# Задача о паросочетаниях. Венгерский алгоритм

Над текстом работала: Кашапова Ольга

## Введение

Что такое “Венгерский алгоритм”. Это алгоритм, который способен решать задачу о паросочетаниях. Он также называется и “Алгоритмом Куна”, отсюда и его новое название - Кун был венгром. Его суть в том, что нужно найти решение задачи о назначениях, то есть, например, назначить работников на должности при минимальной заработной плате им от работодателя.

## Демонстрация работы алгоритма

### Этап 1

Итак, у нас есть матрица смежности графа. Для наглядности разберём этот алгоритм на примере. Суть в том, чтобы взять такие стоимости рёбер в матрице смежностей, чтобы они давали минимальную (максимальную) сумму, при этом на одной строке или в одном столбце не может быть более одного выбранного ребра.

| 32 | 28 | 4 | 26 | 4 |
| --- | --- | --- | --- | --- |
| 17 | 19 | 4 | 17 | 4 |
| 4 | 4 | 5 | 4 | 4 |
| 17 | 14 | 4 | 14 | 4 |
| 21 | 16 | 4 | 13 | 4 |

Если произвольно выбрать минимальную стоимость рёбер, начав с конца, то получится следующее:

| 32 | 28 | 4 | 26 | 4 |
| --- | --- | --- | --- | --- |
| 17 | 19 | 4 | 17 | 4 |
| 4 | 4 | 5 | 4 | 4 |
| 17 | 14 | 4 | 14 | 4 |
| 21 | 16 | 4 | 13 | 4 |

В итоге мы получим: 17+28+4+4+4 = 57 Многовато, не находите ли?

Так давайте же разберёмся, как находить эти паросочетания!

### Этап 2

В части 1 этого алгоритма необходимо будет из каждой строки вычесть минимальное число, находящееся в ней, чтобы в них оказались “0”. При этом должно оказаться так, что в каждом столбце также будет хотя бы один “0”. Иначе - нужно также вычесть минимальное из столбцов, которые не содержат “0”.

На матрице это будет выглядеть так, причём во всех столбцах окажутся “0”:

| 28 | 24 | 0 | 22 | 0 |
| --- | --- | --- | --- | --- |
| 13 | 15 | 0 | 13 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 13 | 10 | 0 | 10 | 0 |
| 17 | 12 | 0 | 9 | 0 |

Вот как-то так!

Но давайте убедимся в том, что эта матрица смежности не даёт нам совершенного паросочетания (сочетания, которое соединяет всё и вся). Для этого существует два способа.

Первый - неформальный - если посмотреть на матрицу, то можно сказать, что нельзя выбрать нули таким образом, чтобы они “выбивали” одновременно все строки и столбцы, при этом не задевая других (короче говоря, как в судоку одно число на одну строку и на один столбец).

Второй - формальный - для того, чтобы узнать количество совершенных (читать - нужных нам) паросочетаний в графе по матрице смежности, нам нужно посчитать перманент матрицы, но не этой, а той, которая получается заменой “0” на “1”, а чисел, больших “0” на “0”.

| 0 | 0 | 1 | 0 | 1 |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |

Нам нужно посчитать перманент этой матрицы, и если он равен “0”, то и решений пока нет. Перманент - это как детерминант, только миноры не меняют знак, то есть все слагаемые будут с “+”. В данном примере, если раскладывать по 1 столбцу, а затем по второму (уже в матрице 4x4), то можно заметить, что в перманенте будут “0”, то есть здесь пока решений не будет (1\*(0\*(...) + 0\*(...) + 0\*(...) + 0\*(...))).

[https://ru.wikipedia.org/wiki/Перманент](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%BC%D0%B0%D0%BD%D0%B5%D0%BD%D1%82) - что такое перманент.

### Этап 3

Итак, что же нам требуется сделать здесь? Если вкратце, то здесь нужно вычеркнуть строки и столбцы, при этом должно быть минимальное число вычёркиваний. Я придумал, как это сделать проще. Смотрите, берём какой-нибудь “0” и смотрим, что в данной клетке лучше вычеркнуть, если строку, то вычёркиваем строку, если столбец, то вычёркиваем столбец, и повторяем с невычеркнутыми “0”. Лучше всего начинать со строк и столбцов, где больше нулей. Например, для левого среднего верхнего нуля в таблице выше проще вычеркнуть строку - так и делаем (далее - серым обозначены вычеркнутые столбцы).

| 28 | 24 | 0 | 22 | 0 |
| --- | --- | --- | --- | --- |
| 13 | 15 | 0 | 13 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 13 | 10 | 0 | 10 | 0 |
| 17 | 12 | 0 | 9 | 0 |

Теперь внимание, сейчас нам нужно, во-первых, найти на оставшихся, “белых” в данном случае клетках наименьшее число и вычесть из этих белых клеток, получив где-то ещё “0”. Во-вторых, это самое наименьшее число мы должны (!!!) прибавить на пересечения вычеркнутых строк и столбцов, то есть на пересечении “серых” строк и столбцов в данном случае.

Здесь это будет число “9”. Мы вычтем “9” из “белых” клеток, но прибавим к “1” и “0” на пересечении “серых”. В итоге мы получим следующую матрицу.

| 19 | 15 | 0 | 13 | 0 |
| --- | --- | --- | --- | --- |
| 4 | 6 | 0 | 4 | 0 |
| 0 | 0 | 10 | 0 | 9 |
| 4 | 1 | 0 | 1 | 0 |
| 8 | 3 | 0 | 0 | 0 |

Проверим.

| 0 | 0 | 1 | 0 | 1 |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |

Аналогично предыдущему случаю разложим матрицу по первой строке и получим “0” (1\*(0\*(...) + 0\*(...) + 0\*(...) + 0\*(...))).

### Повторение

Повторим алгоритм.

| 19 | 15 | 0 | 13 | 0 |
| --- | --- | --- | --- | --- |
| 4 | 6 | 0 | 4 | 0 |
| 0 | 0 | 10 | 0 | 9 |
| 4 | 1 | 0 | 1 | 0 |
| 8 | 3 | 0 | 0 | 0 |

Минимальное число в “белых” клетках - “1”. Вычтем 1 из “белых” клеток, добавим к пересечения “серых”.

| 18 | 14 | 0 | 12 | 0 |
| --- | --- | --- | --- | --- |
| 3 | 5 | 0 | 3 | 0 |
| 0 | 0 | 11 | 0 | 10 |
| 3 | 0 | 0 | 0 | 0 |
| 8 | 3 | 1 | 0 | 1 |

Проверим.

| 0 | 0 | 1 | 0 | 1 |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 |

Разложим по 1 строке. В итоге получим 1\*1\*1(1\*1 + 1\*1) = 1\*2 = 2, то есть получим 2 совершенных паросочетания. Покажем их.

### Ответ

Паросочетание 1.

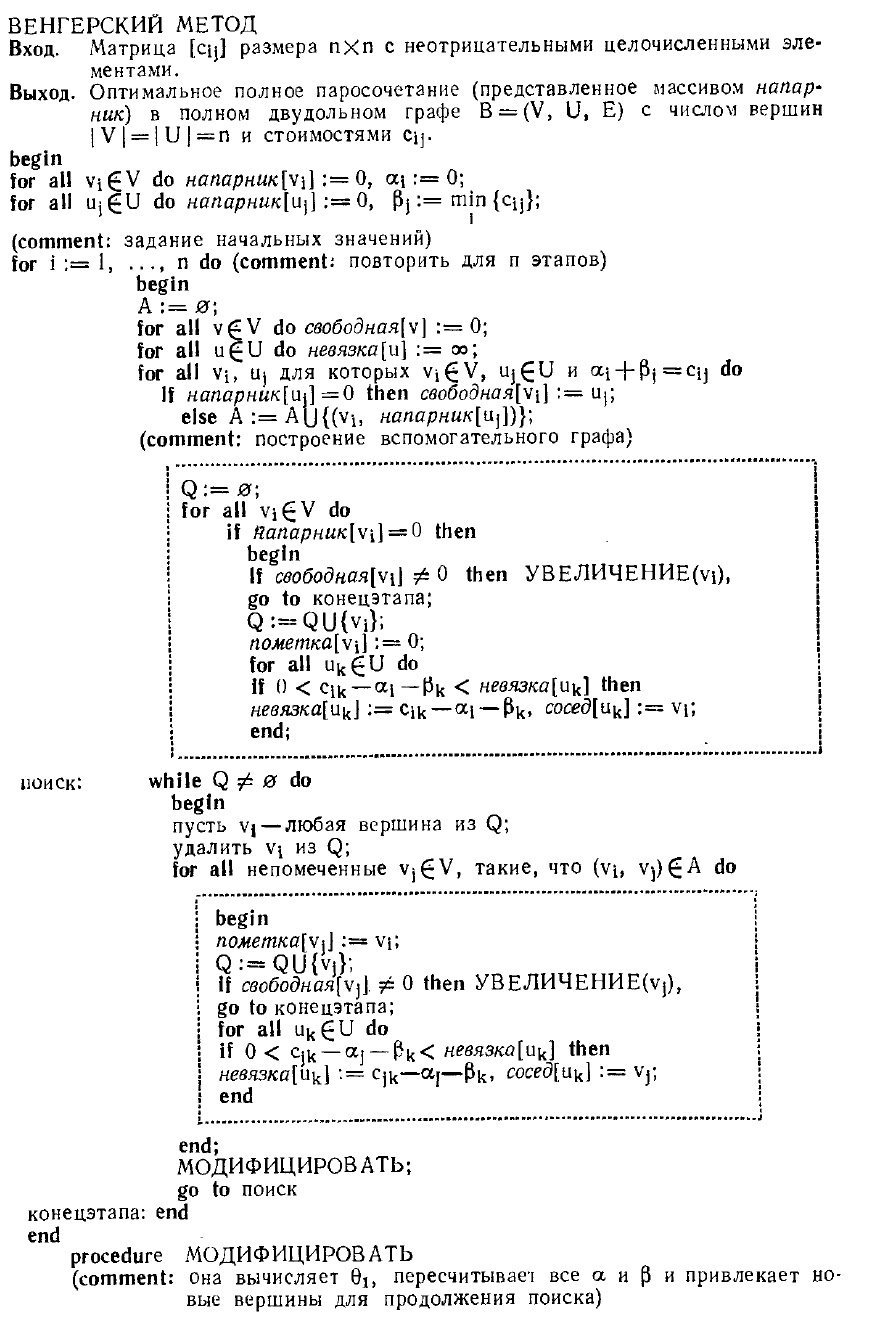
| 32 | 28 | 4 | 26 | 4 |
| --- | --- | --- | --- | --- |
| 17 | 19 | 4 | 17 | 4 |
| 4 | 4 | 5 | 4 | 4 |
| 17 | 14 | 4 | 14 | 4 |
| 21 | 16 | 4 | 13 | 4 |

Паросочетание 2.

| 32 | 28 | 4 | 26 | 4 |
| --- | --- | --- | --- | --- |
| 17 | 19 | 4 | 17 | 4 |
| 4 | 4 | 5 | 4 | 4 |
| 17 | 14 | 4 | 14 | 4 |
| 21 | 16 | 4 | 13 | 4 |

В итоге получаем 4 + 4 + 4 + 14 + 13 = 39. Это и будет нашим ответом.

Как выделять паросочетания в матрице - это отдельная тема, можно, например, искать единственные нули в строках или в столбцах, а затем уже выбирать из оставшихся.



# Ссылки на использованные источники

<https://www.youtube.com/watch?v=HXPALIZzxTI> - хорошее видео об основах венгерского алгоритма, дано объяснение, однако идею с графом (точнее, с построением, было бы неэффективно), интересный момент про перманент матрицы, позволяющий определить наличие решений.

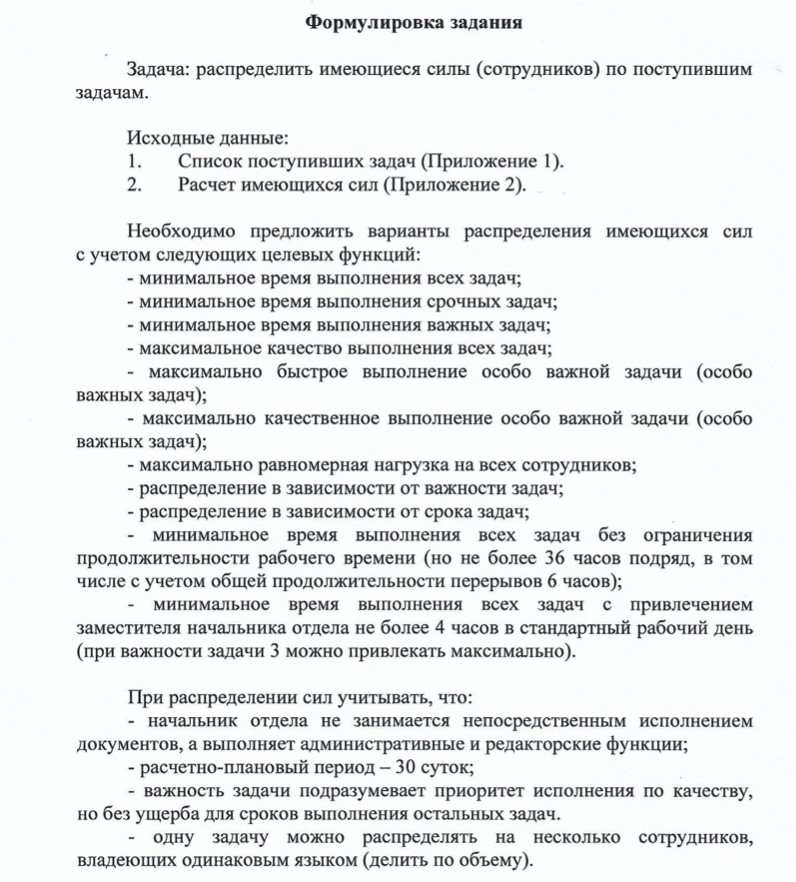
<https://habr.com/ru/post/422009/> - статья о венгерском алгоритме на пальцах, из неё стоит взять идею о том, как лучше заменить этап с построеним графа и обвода строк/столбцов вычёркиванием (что гораздо эффективнее и проще реализовать в программе).

[https://ru.wikipedia.org/wiki/Венгерский\_алгоритм](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BD%D0%B3%D0%B5%D1%80%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) - статья на Wikipedia.

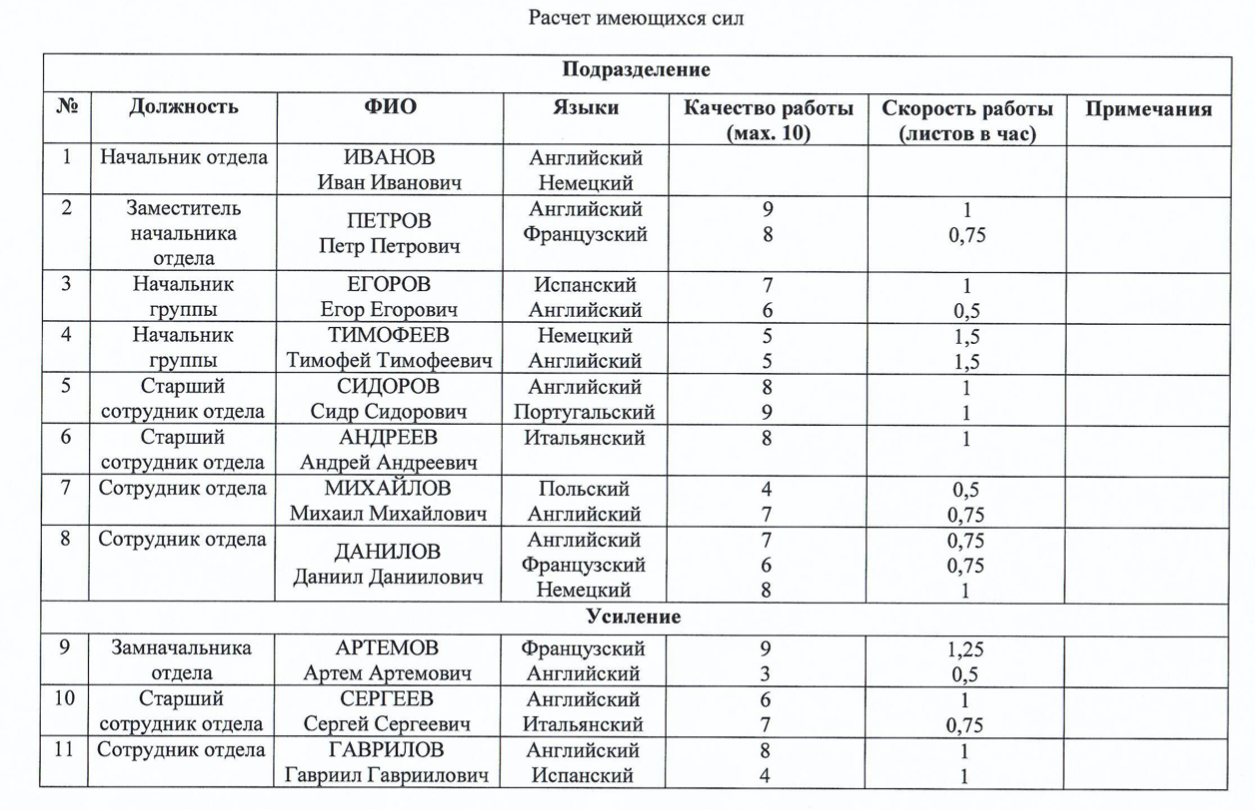
[https://ru.wikipedia.org/wiki/Задача\_о\_назначениях](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D0%BD%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%D1%85) - статья на Wikipedia.

# Данная задача о паросочетаниях

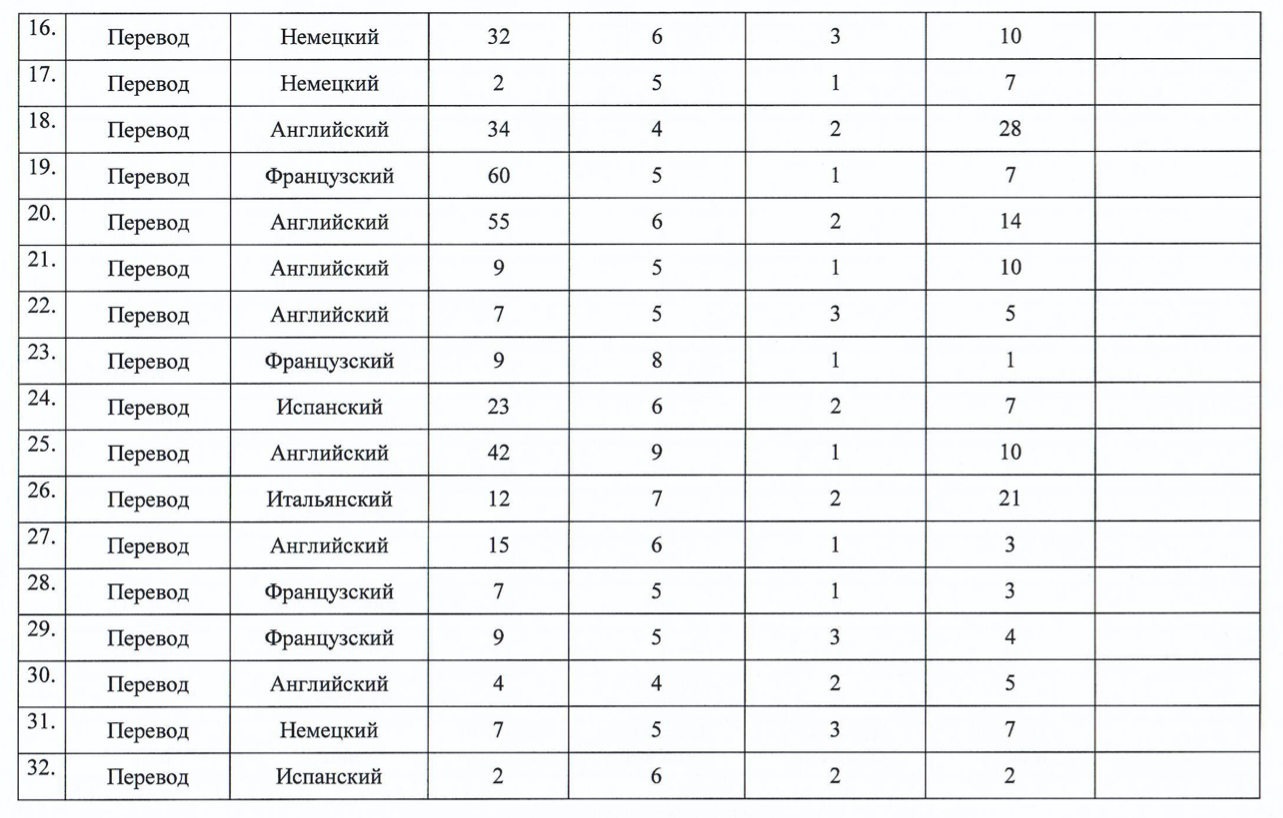
## Формулировка задания



## Исходные данные







## Решение

| должность | имя | языки | Качество | ск | задачи |
| --- | --- | --- | --- | --- | --- |
| зам | петров | английский | 9 | 1 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
| французский | 8 | 0,75 | 3, 6, 19, 23, 28, 29 |
| нач | егоров | испанский | 7 | 1 | 5, 24, 32 |
| английский | 6 | 0,5 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
| нач | тимофеев | немецкий | 5 | 1,5 | 4, 15, 16, 17, 31 |
| английский | 5 | 1,5 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
|  | сидоров | английский | 8 | 1 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
| португальский | 9 | 1 | 12, 13, 14 |
|  | андреев | итальянский | 8 | 1 | 7, 26 |
|  | михайлов | польский | 4 | 0,5 |  |
| английский | 7 | 0,75 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
|  | данилов | английский | 7 | 0,75 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
| французский | 6 | 0,75 | 3, 6, 19, 23, 28, 29 |
| немецкий | 8 | 1 | 4, 15, 16, 17, 31 |
| зам | артемов | французский | 9 | 1,25 | 3, 6, 19, 23, 28, 29 |
| английский | 3 | 0,5 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
|  | сергеев | английский | 6 | 1 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
| итальянский | 7 | 0,75 | 7, 26 |
|  | гаврилов | английский | 8 | 1 | 1, 2, 8, 9, 10, 11, 18, 20, 21, 22, 25, 27, 30 |
| испанский | 4 | 1 | 5, 24, 32 |

# Ссылки на использованные источники (транспортная задача)

<https://studme.org/80790/ekonomika/otkrytaya_model_transportnoy_zadachi> - открытая транспортная задача

<http://matecos.ru/mat/matematika/otkrytaya-transportnaya-zadacha-2.html> - тоже открытая транспортная задача

<https://www.resolventa.ru/data/metodstud/transproblem.pdf>

Мне кажется, что КОЛИЧЕСТВО СТРАНИЦ У ПЕРЕВОДЧИКОВ - это “имеющийся груз”, а КОЛИЧЕСТВО СТРАНИЦ У ЗАКАЗОВ - это “заказанный груз”, а то, что будет стоять непосредственно в матрице - это деление одного на другое, то есть дни, или стоимость “перевозки”. Либо можно подставить вместо этого КАЧЕСТВО.

[https://ru.wikipedia.org/wiki/Транспортная\_задача](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B0) - вот это, так как там предусматривается распределение на несуществующих исполнителей.

<https://habr.com/ru/post/63982/> - формулировка задачи о назначаениях