\{120, 30, 50\} = 30$. $r'_{3,3} = d_{3,3} - x_{33} = d_{3,3} - 0 = 50$. Получаем план коммуникаций X_1.

Невязка полученного плана $\Delta = 220$, расчёты продолжаются.

2-я итерация.

Этап разметки.

$$\begin{array}{ccccc} & + & + & (+) & \\ & 5 & 8 & \dot{0}^{*} & \rightarrow 0^{\nabla} \\ C_1 = & \dot{0} & 3 & \uparrow 6 & 4 \\ & 5 & \ddot{0} & \uparrow \dot{0}^{\nabla} & 2 \end{array} \quad + \quad \begin{array}{cccccc} & & & & & \delta_i \\ & & & 140 & 20 & 0 \\ X_2 = & 120 & & & & 20 \\ & & 50 & 50 & & 70 \\ \delta_j & 0 & 0 & 0 & 90 & \end{array}$$

Нуль, добавленный на предыдущей итерации, неполный, столбцы с первого по третий – обладают нулевыми невязками в плане X_1, выделяем их соответствующим образом.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом, поскольку её невязка нулевая. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. Нуль (3, 3) выделяем штрихом, этот нуль – искомый, будем строить цепочку.

2-й этап.

Цепочка соединяет нули с координатами в матрице C_1 $(3, 3)^{\nabla} \Rightarrow (1, 3)^{*} \Rightarrow (1, 4)^{\nabla}$.

Выбираем корректирующий элемент по формуле (2.47)

$$\theta = \min\{\delta_3 = 90, \delta_4 = 110, x^*_{1,3} = 160, r'_{3,3} = 20, r'_{3,3} = 50\} = 20$$

и перестраиваем план. При этом невязка получится Δ = 180. Итерации продолжаются.

3-я итерация.

Этап разметки.

$C_1 =$

+	+	(+)		
5	*8*	$\dot{0}^*$	0^{∇}	+
$\dot{0}$	*3*	*6*	*4*	
5	$\ddot{0}$	$\ddot{0}$	*2*	

$X_2 =$

				δ_i
		140	20	*0*
120				*20*
	50	50		*70*
0	*0*	*0*	*90*	δ_j

Нули (1, 3), (1, 4) и (2, 1) являются неполными, нули (3, 2) и (3, 3) – полными, выделяем столбцы с 1-го по 3-й.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. Больше неполных нулей в невыделенной части матрицы нет.

3-й этап.

Корректирующий элемент матрицы h = 2. Преобразуем матрицу C_1 в C_2.

1-й этап.

Отмечаем нуль (3, 4) штрихом. Его невязка положительна, он несущественный и неполный. Поиск завершён.

3-й этап.

Цепочка состоит из одного нуля с координатами (3, 4), эта позиция в плане заполняется исходя из невязок строки, столбца и ограничения коммуникации

$$\theta = \min\{\delta_3 = 70, \delta_4 = 90, r'_{3,4} = 50\} = 50.$$

Получим план X_3. Его невязка Δ = *80*. Следовательно, оптимум не получен.

$C_2 =$

+	+			
7	10	$\dot{0}^*$	0^{∇}	+
$\dot{0}$	3	4	2	
5	$\ddot{0}$	–2	0^{∇}	

$X_3 =$

				δ_i
		140	20	0
120				20
	50	50	50	20
0	0	0	40	δ_j

4-я итерация.

Этап разметки.

$C_2 =$

+	+	(+)		
7	10	$\dot{0}^*$	0^{∇}	+
$\dot{0}$	3	4	2	
5	$\ddot{0}$	–2	$\ddot{0}$	

$X_3 =$

				δ_i
		140	20	0
120				20
	50	50	50	20
0	0	0	40	δ_j

Нули (1, 3), (1, 4) и (2, 1) являются неполными, нули (3, 2) и (3, 4) – полными, выделяем столбцы с 1-го по 3-й.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. Больше неполных нулей в невыделенной части матрицы нет.

3-й этап.

Корректирующий элемент матрицы h = 2 выбираем из невыделенной серой части. Преобразуем матрицу C_2 в C_3.

$C_3 =$

+	+			
9	*12*	$\dot{0}^*$	0^{∇}	+
$\dot{0}$	*3*	*2*	0^{∇}	
5	$\ddot{0}$	*–4*	*–2*	

$X_4 =$

				δ_i
		140	20	*0*
120			20	*0*
	50	50	50	*20*
0	*0*	*0*	*20*	δ_j

1-й этап.

Отмечаем штрихом нуль (2, 4), он стоит в строке с положительной невязкой, и неполный несущественный. Поиск закончен успешно.

2-й этап.

Имеем цепочку и одного нуля, помещаем в позицию (2, 4) значение

$$\theta = \min\{\delta_2 = 20, \delta_4 = 40, r'_{2,4} = 50\} = 20 .$$

Получаем план X_4. Его суммарная невязка $\Delta = 40$. Итерации продолжаются.

5-я итерация.

Этап разметки.

$$C_3 = \begin{array}{cccc|c} (+) & + & (+) & & \\ \hline 9 & 12 & \dot{0}^{*} & \dot{0}^{\nabla} & + \\ \dot{0}^{*} & 3 & 2 & \dot{0}^{\nabla} & + \\ 5 & \ddot{0} & -4 & -2 & \end{array} \qquad X_4 = \begin{array}{cccc|c} & & & & \delta_i \\ \hline & & 140 & 20 & 0 \\ 120 & & & 20 & 0 \\ & 50 & 50 & 50 & 20 \\ \hline \delta_j: 0 & 0 & 0 & 20 & \end{array}$$

Нули (1, 3), (1, 4), (2, 1) и (2, 4) являются неполными, нуль (3, 2) – полным, выделяем столбцы с 1-го по 3-й.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. В третьем столбце нулей нет, а в четвёртом выделяем нуль (2, 4).

Так как невязка в торой сторки – нулевая, выделяем её плюсом. На её пересечении с 1-м столбцом стоит существенный нуль (2, 1), обозначаем позицию звёздочкой и снимаем выделение с первого столбца. Больше в невыделенной части матрицы (серая) нулевых элементов нет. Поиск закончен неудачно.

3-й этап.

Отрицательные элементы выделены однократно, единственный положительный элемент равен 5. Коррекция матрицы с его использованием даёт C_4.

$$C_4 = \begin{array}{cccc|c} & + & & & \\ \hline 9 & 17 & \dot{0}^{*} & \dot{0}^{\nabla} & + \\ \dot{0}^{*}\to & 8 & 2 & \to\dot{0}^{\nabla} & + \\ \uparrow 0^{\nabla} & \ddot{0} & -9 & -7 & \end{array} \qquad X_5 = \begin{array}{cccc|c} & & & & \delta_i \\ \hline & & 140 & 20 & 0 \\ 100 & & & 40 & 0 \\ 20 & 50 & 50 & 50 & 20 \\ \hline \delta_j: 0 & 0 & 0 & 20 & \end{array}$$

1-й этап.

Отмечаем нуль (3, 1) штрихом. Невязка соответствующей строки – положительна, поисковый этап закончился положительно.

2-й этап.

Цепочка соединяет нули с координатами в матрице C_3 $(3, 1)^{\nabla} \Rightarrow (2, 1)^{*} \Rightarrow (2, 4)^{\nabla}$.

Элемент для её коррекции есть

$$\theta = \min\{\delta_3 = 20, \delta_4 = 20, x^{*}_{1,3} = 120, r'_{3,1} = 50, r'_{2,4} = 30\} = 20 .$$

Получаем план X_5, невязка которого $\Delta = 0$. Следовательно, полученный план – оптимальный.

Целевая функция, соответствующая этому плану

$$L = 140 \times 20 \times 2 + 100 \times 4 + 40 \times 8 + 20 \times 9 + 50 \times (2 + 3 + 6) = 1630.$$

По сравнению с целевой функцией плана перевозок, полученного без учёта ограничений, целевая функция имеет большее значение. Это объясняется тем, что “благодаря” ограничениям не удаётся переместить потребное число единиц товара машрутом с минимальной стоимостью.

2.3.4.8. Решение задачи о назначениях

Задача о *назначениях*, она же задача *распределения* или задача *выбора*, имеет следующую содержательную постановку.

Предположим, что имеется n различных работ: $B_1, B_2, \ldots, B_n$ и столько же исполнителей этих работ (механизмов, например): $A_1, A_2, \ldots, A_n$. Причём, каждый из исполнителей способен выполнять любую работу, но единовременно может быть задействован только на одной из них. Пусть работа механизма A_i при выполнении работы B_j характеризуется некоторой неотрицательной величиной $c_{i,j}$, $i = 1, n, j = 1,n$.

Требуется таким образом распределить работы среди исполнителей, чтобы был достигнут определённый эффект (оптимум), например, достичь максимальной производительности труда или минимального расхода ресурсов.

Формально задача ставится в виде: для заданной матрицы

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n} \\ \dots & \dots & \dots & \dots \\ c_{i,1} & c_{i,2} & \dots & c_{1,n} \\ \dots & \dots & \dots & \dots \\ c_{n,1} & c_{2,n} & \dots & c_{n,n} \end{bmatrix}$$

найти распределение (расстановку, выбор и т.д.) $X = [x_{i,j}]$, $i = 1, n, j = 1, n$, обеспечивающие оптимум целевой функции

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j} \to opt\,. \qquad (2.49)$$

Если матрица C представляет собой ***производительность*** механизмов при выполнении отдельных работ, то задача решается на ***максимум***, а если смысл матрицы C – ***стоимостные или временные издержки***, то задача решается на ***минимум***.

Формально можно полагать, что задачи максимизации и минимизации связаны между собой выражением.

$$c_{i,j}^{Издержки} = 1 \Big/ c_{i,j}^{Производительность}\,. \qquad (2.50)$$

Житейски (2.50) можно прокомментировать так, что кто “с огоньком” работает, тот более выгоден, или, если так можно выразиться, лучше окупается.

Попутно заметим, что если, вследствие применения (2.50) числа окажутся дробными, то матрицу C путем умножения на соответствующую степень десятки и отбрасыванием оставшейся дробной части, легко привести к целому виду для удобства расчётов.

Так как при этом (2.49), в силу линейности, возрастёт, то необходимо будет её после расчётов откорректировать делением.

Матрица решения X состоит, в основном, из нулей и содержит всего n единиц, размещающихся в тех позициях, в которые соответствуют назначению i-го исполнителя на j-ю работу. Поэтому, при решении задачи вручную, определение значения (2.49) состоит в суммировании тех элементов исходной матрицы C, которым соответствуют единицы в матрице X.

Методы решения задачи о назначениях

Задача о назначениях представляет собой частный случай транспортной задачи без ограничения на пропускную способность коммуникаций.

Частности состоят в следующем:

- число поставщиков равно числу потребителей, отчего матрица C – квадратная;
- так как имеется условие о выполнении одним исполнителем в текущий момент времени только одной работы, то вектора с объёмами производства и объёмами потребления следует сделать единичными.

Исходя из этих допущений, для решения задачи о назначениях пригоден любой метод решения ТЗ, учитывающий матрицу стоимостей.

Понятно, что появились методы, учитывающие особенности задачи о назначениях. Наиболее популярна из них – модификация венгерского метода.

Ниже нами будет рассмотрен, для разнообразия, алгоритм решения задачи ***максимизации***. В этом случае полагается, что элементы матрицы C есть производительность i-го исполнителя при постановке его на выполнение j-й работы. Метод использует понятие “независимого нуля”.

Независимым нулём называется нуль матрицы C, не содержащий в строке и в столбце, на пересечении которых он находится, других независимых нулей.

Алгоритм решения задачи о назначениях с максимизацией целевой функции

Алгоритм имеет классическую структуру, характерную для всех венгерских методов.

Состоит из предварительного этапа и трёхэтапной итерации, в начале которой выполняется разметка.

Предварительный этап.

Состоит в построении матрицы C_0 и расстановке независимых нулей.

- В каждом столбце матрицы C отыскивается максимальный элемент, из которого затем вычитаются все элементы этого столбца. Результат записывается на место вычитаемого. В результате, матрица C преобразуется в матрицу C\`.
- В каждой строке матрицы C\` отыскивается минимальный элемент, который затем вычитается из всех элементов этой строки. В результате, матрица $C$\` преобразуется в матрицу C_0.

- Произвольно отмечаем нуль в первом столбце матрицы C_0 звёздочкой, полагая его независимым. Просматриваем по порядку остальные столбцы матрицы, руководствуясь определением. Если в рассматриваемых столбцах есть нулевой элемент, стоящий на строке, где нет нуля со звёздочкой, то его полагаем независимым, отмечаем звёздочкой и переходим к рассмотрению следующего столбца.

Заметим, что если бы задача решалась на минимум, то, в этом случае, построение матрицы C_0 не отличалось бы от обычного венгерского метода.

Проверка условия окончания

Подсчитываем число нулей со звёздочками (независимых нулей). Если их число равно размерности матриц n, то достигнут оптимум.

Необходимо рассчитать целевую функцию (2.49). План X, содержащий оптимальную расстановку, полностью определяется текущей матрицей C: нули, отмеченные звёздочками, соответствуют, единичным позициям оптимального плана, все прочие позиции – нулевые.

ИТЕРАЦИОННАЯ ЧАСТЬ АЛГОРИТМА

По завершении итерации, число независимых нулей в текущей матрице C увеличивается на один.

Таким образом, точное число итераций есть

$$N_{\text{итераций}} = n - n^*, \qquad (2.51)$$

где n – размерность матрицы, n^* - число независимых нулей.

Возможное число итераций, даже в начале счёта не более $n - 2$. (Какую топологию при этом имеют элементы исходной матрицы C, выносится потоку на обсуждение).

Разметка.

Разметка текущей матрицы C выполняется в начале итерации и сохраняется до её конца с теми изменениями, которые вносятся в неё по мере выполнения алгоритма.

Необходимо выделить знаком «плюс» j-е столбцы матрицы C, в которых присутствуют независимые нули.

Этап 1 – этап поиска.

Область поиска: невыделенная часть матрицы C – невыделенные столбцы и строки.

Цель поиска: найти в невыделенной части матрицы C нуль, стоящий в строке, в которой нет независимого нуля.

Поисковый этап заканчивается, как и в любом алгоритме венгерского метода, одним из случаев:

- если все нули матрицы C находятся в выделенной части, то необходимо перейти к этапу 3 – эквивалентных преобразований матрицы C;

- поиск завершился успешно, найден ноль в строке, где нет других независимых нулей. В этом случае далее выполняется этап 2 – построение цепочки и коррекция плана назначений.

В ходе поиска невыделенная часть матрицы C просматривается по столбцам сверху вниз а столбцы – слева направо.

Пусть среди элементов найден ноль. Его отмечают апострофом (штрихом) и анализируют строку, на которой он находится.

Если нет других независимых нулей (со звёздами), то этот ноль со штрихом является искомым, а поиск заканчивается успешно.

Если в строке уже имеется независимый нуль, то текущая стока выделяется знаком «плюс», и просматривается по местам её пересечения с выделенными столбцами. Если в месте пересечения стоит нуль со звёздочкой (*), то знак выделения над столбцом уничтожают, обводя кружком или заключая в скобки. Столбец становится невыделенным и делается доступным для поиска. Поиск далее продолжают по этому столбцу со снятым выделением.

Этап 2 – этап построения цепочки и коррекции плана назначений.

1. Цепочка ***не замкнута***, составляется из нулей со штрихом (0’) и нулей со звёздочками (0^*), содержит нечётное число элементов, и, в принципе, может состоять и из одного нуля со штрихом.

2. Цепочка начинается от последнего найденного нуля со штрихом 0’ к нулю со звездой 0^* по столбцу ($0' \to 0^*$), далее, по направлению под 90° к предыдущему, по строке от нуля со звездой к нулю со штрихом ($0^* \to 0'$) и так далее. На нечётных местах цепочки будут стоять нули со штрихом, а на чётных – нули со звёздами. Цепочка начинается в строке, в которой нет независимых нулей, и заканчивается в столбце, который в ходе разметки избежал выделения.

При переписи матрицы C, апострофы (штрихи) возле нулей, составляющих цепочку, заменяются звёздами, а звёздочку – устраняют. В результате число независимых нулей увеличивается на единицу.

Этап 3 – этап эквивалентного преобразования матрицы C.

Этот этап ничем не отличается от изложенного ранее, приводится здесь для полноты восприятия.

1. Среди невыделенных элементов текущей матрицы C выбирается минимальный положительный элемент $h > 0$. Этот элемент называется ***корректирующим***.

2. Корректирующий элемент вычитается от невыделенных строк матрицы C.

3. Корректирующий элемент прибавляется к выделенным столбцам матрицы C.

Пример. Решить задачу о назначениях на максимум.

$C =$

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

Предварительный этап.

Максимальные элементы в столбцах C выделены серым. В результате преобразования исходной матрицы по столбцам имеем

$C' =$

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

Минимальные элементы в строках C' выделены серым. После вычитания получаем C_0 вида

$C_0 =$

(+)	(+)	(+)		+	+	
0^*	1	0^∇	2	4	0	+
7	0^*	2	0^∇	2	5	+
5	6	1	1	0^*	4	
5	2	4	5	0	4	
2	3	3	2	0	0^*	
2	0^∇	0^*	3	1	0	+

Расставляем звёзды при независимых нулях: (1, 1),. (2, 2), (6, 3), (3, 5) и (5, 6).

Проверка условия окончания.

Число независимых нулей равно пяти, следовательно, задача решится за одну итерацию.

Итерация.

Разметка.

Размечаем 1, 2, 3, 5, 6 столбцы плюсами.

1-й этап, поиск.

Отмечаем нуль (2, 4) штрихом. На этой строке – независимый ноль, выделяем её, а выделение со второго столбца снимаем.

Во втором столбце отмечаем нуль (6, 2) штрихом, и, так как в шестой строке стоит независимый нуль, выделяем и её. После чего, снимаем выделение с третьего столбца, и отмечаем нуль (1, 3).

Первая строка выделяется плюсом, а выделение с первого столбца снимается. Поиск завершился неудачно.

3-й этап.

В невыделенной части матрицы (серый фон) находим корректирующий элемент. Он равен единице (таких элементов в матрице целых два). Корректируем матрицу C_0, получаем C_1.

$C_1 =$

				(+)	+	
0^*	1	0^∇	2	5	1	+
7	$0^*\rightarrow$	2	$\rightarrow 0^\nabla$	3	6	+
4	5	$\downarrow 0^\nabla$	0	$\leftarrow 0^*$	4	+
4	1	1	4	$\uparrow 0^\nabla$	4	
1	2	2	1	0	0^*	
2	$\uparrow 0^\nabla$	$\leftarrow 0^*$	3	2	1	+

1-й этап, снова поиск.

Нуль (3, 3) отмечаем штрихом, третья строка отмечается плюсом, и снимается выделение с пятого столбца.

В четвёртой строке отмечаем нуль (4, 5), который оказывается искомым. Этап удачно завершён.

2-й этап, коррекция плана.

Строим цепочку (4, 5) $\rightarrow$ (3, 5) $\rightarrow$ (3, 3) $\rightarrow$ (6, 3) $\rightarrow$ (6, 2) $\rightarrow$ (2, 2) $\rightarrow$ (2, 4). Проводимые согласно индексации замены даю следующую топологию размещения независимых нулей.

$C_1 =$

0^*	1	0	2	5	1
7	0	2	0^*	3	6
4	5	0^*	0	0	4
4	1	1	4	0^*	4
1	2	2	1	0	0^*
2	0^*	0	3	2	1

Число независимых нулей равно шести, расчёты закончены. Имеем следующий план расстановки исполнителей по работам:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Целевая функция полученного плана $L = 9 + 8 + 7 + 6 + 8 + 6 = 44$.

Решение предписывает, что 1-й исполнитель задействован на выполнении 1-й работы, 2-й – на 4-й, 3-й – на 3-й, 4-й – на 5-й, 5-й – на 6-й, а 6-й – на 2-й.

2.3.4.9. Вопросы для самоконтроля

1. Каковы недостатки алгоритма метода северо-западного угла?
2. В чём привлекательность метода минимальной стоимости?
3. Почему метод штрафов лучше метода минимальной стоимости?
4. Какой опорный план называется вырожденным и почему?
5. Как связаны между собой ЗЛП и ТЗ?
6. Что такое потенциал, и в чём его практический смысл?
7. Какова геометрическая интерпретация опорного плана *T*-задачи?
8. В чём состоит понятие баланса в *T*-задачах?
9. Для чего нужны цепочки при коррекции планов?
10. В чём заключается физический смысл построения цепочки?
11. Почему цепочки метода потенциалов замкнутые?
12. Каковы приёмы, применяемые для поддержания невырожденности плана?
13. В чём заключаются трудности, связанные с реализацией метода потенциалов на ЭВМ?
14. Какая транспортная модель называется замкнутой?
15. Для чего проверяется условие баланса и выполняется балансировка?
16. Что такое невязка?
17. Почему для венгерского метода нет необходимости в использовании опорного плана?
18. Что такое существенные и несущественные нули?
19. Как прогнозируется число итераций, оставшихся до получения оптимального решения?

20. Что является целью этапа поиска венгерского метода?
21. Где выполняется поиск?
22. В чём состоит назначение и сущность этапа эквивалентных преобразований матрицы *C*?
23. В чем сходство и различие при построении начальных планов для решения задачи венгерским методом и методом потенциалов?
24. Каковы, на Ваш взгляд, преимущества и недостатки обоих методов решения ТЗ? Обоснуйте высказываемые суждения.
25. Почему венгерский метод удобен при реализации на ЭВМ?
26. Поясните физический смысл, заключенный в процедуре построения цепочки.
27. Почему цепочка, которая строится в алгоритме метода потенциалов, замкнута, а в венгерском методе нет?
28. Каковы признаки неразрешимости *Td*-задачи?
29. Обоснуйте, почему *Td*-задачу трудно или невозможно решить методом потенциалов?
30. С чем связано увеличение целевой функции *Td*-задачи по сравнению с обычной *T*-задачей?
31. В чём состоят особенности применения алгоритма венгерского метода при наличии ограничений?
32. Приведите постановки *T*-задач применительно к технике передачи информации и компьютерным сетям.
34. Почему *Td*-задача не всегда может быть решена?
35. Как задается условие дискретности в задаче о назначениях?
36. Как связаны между собой задача о назначениях и Т-задача?
37. Всегда ли разрешима задача данного типа? Ответ обосновать.
38. Какой ноль называется независимым?
39. Какой алгоритм из известных алгоритмов решения транспортных задач кажется Вам наиболее эффективным? Дать обоснование.
40. Что будет, если наложить условие на допуск исполнителей к определенным работам?
41. Что, на Ваш взгляд, поменяется в постановке и алгоритме задачи, если каждый исполнитель будет в состоянии выполнять более одной работы, а каждая работа – быть исполнимой более чем одним исполнителем?
42. Как оценить число итераций, оставшихся до получения оптимального решения?