

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
Высшего образования  
**«Северо-Осетинский государственный университет  
имени Коста Левановича Хетагурова»**

Курсовая работа  
**Транспортная задача. Методы решения.**

**Выполнил:**  
Студент 3 курса направления:  
«Прикладная математика и информатика»  
*Гамосов Станислав Станиславович* \_\_\_\_\_

**Научный руководитель:**  
кандидат физико-математических наук  
*Тотиева Жанна Дмитриевна* \_\_\_\_\_

**«Работа допустима к защите»**  
Заведующий кафедрой алгебры и анализа,  
кандидат физико-математических наук  
*Джусоева Нонна Анатольевна* \_\_\_\_\_

Владикавказ 2021

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Введение</b>  | <b>2</b>  |
| <b>2</b> | <b>Постановка задачи</b>                                     | <b>3</b>  |
| <b>3</b> | <b>Методы построения опорного плана</b>                      | <b>8</b>  |
| 3.1      | Метод северо-западного угла . . . . .                        | 8         |
| 3.2      | Метод наименьшего элемента . . . . .                         | 11        |
| <b>4</b> | <b>Вырожденность опорного плана</b>                          | <b>13</b> |
| <b>5</b> | <b>Метод потенциалов</b>                                     | <b>15</b> |
| <b>6</b> | <b>Решение с помощью теории графов</b>                       | <b>18</b> |
| 6.1      | Транспортные сети и потоки . . . . .                         | 18        |
| 6.2      | Увеличивающие пути . . . . .                                 | 20        |
| 6.3      | Увеличение и сокращение потока . . . . .                     | 20        |
| 6.4      | Транспортная сеть с несколькими истоками и стоками . . . . . | 20        |
| 6.5      | Метод Форда-Фалкерсона . . . . .                             | 21        |
| 6.6      | Поток минимальной стоимости . . . . .                        | 22        |
| <b>7</b> | <b>Код решения</b>   | <b>23</b> |
| <b>8</b> | <b>Заключение</b>  | <b>25</b> |
|          | <b>Список литературы</b>                                     | <b>26</b> |

# 1 Введение

**Транспортная задача (задача Монжа — Канторовича)** – это спектр задач с единой математической моделью, классическая формулировка, которой звучит: *«Задача о наиболее экономном плане перевозок однородного продукта или взаимозаменяемых продуктов из пунктов производства в пункты потребления»*. Такая форма встречается чаще всего в линейном программировании, а если точнее в его практических приложениях.

**Линейное программирование** является одним из разделов математического программирования – области математики, разрабатывающей теорию и численные методы решения многомерных экстремальных задач с ограничениями.

Проблема была впервые формализована французским математиком *Гаспаром Монжем* в 1781 году. Прогресс в решении проблемы был достигнут во время Великой Отечественной войны советским математиком и экономистом *Леонидом Канторовичем*. Поэтому иногда эта проблема называется **транспортной задачей Монжа — Канторовича**.

Если вернуться к самой задаче огромное количество возможных вариантов перевозок затрудняет получение достаточно экономного плана эмпирическим или экспертным путем. Применение математических методов и вычислительных в планировании перевозок дает большой экономический эффект. Транспортные задачи могут быть решены **симплексным методом** однако матрица системы ограничений транспортной задачи настолько своеобразна, что для ее решения разработаны специальные методы. Они, как и **симплексный метод**, позволяют найти начальное опорное решение, а затем, улучшая его получить оптимальный результат. Транспортная задача может также решаться с ограничениями и без ограничений.

В зависимости от способа представления условий транспортной задачи она может быть представлена в **графовой** или **матричной** форме.

## 2 Постановка задачи

Задача эта возникает, когда речь идет о рациональной перевозке некоторого однородного продукта от производителей к потребителям. В этом случае для каждого потребителя безразлично, откуда, из каких пунктов производства будет поступать этот продукт, лишь бы он поступал в нужном объеме. Однако от того, насколько рациональным будет прикрепление пунктов потребления к пунктам производства, существенно зависит объем транспортной работы. В связи с этим естественно возникает вопрос о наиболее эффективном прикреплении, правильном направлении перевозок груза, при котором потребности удовлетворяются, а затраты на транспортировку минимальны. Более точно задача формулируется так.

Пусть имеются пункты производства  $(A_1, A_2, \dots, A_n)$  с объемами производства в единицу времени, равными соответственно  $(a_1, a_2, \dots, a_n)$ , и пункты потребления  $(B_1, B_2, \dots, B_m)$  с объемами потребления, равными  $(b_1, b_2, \dots, b_m)$  соответственно. Будем предполагать, что производство и потребление сбалансированы — сумма объемов производства равна сумме объемов потребления. Такой вид транспортной задачи называется **закрытым**.

$$\sum_{i=1}^n a_i = \sum_{j=1}^m b_j \quad (1)$$

В дальнейшем будем рассматривать только такой тип задачи. Однако любую **открытую** транспортную задачу  $(\sum_{i=1}^n a_i \neq \sum_{j=1}^m b_j)$  легко закрыть. Нужно ввести дополнительный пункт производства (пункт потребления) с недостающим объемом производства (объемом потребления) и с нулевыми стоимостями перевозок.

Предполагается, что известны величины  $c_{ij}$  — затраты по перевозке единицы продукта из  $i$ -го пункта производства в  $j$ -й пункт потребления. Они могут быть выражены в стоимостной (денежной) форме или в натуральной (километрах). Требуется найти такой план перевозок, при котором были бы удовлетворены потребности в пунктах  $(B_1, B_2, \dots, B_m)$  и при этом суммарные затраты на перевозку были бы минимальны. Обозначая через  $x_{ij}$  количество продукта, перевозимое из  $i$ -го пункта производства в  $j$ -го пункт потребления, приходим к следующей математической формулировке задачи:

**Найти минимум целевой функции:**

$$\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \longrightarrow \min \quad (2)$$

Так же для корректности задачи необходимо соблюдать три условия:

$$\text{Н1. } \sum_{i=1}^n x_{ij} = b_j; \quad (j = \overline{1, m})$$

$$\text{Н2. } \sum_{j=1}^m x_{ij} = a_i; \quad (i = \overline{1, n})$$

$$\text{Н3. } x_{ij} \geq 0; \quad (i = \overline{1, n}; \quad j = \overline{1, m})$$

Получается суммарные затраты на транспортировку в каждый пункт потребления завозится требуемое количество продукта, а так же из каждого пункта производства полностью вывозится произведенный продукт.

Всякий набор величин  $x_{ij} (i = \overline{1, n}; j = \overline{1, m})$ , удовлетворяющих условиям (Н1–Н3), мы будем называть допустимым планом перевозок. План, для которого суммарные затраты (2) достигают минимума, называется оптимальным.

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Условия транспортной задачи удобно представить в виде матрицы, которая имеет название **матрица перевозок**. В первой строке указаны величины потребностей, в первом столбце - значения запасов. В клетках внутренней матрицы ( $m \times n$  штук) записывают стоимости перевозок и сами перевозки. Нумеровать будем только строки и столбцы внутренней матрицы.

| Поставщики           | Потребители          |                      |     |                      |     |                      | Запасы поставщика                        |
|----------------------|----------------------|----------------------|-----|----------------------|-----|----------------------|--|
|                      | 1                    | 2                    | ... | $j$                  | ... | $m$                  |  |
| 1                    | $c_{11}$<br>$x_{11}$ | $c_{12}$<br>$x_{12}$ | ... | $c_{1j}$<br>$x_{1j}$ | ... | $c_{1m}$<br>$x_{1m}$ | $a_1$                                    |
| 2                    | $c_{21}$<br>$x_{21}$ | $c_{22}$<br>$x_{22}$ | ... | $c_{2j}$<br>$x_{2j}$ | ... | $c_{2m}$<br>$x_{2m}$ | $a_2$                                    |
| $\vdots$             | $\vdots$             | $\vdots$             | ... | $\vdots$             | ... | $\vdots$             | $\vdots$                                 |
| $i$                  | $c_{i1}$<br>$x_{i1}$ | $c_{i2}$<br>$x_{i2}$ | ... | $c_{ij}$<br>$x_{ij}$ | ... | $c_{im}$<br>$x_{im}$ | $a_i$                                    |
| $\vdots$             | $\vdots$             | $\vdots$             | ... | $\vdots$             | ... | $\vdots$             | $\vdots$                                 |
| $n$                  | $c_{n1}$<br>$x_{n1}$ | $c_{n2}$<br>$x_{n2}$ | ... | $c_{nj}$<br>$x_{nj}$ | ... | $c_{nm}$<br>$x_{nm}$ | $a_n$                                    |
| Спрос<br>потребителя | $b_1$                | $b_2$                | ... | $b_j$                | ... | $b_m$                | $\sum_{i=1}^n a_i$<br>$\sum_{j=1}^m b_j$ |

Рассмотрим теорему о разрешимости транспортной задачи:

**Теорема 1.** *Транспортная задача имеет решение, если суммарный запас груза в пунктах отправления равен суммарному спросу в пунктах назначения, т.е. если выполняется равенство (1).*

*Доказательство.* В случае превышения запаса над потребностью  $\sum_{i=1}^n a_i > \sum_{j=1}^m b_j$  как уже было обговорено выше, вводится фиктивный  $(m+1)$ -ый пункт назначения с потребностью  $b_{m+1} = \sum_{i=1}^n a_i - \sum_{j=1}^m b_j$

Соответствующие тарифы считаются равными нулю:

$$c_{im+1} = 0 \quad (i = \overline{1, m})$$

После этих преобразований получим закрытую модель транспортной задачи.

Аналогично, при  $\sum_{i=1}^n a_i < \sum_{j=1}^m b_j$  вводится фиктивный  $(n+1)$  пункт отправления с грузом,  $a_{n+1} = \sum_{j=1}^m b_j - \sum_{i=1}^n a_i$  а тарифы полагаются равными нулю:

$$c_{n+1j} = 0 \quad (j = \overline{1, n})$$

После этих преобразований получим закрытую модель транспортной задачи.

Теперь если в любом случае мы можем свести задачу к виду  $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j = A$  можем получить такие величины

$$x_{ij} = \frac{a_i b_j}{A}; \quad i = \overline{1, n}; \quad j = \overline{1, m}$$

Исходя из выше полученного имеем решение:

$$x_{ij} \geq 0$$

Так как  $\sum_{i=1}^n a_i = \sum_{j=1}^m b_j = A$  получаем:

$$\sum_{j=1}^m x_{ij} = \sum_{j=1}^m \frac{a_i b_j}{A} = \frac{a_i \sum_{j=1}^m b_j}{A} = a_i \quad (3)$$

$$\sum_{i=1}^n x_{ij} = \sum_{i=1}^n \frac{a_i b_j}{A} = \frac{b_j \sum_{i=1}^n a_i}{A} = b_j \quad (4)$$

Следовательно, система величин  $x_{ij}$ , удовлетворяя всем условиям транспортной задачи, является ее решением.  $\square$

Так же стоит доказать еще одну теорему связанную с тем что все базисы транспортной задачи треугольные. То есть система уравнений полученная содержит по крайней мере одно уравнение с единственным неизвестным, и даже если его исключить, то опять найдется по крайней мере одно уравнение с единственной неизвестной.

**Теорема 2.** *Все базисы в транспортной задаче задаются треугольной системой уравнений.*

*Доказательство.* Рассмотрим ячейки матрицы перевозок и пока покажем, что по крайней мере одна строка или по крайней мере один столбец содержит лишь одну базисную переменную и после удаления этой строки или этого столбца оставшийся массив сохранит это свойство.

Прежде всего каждая строка и каждый столбец содержит по крайней мере одну базисную переменную. В противном случае не выполняется условие отличия от 0 суммы по строке или по столбцу.

Для массива размерностью  $n \times m$ , если каждая строка содержит по крайней мере две базисные переменные, то количество базисных переменных не меньше  $2n$ ; если каждый столбец содержит по крайней мере две базисные переменные, то количество базисных переменных не меньше  $2m$ . Таким образом, общее количество базисных переменных не меньше  $m + n$ , но это невозможно, поскольку имеется всего  $n + m - 1$  базисных переменных. Поэтому хотя бы одна строка или хотя бы один столбец содержат лишь одну базисную переменную.

Если вычеркнуть эту строку или этот столбец, рассуждение можно повторить, а значит, описанное выше свойство выполняется для оставшегося массива, что и требовалось доказать.  $\square$

Так же стоит заметить, что если все  $a_i$  или  $b_j$  - целые, значения базисных переменных в допустимом базисном решении тоже целые. Поскольку это задача линейного программирования, оптимальное решение является базисным допустимым решением и, следовательно, целым. Это очень важно. Ведь тогда гарантируется, что в задачах не будет получено абсурдное решение.

С учетом выше представленного задача принимает вид:

$$\begin{cases} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \longrightarrow \min \\ \sum_{j=1}^m x_{ij} = a_i; \quad i = \overline{1, n} \\ \sum_{i=1}^n x_{ij} = b_j; \quad j = \overline{1, m} \\ x_{ij} \geq 0; \quad i = \overline{1, n}; \quad j = \overline{1, m} \end{cases} \quad (5)$$

**Пример:**

Составить математическую модель транспортной задачи перевоза груза из 3 складов в 5 магазинов. Матрица перевозок будет выглядеть так:

| Пункты      | B1 |   | B2 |   | B3 |   | B4 |   | B5 |   | Запасы |
|-------------|----|---|----|---|----|---|----|---|----|---|--------|
| A1          |    |   |    |   |    |   |    |   |    |   | 30     |
|             |    | 2 |    | 7 |    | 3 |    | 6 |    | 2 |        |
| A2          |    |   |    |   |    |   |    |   |    |   | 70     |
|             |    | 9 |    | 4 |    | 5 |    | 7 |    | 3 |        |
| A3          |    |   |    |   |    |   |    |   |    |   | 50     |
|             |    | 5 |    | 7 |    | 6 |    | 2 |    | 4 |        |
| Потребность | 10 |   | 40 |   | 20 |   | 60 |   | 20 |   |        |

$$\sum_{i=1}^3 a_i = 30 + 70 + 50 = 150 \quad \sum_{j=1}^5 b_j = 10 + 40 + 20 + 60 + 20 = 150$$

В качестве примера открытой модели давайте заменим потребность  $B4$ , которая равняется 60 на 40. В таком случае нужно было бы ввести еще одного потребителя с потребностью  $B6 = 20$  и с нулевыми стоимостями  $c_{16} = c_{26} = c_{36} = 0$ . Матрица перевозок тогда станет следующей:

| Пункты      | B1 |   | B2 |   | B3 |   | B4 |   | B5 |   | B6 |   | Запасы |
|-------------|----|---|----|---|----|---|----|---|----|---|----|---|--------|
| A1          |    |   |    |   |    |   |    |   |    |   |    |   | 30     |
|             |    | 2 |    | 7 |    | 3 |    | 6 |    | 2 |    | 0 |        |
| A2          |    |   |    |   |    |   |    |   |    |   |    |   | 70     |
|             |    | 9 |    | 4 |    | 5 |    | 7 |    | 3 |    | 0 |        |
| A3          |    |   |    |   |    |   |    |   |    |   |    |   | 50     |
|             |    | 5 |    | 7 |    | 6 |    | 2 |    | 4 |    | 0 |        |
| Потребность | 10 |   | 40 |   | 20 |   | 40 |   | 20 |   | 20 |   |        |



## 3 Методы построения опорного плана

### 3.1 Метод северо-западного угла

**Теорема 3.** *Существует план, содержащее не более чем  $(m + n - 1)$  положительных перевозок  $x_{ij}$ . При этом система векторов соответствующая таким перевозкам  $x_{ij}$ , линейно независима.*

*Доказательство.* Конструктивным доказательством теоремы может послужить процесс получения первого опорного плана, предложенный Данцигом и названный Чарнесом и Купером «**правилом северо-западного угла**». Применим это правило к следующей таблице:

|          |          |          |          |       |
|----------|----------|----------|----------|-------|
| $x_{11}$ | $x_{12}$ | $x_{13}$ | $x_{14}$ | $a_1$ |
| $x_{21}$ | $x_{22}$ | $x_{23}$ | $x_{24}$ | $a_2$ |
| $x_{31}$ | $x_{32}$ | $x_{33}$ | $x_{34}$ | $a_3$ |
| $b_1$    | $b_2$    | $b_3$    | $b_4$    |       |

Определим сначала значение переменной  $x_{11}$ , стоящей в верхнем левом углу. Пусть  $x_{11} = \min(a_1, b_1)$ ; если  $a_1 \leq b_1$ , то  $x_{11} = a_1$  и все  $x_{1j} = 0$  для  $j = 2, 3, 4$ . Если  $a_1 \geq b_1$ , то  $x_{11} = b_1$ , и все  $x_{i1} = 0$  для  $i = 2, 3, 4$ . Для определенности допустим, что справедливо первое предположение; тогда таблица преобразуется, как показано ниже в шаге 1. Здесь общее количество продукта, вывозимого из первого пункта отправления, уменьшается до нуля, а общее количество, которое необходимо подвезти к первому пункту назначения, равно  $b_1 - a_1$ .

*Шаг 1:* Пример  $b_1 > a_1$

|                |          |          |          |       |
|----------------|----------|----------|----------|-------|
| $x_{11} = a_1$ | 0        | 0        | 0        | 0     |
| $x_{21}$       | $x_{22}$ | $x_{23}$ | $x_{24}$ | $a_2$ |
| $x_{31}$       | $x_{32}$ | $x_{33}$ | $x_{34}$ | $a_3$ |
| $b_1 - a_1$    | $b_2$    | $b_3$    | $b_4$    |       |

После этого определяем значение первой переменной во второй строке. Пусть  $x_{21} = \min(a_2, b_1 - a_1)$ . Если допустить, что  $a_2 > b_1 - a_1$ ,  $x_{21} = b_1 - a_1$ , и  $x_{31} = 0$ . Это показано в шаге 2. Количество продукта, которое осталось перевезти из пункта отправления 2, теперь равно  $a_2 - b_1 + a_1$ . В свою очередь потребность первого пункта назначения полностью удовлетворена.

*Шаг 2:* Допустим, что  $a_2 > b_1 - a_1$ .

|                      |          |          |          |                   |
|----------------------|----------|----------|----------|-------------------|
| $x_{11} = a_1$       | 0        | 0        | 0        | 0                 |
| $x_{21} = b_1 - a_1$ | $x_{22}$ | $x_{23}$ | $x_{24}$ | $a_2 - b_1 + a_1$ |
| 0                    | $x_{32}$ | $x_{33}$ | $x_{34}$ | $a_3$             |
| 0                    | $b_2$    | $b_3$    | $b_4$    |                   |

Подобным же образом в зависимости от допущений, указанных далее, получаем следующие шаги. В каждом из них определяется значение переменной  $x_i$ ,

и сводится к нулю либо запас  $i$ -го пункта отправления, либо потребность  $j$ -го пункта назначения, или и то и другое вместе.

*Шаг 3:* Положим  $a_2 - b_1 + a_1 > b_2$ .

|                      |                |          |          |                         |
|----------------------|----------------|----------|----------|-------------------------|
| $x_{11} = a_1$       | 0              | 0        | 0        | 0                       |
| $x_{21} = b_1 - a_1$ | $x_{22} = b_2$ | $x_{23}$ | $x_{24}$ | $a_2 - b_1 + a_1 - b_2$ |
| 0                    | 0              | $x_{33}$ | $x_{34}$ | $a_3$                   |
| 0                    | 0              | $b_3$    | $b_4$    |                         |

*Шаг 4:* Пусть  $a_2 - b_1 + a_1 - b_2 < b_3$ .

|                      |                |                                  |          |       |
|----------------------|----------------|----------------------------------|----------|-------|
| $x_{11} = a_1$       | 0              | 0                                | 0        | 0     |
| $x_{21} = b_1 - a_1$ | $x_{22} = b_2$ | $x_{23} = a_2 - b_1 + a_1 - b_2$ | 0        | 0     |
| 0                    | 0              | $x_{33}$                         | $x_{34}$ | $a_3$ |
| 0                    | 0              | $b_3 - a_2 + b_1 - a_1 + b_2$    | $b_4$    |       |

Из шага 4 видно, что  $x_{33} = b_3 - a_2 + b_1 - a_1 + b_2$  и  $x_{34} = b_4$ . Следует отметить, что каждая из перевозок  $x_i$  была получена прибавлением и вычитанием различных комбинаций  $a_i$  и  $b_j$ . Поэтому если  $a_i$  и  $b_j$  были первоначально неотрицательными целыми числами, то и решение, получаемое в результате описанного выше процесса, будет состоять из неотрицательных целых чисел. Нетрудно видеть, что этот план может содержать самое большее  $n + m - 1$  положительных перевозок. При наших предположениях относительно величин  $a_i$  и  $b_j$ , и допущениях, сделанных при построении плана в рассмотренном примере с тремя пунктами отправления и четырьмя пунктами назначения, положительными перевозками являются:

$$\begin{aligned} x_{11} &= a_1; & x_{21} &= b_1 - a_1; \\ x_{22} &= b_2; & x_{23} &= a_2 - b_1 + a_1 - b_2; \\ x_{33} &= b_3 - a_2 + b_1 - a_1 + b_2; & x_{34} &= b_4; \end{aligned}$$

Используя приведенный алгоритм построения решения, можно доказать линейную независимость системы векторов, отвечающих выписанным положительным перевозкам. Тем самым будет установлено, что построенный план является опорным решением.  $\square$

### Пример:

Имеется транспортная таблица с исходными данными.

| Пункты      | В1 | В2 | В3 | Запасы |
|-------------|----|----|----|--------|
| А1          |    |    |    | 10     |
|             | 5  | 3  | 1  |        |
| А2          |    |    |    | 20     |
|             | 3  | 2  | 4  |        |
| А3          |    |    |    | 30     |
|             | 4  | 1  | 2  |        |
| Потребность | 15 | 20 | 25 |        |

Внесем в верхнюю левую клетку максимально возможного объема перевозки.

| Пункты      | В1        | В2 | В3 | Запасы |
|-------------|-----------|----|----|--------|
| A1          | <b>10</b> | 0  | 0  | 0      |
|             | 5         | 3  | 1  |        |
| A2          |           |    |    | 20     |
|             | 3         | 2  | 4  |        |
| A3          |           |    |    | 30     |
|             | 4         | 1  | 2  |        |
| Потребность | 5         | 20 | 25 |        |

Запасы на складе A1 закончились, поэтому в оставшиеся ячейки данной строки ставим прочерки. Затем переходим к следующей строке и заполняем ее ячейки слева направо.

| Пункты      | В1        | В2 | В3 | Запасы |
|-------------|-----------|----|----|--------|
| A1          | <b>10</b> | 0  | 0  | 0      |
|             | 5         | 3  | 1  |        |
| A2          | <b>5</b>  |    |    | 15     |
|             | 3         | 2  | 4  |        |
| A3          | 0         |    |    | 30     |
|             | 4         | 1  | 2  |        |
| Потребность | 0         | 20 | 25 |        |

| Пункты      | В1        | В2        | В3 | Запасы |
|-------------|-----------|-----------|----|--------|
| A1          | <b>10</b> | 0         | 0  | 0      |
|             | 5         | 3         | 1  |        |
| A2          | <b>5</b>  | <b>15</b> | 0  | 0      |
|             | 3         | 2         | 4  |        |
| A3          | 0         |           |    | 30     |
|             | 4         | 1         | 2  |        |
| Потребность | 0         | 5         | 25 |        |

Переходим к третьей строке и тоже заполняем ее слева направо.

| Пункты      | В1        | В2        | В3 | Запасы |
|-------------|-----------|-----------|----|--------|
| A1          | <b>10</b> | 0         | 0  | 0      |
|             | 5         | 3         | 1  |        |
| A2          | <b>5</b>  | <b>15</b> | 0  | 0      |
|             | 3         | 2         | 4  |        |
| A3          | 0         | <b>5</b>  |    | 25     |
|             | 4         | 1         | 2  |        |
| Потребность | 0         | 0         | 25 |        |

| Пункты      | В1        | В2        | В3        | Запасы |
|-------------|-----------|-----------|-----------|--------|
| A1          | <b>10</b> | 0         | 0         | 0      |
|             | 5         | 3         | 1         |        |
| A2          | <b>5</b>  | <b>15</b> | 0         | 0      |
|             | 3         | 2         | 4         |        |
| A3          | 0         | <b>5</b>  | <b>25</b> | 0      |
|             | 4         | 1         | 2         |        |
| Потребность | 0         | 0         | 0         |        |

Получено опорное решение:

$$X = \begin{pmatrix} 10 & 0 & 0 \\ 5 & 15 & 0 \\ 0 & 5 & 25 \end{pmatrix}$$

## 3.2 Метод наименьшего элемента

Ключевая идея заключается в следующем: в таблице из всех значений стоимостей выбираем наименьшее и в клетку  $(i, j)$  с наименьшей стоимостью записываем меньшее из чисел  $a_i, b_j$ . Исключаем из рассмотрения строку  $i$ , если запас  $a_i$ , вывезен полностью; или столбец  $j$ , если спрос  $b_j$ , удовлетворен полностью; или и строку и столбец, если  $a_i = b_j$ . Среди остальных стоимостей снова выбираем наименьшую и заполняем соответствующую клетку таблицы. Таким же образом продолжаем заполнять клетки таблицы, пока не будет найдено опорное решение.

| Пункты      | B1 | B2 | B3 | Запасы |
|-------------|----|----|----|--------|
| A1          |    |    |    | 10     |
|             | 5  | 3  | 1  |        |
| A2          |    |    |    | 20     |
|             | 3  | 2  | 4  |        |
| A3          |    |    |    | 30     |
|             | 4  | 1  | 2  |        |
| Потребность | 15 | 20 | 25 |        |

Объемы запасов и потребностей уменьшаются на величину груза. Если запасы склада исчерпаны, то полностью вычеркиваем эту строку таблицы. Если потребности завода полностью удовлетворены — полностью вычеркиваем этот столбец таблицы.

| Пункты      | B1 | B2 | B3 | Запасы |
|-------------|----|----|----|--------|
| A1          | 0  | 0  | 10 | 0      |
|             | 5  | 3  | 1  |        |
| A2          |    |    |    | 20     |
|             | 3  | 2  | 4  |        |
| A3          |    |    |    | 30     |
|             | 4  | 1  | 2  |        |
| Потребность | 15 | 20 | 25 |        |

Продолжаем в том же духе до тех пор, пока все запасы не будут исчерпаны, а все потребности удовлетворены.

| Пункты      | B1 | B2 | B3 | Запасы |
|-------------|----|----|----|--------|
| A1          | 0  | 0  | 10 | 0      |
|             | 5  | 3  | 1  |        |
| A2          |    | 0  |    | 20     |
|             | 3  | 2  | 4  |        |
| A3          |    | 20 |    | 10     |
|             | 4  | 1  | 2  |        |
| Потребность | 15 | 0  | 25 |        |

| Пункты      | B1 | B2 | B3 | Запасы |
|-------------|----|----|----|--------|
| A1          | 0  | 0  | 10 | 0      |
|             | 5  | 3  | 1  |        |
| A2          |    | 0  |    | 20     |
|             | 3  | 2  | 4  |        |
| A3          | 0  | 20 | 10 | 0      |
|             | 4  | 1  | 2  |        |
| Потребность | 15 | 0  | 5  |        |

| Пункты      | В1 | В2 | В3 | Запасы |
|-------------|----|----|----|--------|
| А1          | 0  | 0  | 10 | 0      |
|             | 5  | 3  | 1  |        |
| А2          | 15 | 0  |    | 5      |
|             | 3  | 2  | 4  |        |
| А3          | 0  | 20 | 10 | 0      |
|             | 4  | 1  | 2  |        |
| Потребность | 0  | 0  | 5  |        |

| Пункты      | В1 | В2 | В3 | Запасы |
|-------------|----|----|----|--------|
| А1          | 0  | 0  | 10 | 0      |
|             | 5  | 3  | 1  |        |
| А2          | 15 | 0  | 5  | 0      |
|             | 3  | 2  | 4  |        |
| А3          | 0  | 20 | 10 | 0      |
|             | 4  | 1  | 2  |        |
| Потребность | 0  | 0  | 0  |        |

В итоге мы получим опорный план перевозок для транспортной задачи.

$$X = \begin{pmatrix} 0 & 0 & 10 \\ 15 & 0 & 5 \\ 0 & 20 & 10 \end{pmatrix}$$

## 4 Вырожденность опорного плана

Вырожденность в транспортной задаче возникает, если одна или более базисных переменных обращаются в 0. Вырожденное решение может быть получено, если частичные суммы по столбцам равны частичным суммам по строкам. Проблему можно решить без особых трудностей. На каждом шаге следует различать базисные переменные, которые равны 0 и стоят в соответствующих ячейках, и небазисные переменные.

При построении опорного решения могут возникнуть трудности, если суммы и по строкам, и по столбцам равны между собой и обратились в 0. В этом случае из дальнейших рассмотрений следует исключить только одну из них. Другая сумма будет ликвидирована при присвоении базисной переменной значения 0. Поскольку на каждом шаге, кроме последнего, удаляется только одна строка или только один столбец, то в результате получается  $n + m - 1$  базисных переменных и столько заполненных клеток, сколько требуется (даже если некоторые базисные переменные обратились в 0).

Трудности могут возникнуть и при улучшении базисного допустимого решения. Применение правил может обратить в 0 более одной базисной переменной. В этом случае важно помнить, что только одна из них должна стать небазисной; остальные следует сохранить базисными, но с нулевыми значениями.

### Пример:

Построить первоначальный план для следующей ситуации:

| Пункты      | B1 | B2 | B3 | B4 | Запасы |
|-------------|----|----|----|----|--------|
| A1          |    |    |    |    | 60     |
|             |    |    |    |    |        |
| A2          |    |    |    |    | 30     |
|             |    |    |    |    |        |
| A3          |    |    |    |    | 40     |
|             |    |    |    |    |        |
| Потребность | 40 | 50 | 10 | 30 |        |

$60 + 30 + 40 = 40 + 50 + 10 + 30$  — спрос равен предложению — задача закрытая. Методом северо — западного угла получим опорный план. Начинаем с самой верхней левой ячейки.

| Пункты      | B1 | B2 | B3 | B4 | Запасы |
|-------------|----|----|----|----|--------|
| A1          | 40 |    |    |    | 60     |
|             |    |    |    |    |        |
| A2          | 0  |    |    |    | 30     |
|             |    |    |    |    |        |
| A3          | 0  |    |    |    | 40     |
|             |    |    |    |    |        |
| Потребность | 40 | 50 | 10 | 30 |        |

Потребности первого магазина выполнены полностью, но на складе еще остался груз. Заполняем дальше. Остатки груза с первого склада  $60 - 40 = 20$  перевозим в магазин второй. При этом, первый склад опустел, но потребности магазина не выполнены полностью.

| Пункты      | B1 | B2 | B3 | B4 | Запасы |
|-------------|----|----|----|----|--------|
| A1          | 40 | 20 | 0  | 0  | 60     |
|             |    |    |    |    |        |
| A2          | 0  |    |    |    | 30     |
|             |    |    |    |    |        |
| A3          | 0  |    |    |    | 40     |
|             |    |    |    |    |        |
| Потребность | 40 | 50 | 10 | 30 |        |

Переходим ко второму складу. Все 30 единиц груза переносим в магазин второй, потребности которого совпали с предложением склада  $50 - 20 = 30$ .

При данном распределении склад опустошается и потребности второго магазина выполняются полностью. Происходит потеря базисной клетки.

| Пункты      | B1 | B2 | B3 | B4 | Запасы |
|-------------|----|----|----|----|--------|
| A1          | 40 | 20 | 0  | 0  | 60     |
|             |    |    |    |    |        |
| A2          | 0  | 30 | 0  | 0  | 30     |
|             |    |    |    |    |        |
| A3          | 0  | 0  |    |    | 40     |
|             |    |    |    |    |        |
| Потребность | 40 | 50 | 10 | 30 |        |

В данном случае необходимо к базисным клеткам добавить клетку с нулевым значением, расположенную рядом с только что заполненной, которая обусловила потерю.

С третьего склада направим 10 единиц груза в магазин 4 для полного выполнения его потребностей. На 3-м складе остается  $40 - 10 = 30$  единиц груза, которые перенесем в последний магазин.

| Пункты      | B1 |  | B2 |  | B3 |  | B4 |  | Запасы |
|-------------|----|--|----|--|----|--|----|--|--------|
| A1          | 40 |  | 20 |  | 0  |  | 0  |  | 60     |
|             |    |  |    |  |    |  |    |  |        |
| A2          | 0  |  | 30 |  | 0  |  | 0  |  | 30     |
|             |    |  |    |  |    |  |    |  |        |
| A3          | 0  |  | 0  |  | 10 |  | 30 |  | 40     |
|             |    |  |    |  |    |  |    |  |        |
| Потребность | 40 |  | 50 |  | 10 |  | 30 |  |        |

Опорный план составлен. Количество базисных ячеек равно  $6 = 3 + 4 - 1$ . Условие невырожденности выполнено.

## 5 Метод потенциалов

Рассмотрим теорему об оптимальности решения транспортной задачи методом потенциалов.

$$u_i + v_j = c_{ij} : \quad \forall x_{ij} > 0 \quad (6)$$

$$u_i + v_j \leq c_{ij} : \quad \forall x_{ij} = 0 \quad (7)$$
$$i = \overline{1, n}; \quad j = \overline{1, m}$$

Немного перепишем исходную задачу и введем пару новых понятий

$u_i$  ( $i = \overline{1, n}$ ) - оценка единицы запаса (потенциал поставщика)  
 $v_j$  ( $j = \overline{1, m}$ ) - оценка единицы спроса (потенциал потребителя)

Тогда задача примет вид:

$$X = \sum_{i=1}^n a_i u_i + \sum_{j=1}^m b_j v_j \longrightarrow \max$$

при ограничениях

$$u_i + v_j \leq c_{ij} \quad (i = \overline{1, n}) \quad (j = \overline{1, m})$$

$u_i (i = \overline{1, n}); \quad v_j (j = \overline{1, m})$  - произвольного знака

### Алгоритм решения

1. Находится первый опорный план по одному из рассмотренных методов.
2. Проверяется найденный опорный план на оптимальность, для чего:
  - 2.1. Находятся потенциалы поставщиков  $u_i (i = \overline{1, n})$  и потребителей  $v_j (j = \overline{1, m})$  по формуле (6).

**Примечание:** Так как в опорном плане заполнено  $n + m - 1$  клеток таблицы транспортной задачи, то для нахождения потенциалов по данному плану можно составить систему из  $n + m - 1$  линейно независимых уравнений с  $n + m$  неизвестными. Такая система является неопределенной, и поэтому одной неизвестной (обычно  $u_1$ ) придают нулевое значение, а остальные находятся однозначно по формуле (6).
  - 2.2. Проверяется, выполнено ли условие (7) или, что то же самое, условие  $s_{ij} = c_{ij} - (u_i + v_j) \geq 0$ , где  $s_{ij}$ , - характеристика каждой свободной ячейки матрицы. Если для всех свободных ячеек матрицы условие (7) выполнено, т.е.  $s_{ij} \geq 0$ , то опорный план транспортной задачи является оптимальным (решение задачи завершено). Если же для некоторых свободных клеток таблицы  $s_{ij} < 0$ , то клетка с наименьшим значением  $s_{ij}$ , является перспективной, и выполняется следующий пункт алгоритма.



2.3. К перспективной ячейке строится цикл, расставляются знаки по циклу, при этом в перспективную ячейку ставится плюс, а остальные знаки в вершинах цикла чередуются, и определяется величина перераспределения груза по формуле  $Q = \min x_{ij}$ , где  $x_{ij}$  - объем перевозки груза, записанный в клетках (вершинах) цикла таблицы, отмеченных знаком минус.

2.4. Осуществляется перераспределение груза по циклу на величину  $Q$ . В результате выполнения этого пункта будет получен новый опорный план, который проверяется на оптимальность, т.е. производится переход к пункту 2.1 алгоритма.

### Пример:

Три завода производят однородную продукцию в количестве 650, 850 и 700 единиц соответственно. Эта продукция требуется четырем потребителям в количествах 500, 800, 300 и 600 единиц каждому. Затраты на перевозку единицы продукции (тыс. руб. ) от каждого завода к каждому потребителю заданы матрицей:

$$\begin{pmatrix} 30 & 50 & 62 & 10 \\ 40 & 50 & 80 & 20 \\ 50 & 10 & 30 & 30 \end{pmatrix}$$

Требуется спланировать перевозку груза так, чтобы суммарные транспортные затраты были минимальными.

Занесем данные транспортной задачи и найдем опорный план перевозок продукции методом минимальной стоимости. Найденный план является невырожденным, так как в таблице заполнено ровно  $m+n-1=3+4-1=6$  клеток. Затраты на перевозку продукции для данного плана равны:

$$X = 1\,500 + 6\,000 + 18\,000 + 5\,000 + 24\,000 + 7\,000 = 61\,500$$

Применяя далее алгоритм решения задачи, находим потенциалы поставщиков и потребителей. Полагая  $u_1 = 0$ , определяем, что  $v_1 = 30$ , так как  $u_1 + v_1 = 30$ , и из соотношения  $u_1 + v_4 = 10$  находим  $v_4 = 10$ . Аналогично найдены все другие потенциалы.

| Пункты      | B1  | B2    | B3    | B4  | Запасы       | $u_i$ |
|-------------|-----|-------|-------|-----|--------------|-------|
| A1          | 30  | 50    | 62    | 10  | 650          | 0     |
|             | 50  |       |       |     |              |       |
| A2          | 40  | 50    | 80    | 20  | 850          | 10    |
|             | 450 | ⊕ 100 | ⊖ 300 |     |              |       |
| A3          | 50  | 10    | 30    | 30  | 700          | -30   |
|             |     | ⊖ 700 | ⊕     |     |              |       |
| Потребность | 500 | 800   | 300   | 600 | 2200<br>2200 |       |
| $v_j$       | 30  | 40    | 70    | 10  |              |       |

Найдем характеристики свободных клеток таблицы транспортной задачи по формуле

$$s_{ij} = c_{ij} - (u_i + v_j)$$

$$s_{12} = 50 - (0 + 40) = 10, \quad s_{13} = 62 - (0 + 70) = -8,$$

$$s_{24} = 20 - (10 + 10) = 0, \quad s_{31} = 50 - (30 - 30) = 50,$$

$$s_{33} = 30 - (-30 + 70) = -10, \quad s_{34} = 30 - (-30 + 10) = 50.$$

Так как  $s_{13} < 0$  и  $s_{33} < 0$ , то план неоптимальный, и перспективной ячейкой в ней будет ячейка (3, 3) с наименьшей характеристикой  $s_{33} = -10$ .

Строим цикл к перспективной клетке и находим величину перераспределения груза  $Q = \min(300, 700) = 300$ .

Осуществляем перераспределение груза по циклу, добавляя величину  $Q = 300$  в клетках со знаком  $+$  и вычитая  $-$  в клетках со знаком  $-$ . В результате получаем новый опорный план.

Применяя правило пункта 2 алгоритма к полученному плану транспортной задачи, убеждаемся, что он оптимальный, так как для всех свободных клеток таблицы  $s_{ij} \geq 0$ . Суммарные минимальные затраты на перевозку продукции:

$$Z_{\min} = 61\,500 - 10 \cdot 300 = 58\,500$$

| Пункты      | B1  | B2  | B3  | B4  | Запасы       | $u_i$ |
|-------------|-----|-----|-----|-----|--------------|-------|
| A1          | 30  | 50  | 62  | 10  | 650          | 0     |
|             | 50  |     |     | 600 |              |       |
| A2          | 40  | 50  | 80  | 20  | 850          | 10    |
|             | 450 | 400 |     |     |              |       |
| A3          | 50  | 10  | 30  | 30  | 700          | -30   |
|             |     | 400 | 300 |     |              |       |
| Потребность | 500 | 800 | 300 | 600 | 2200<br>2200 |       |
| $v_j$       | 30  | 40  | 70  | 10  |              |       |

Эти минимальные затраты достигаются при следующих объемах:

$$x_{11} = 50, \quad x_{14} = 600, \quad x_{21} = 450,$$

$$x_{22} = 400, \quad x_{32} = 400, \quad x_{33} = 300.$$

Остальные неизвестные оставляем равны нулю.

## 6 Решение с помощью теории графов

В этом разделе рассмотрен алгоритм решения транспортной задачи при помощи теории графов и потоков. Методы представленные ниже подходят для решения гораздо большего класса задач, однако они эффективно справляются и с ранее поставленной проблемой. Прежде чем переходить на прямую к алгоритму, который решает транспортную задачу стоит рассмотреть пару дополнительных определений и теорем.

### 6.1 Транспортные сети и потоки

**Транспортная сеть**  $G = (V, E)$  представляет собой ориентированный граф, в котором каждое ребро  $(u, v) \in E$  имеет неотрицательную **пропускную способность**  $c(u, v) \geq 0$ . Далее мы потребуем, чтобы в случае, если содержит ребро  $(u, v)$ , обратного ребра  $(v, u)$  не было (вскоре мы увидим, как обойти это ограничение). Если  $(u, v) \notin E$ , то для удобства определим  $c(u, v) = 0$ , а также запретим петли. В транспортной сети выделяются две вершины: **исток**  $s$  и **сток**  $t$ . Для удобства предполагается, что каждая вершина лежит на некоем пути от истока к стоку, т.е. для любой вершины  $v \in V$  транспортная сеть содержит путь  $s \rightarrow v \rightarrow t$ . Таким образом, граф является связным и, поскольку каждая вершина, отличная от  $s$ , содержит как минимум одно входящее ребро,  $|E| > |V| - 1$ . Рис. 1 показан пример транспортной сети.

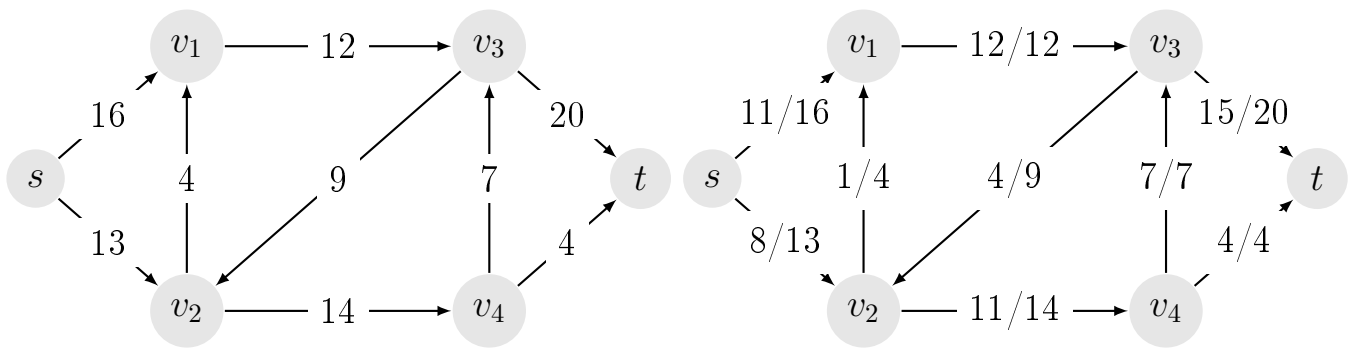


Рис. 1: Транспортная сеть и Поток в транспортной сети

Транспортная сеть  $G = (V, E)$  для задачи о грузовых перевозках компании. Истоком  $s$  является фабрика откуда груз поступает, а стоком  $t$  - склад, куда груз направляется. Продукты доставляются через промежуточные пункты, но за день из города  $u$  в город  $v$  можно отправить только  $(u, v)$  товара. На рисунке указана пропускная способность каждого ребра сети.

Поток  $f$  в транспортной сети  $G$  со значением  $|f| = 19$ . Каждое ребро  $(u, v)$  имеет метку  $f(u, v)/c(u, v)$  (косая черта используется только для того, чтобы отделить поток от пропускной способности, и не обозначает деление).

Теперь можно дать более формальное определение потоков. Пусть  $G = (V, E)$  - транспортная сеть с функцией пропускной способности  $c$ . Пусть  $s$  является исто-

ком, а  $t$  - стоком. **Потоком** в  $G$  является действительная функция  $f : V \times V \rightarrow R$ , удовлетворяющая следующим двум условиям:

**Ограничение пропускной способности.**

$$\forall u, v \in V: 0 \leq f(u, v) \leq c(u, v)$$

**Сохранение потока.**

$$\forall u \in V - \{s, t\}: \sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v)$$

Когда  $(u, v) \notin E$ , поток из  $u$  в  $v$  быть не может, так что  $f(u, v) = 0$ .

Неотрицательную величину  $f(u, v)$  мы называем потоком из вершины  $u$  в вершину  $v$ . **Величина**  $|f|$  потока  $f$  определяется как:

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) \quad (8)$$

т.е. как суммарный поток, выходящий из истока, минус входящий в него. (Здесь запись  $|\cdot|$  означает величину потока, а не абсолютное значение или мощность. Обычно транспортная сеть не имеет ребер, входящих в исток, и поток в исток, задаваемый суммой  $f(v, s)$ , равен 0. Однако мы включаем его, поскольку позже в этой главе при рассмотрении остаточных сетей потоки в исток станут важными. В задаче о **максимальном потоке** дана некоторая транспортная сеть  $G$  с истоком  $s$  и стоком  $t$ , и необходимо найти поток максимальной величины.

**Остаточной пропускной способностью** ребра  $(u, v)$  называется величина дополнительного потока, который мы можем направить из  $u$  в  $v$ , не превысив пропускную способность  $c(u, v)$ . Иными словами:

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{если } (u, v) \in E \\ f(v, u) & \text{если } (v, u) \in E \\ 0 & \text{в противном случае} \end{cases} \quad (9)$$

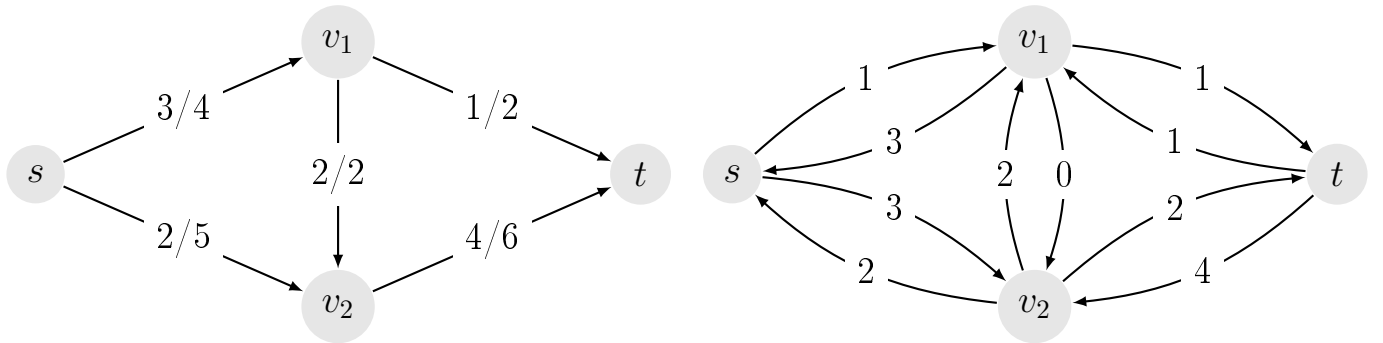


Рис. 2: Граф с некоторым потоком и Остаточная сеть этого графа

## 6.2 Увеличивающие пути

Для заданных транспортной сети  $G = (V, E)$  и потока  $f$  **увеличивающим путем** является простой путь из  $s$  в  $t$  в остаточной сети  $G_f$ . Согласно определению остаточной сети мы можем увеличить поток в ребре  $(u, v)$  увеличивающего пути до  $c_f(u, v)$  без нарушения ограничения пропускной способности соответствующего ребра в исходной сети.

Максимальная величина, на которую можно увеличить поток в каждом ребре увеличивающего пути  $p$ , называется остаточной пропускной способностью пути и задается формулой

$$C_f(p) = \min \{c_f(u, v) : (u, v) \text{ принадлежит } p\}$$

## 6.3 Увеличение и сокращение потока

Поток в остаточной сети предоставляет указания по добавлению потока к исходной транспортной сети. Если  $f$  представляет собой поток в  $G$ , а  $f'$  представляет собой поток в соответствующей остаточной сети  $G_f$  определим **увеличение**  $f \nearrow f'$  потока  $f$  на  $f'$  как функцию, отображающую  $V \times V$  на  $\mathbb{R}$  определенную следующим образом:

$$f \nearrow f'(u, v) = \begin{cases} f(u, v) + f'(u, v) - f'(v, u) & \text{если } (u, v) \in E \\ 0 & \text{в противном случае} \end{cases}$$

Интуитивно понятно, что это определение следует из определения остаточной сети. Мы увеличиваем поток в ребро  $(u, v)$  на  $f'(u, v)$ , но уменьшаем его на  $f'(v, u)$ , поскольку пропускание потока по обратным ребрам в остаточной сети означает уменьшение потока в исходной сети. Пропускание потока по обратному ребру в остаточной сети известно также как **сокращение**. Например, если мы пересылаем 5 ящиков с хоккейными шайбами из  $u$  в  $v$ , а 2 ящика - из  $v$  в  $u$ , то это эквивалентно (конечно, с точки зрения окончательного результата, а не оплаты перевозок) пересылке 3 ящиков из  $u$  в  $v$ , и ничего - из  $v$  в  $u$ . Сокращения такого вида являются ключевым моментом любого алгоритма максимального потока.

## 6.4 Транспортная сеть с несколькими истоками и стоками

В задаче о максимальном потоке может быть несколько истоков и стоков. Например, у компании может быть  $m$  фабрик  $\{s_1, s_2, \dots, s_m\}$  и  $n$  складов  $\{t_1, t_2, \dots, t_n\}$ , как показано ниже, где приведен пример транспортной сети с пятью истоками и тремя стоками. К счастью, эта задача не сложнее, чем обычная задача о максимальном потоке.

Преобразование задачи о максимальном потоке с несколькими истоками и стоками в задачу с одним истоком и одним стоком. Слева транспортная сеть с пятью

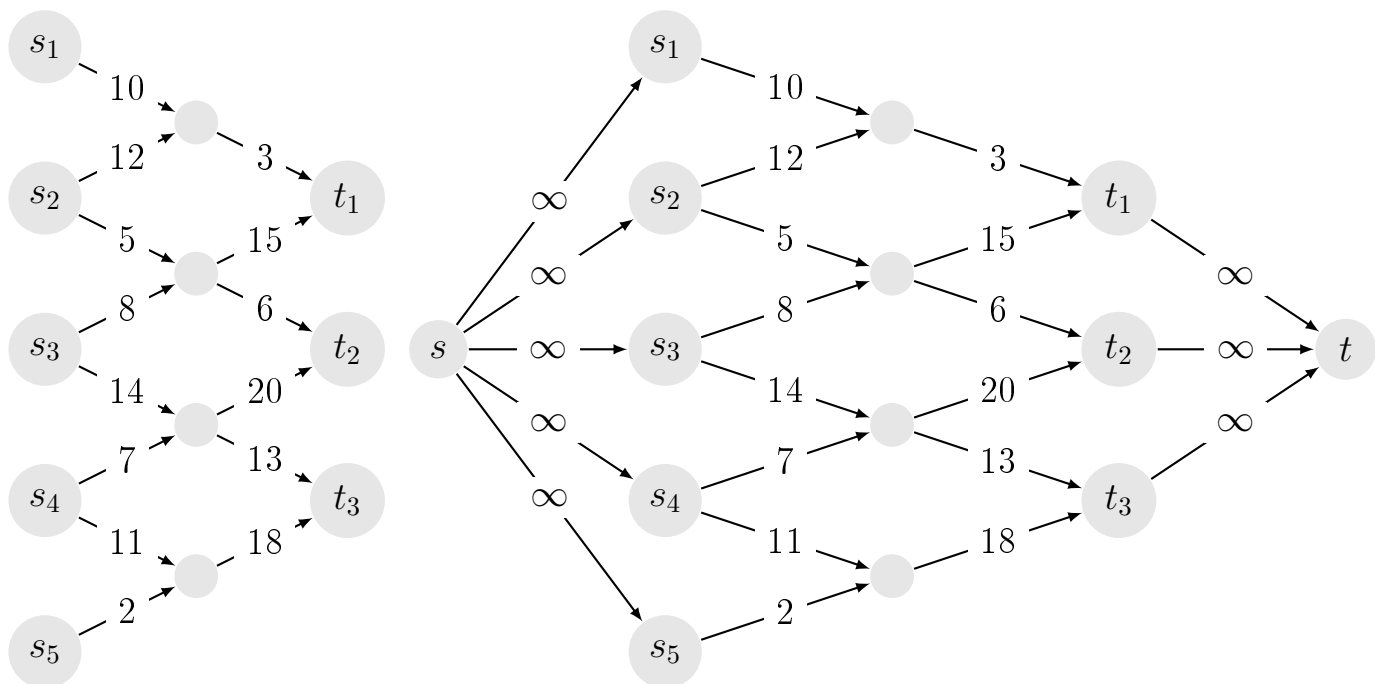


Рис. 3: Добавление фиктивных вершин

истоками  $S = \{s_1, s_2, s_3, s_4, s_5\}$  и тремя стоками  $T = \{t_1, t_2, t_3\}$ . Справа эквивалентная сеть с одним истоком и одним стоком. Мы добавляем фиктивный исток  $s$  и ребра с бесконечной пропускной способностью от до каждого из исходных истоков. Кроме того, мы добавляем фиктивный сток  $t$  и ребра с бесконечной пропускной способностью из каждого из исходных истоков в  $t$ .

Задача определения максимального потока в сети с несколькими истоками и несколькими стоками сводится к обычной задаче о максимальном потоке. Выше показано, как сеть с несколькими стоками и истоками, можно превратить в обычную транспортную сеть с одним истоком и одним стоком. Для этого в сеть добавляются **фиктивный исток**  $s$  и ориентированные ребра  $(s, s_i)$  с пропускной способностью  $(s, s_i) = \infty$  для каждого  $i = \overline{1, m}$ . Точно так же создается **фиктивный сток**  $t$  и добавляются ориентированные ребра  $(t_i, t)$  с  $c(t_i, t) = \infty$  для каждого  $i = \overline{1, n}$ .

Интуитивно понятно, что любой поток в сети слева соответствует потоку в сети справа, и наоборот. Единственный исток  $s$  просто обеспечивает поток любой требуемой величины к истокам  $s_i$ , а единственный сток  $t$  аналогичным образом потребляет поток любой желаемой величины от множественных стоков  $t_i$ .

## 6.5 Метод Форда-Фалкерсона

### Идея

Метод Форда Фалкерсона итеративно увеличивает значение потока. Вначале поток обнуляется:  $f(u, v) = 0$  для всех  $u, v \in V$ . Затем величина потока итеративно увеличивается посредством поиска увеличивающего пути (путь от источника  $s$  к стоку  $t$ , вдоль которого можно послать ненулевой поток). В стандартном алго-

ритме, осуществляющий этот поиск с помощью обхода в глубину (DFS). Процесс повторяется, пока можно найти увеличивающий путь.

## 6.6 Поток минимальной стоимости

Пусть дана сеть  $G(V, E)$ .  $s, t \in V$  — источник и сток. Ребра  $(u, v) \in E$  имеют пропускную способность  $c(u, v)$ , поток  $f(u, v)$  и цену за единицу потока  $a(u, v)$ . Тогда общая стоимость потока из  $s$  в  $t$ :

$$p(u, v) = \sum_{u, v \in V, f(u, v) > 0} a(u, v) \cdot f(u, v)$$

Дана сеть  $G(V, E)$ .  $s, t \in V$  — источник и сток. Ребра  $(u, v) \in E$  имеют пропускную способность  $c(u, v)$ , поток  $f(u, v)$  — и цену за единицу потока  $a(u, v)$ . Требуется найти максимальный поток, суммарная стоимость которого минимальна.

## 7 Код решения

---

```
1 struct rib {
2     int b, u, c, f;
3     size_t back;
4 };
5
6 void add_rib (vector < vector<rib> > & g, int a, int b, int u, int c) {
7     rib r1 = { b, u, c, 0, g[b].size() };
8     rib r2 = { a, 0, -c, 0, g[a].size() };
9     g[a].push_back (r1);
10    g[b].push_back (r2);
11 }
12
13 int main()
14 {
15     int n, m, k;
16     vector < vector<rib> > g (n);
17     int s, t;
18     ... чтение графа ...
19
20     int flow = 0, cost = 0;
21     while (flow < k) {
22         vector<int> id (n, 0); vector<int> d (n, INF);
23         vector<int> q (n); vector<int> p (n);
24         vector<size_t> p_rib (n);
25         int qh = 0, qt = 0;
26         q[qt++] = s;
27         d[s] = 0;
28         while (qh != qt) {
29             int v = q[qh++];
30             id[v] = 2;
31             if (qh == n) qh = 0;
32             for (size_t i=0; i<g[v].size(); ++i) {
33                 rib & r = g[v][i];
34                 if (r.f < r.u && d[v] + r.c < d[r.b]) {
35                     d[r.b] = d[v] + r.c;
36                     if (id[r.b] == 0) {
37                         q[qt++] = r.b;
38                         if (qt == n) qt = 0;
39                     }
40                     else if (id[r.b] == 2) {
41                         if (--qh == -1) qh = n-1;
42                         q[qh] = r.b;
43                     }
44                     id[r.b] = 1;
45                     p[r.b] = v;
46                     p_rib[r.b] = i;
47                 }
48             }
49         }
```



```

50     if (d[t] == INF) break;
51     int addflow = k - flow;
52     for (int v=t; v!=s; v=p[v]) {
53         int pv = p[v]; size_t pr = p_rib[v];
54         addflow = min (addflow, g[pv][pr].u - g[pv][pr].f);
55     }
56     for (int v=t; v!=s; v=p[v]) {
57         int pv = p[v]; size_t pr = p_rib[v], r = g[pv][pr].back;
58         g[pv][pr].f += addflow;
59         g[v][r].f -= addflow;
60         cost += g[pv][pr].c * addflow;
61     }
62     flow += addflow;
63 }
64
65 ... выводрезультата ...
66
67 }

```

---

## 8 Заключение

В работе изложены основные подходы и методы решения транспортной задачи, являющейся одной из наиболее распространенных задач линейного программирования.

Решение данной задачи позволяет разработать наиболее рациональные пути и способы транспортирования товаров, устранить чрезмерно дальние, встречные, повторные перевозки. Все это сокращает время продвижения товаров, уменьшает затраты предприятий и фирм, связанные с осуществлением процессов снабжения сырьем, материалами, топливом, оборудованием и т.д.

Алгоритм и методы решения транспортной задачи могут быть использованы при решении некоторых экономических задач, не имеющих ничего общего с транспортировкой груза. В этом случае величины тарифов  $c_{ij}$  имеют различный смысл в зависимости от конкретной экономической задачи. К таким задачам относятся следующие: оптимальное закрепление за станками операций по обработке деталей. В них  $c_{ij}$  является таким экономическим показателем, как производительность. Задача позволяет определить, сколько времени и на какой операции нужно использовать каждый из станков, чтобы обработать максимальное количество деталей. Так как транспортная задача требует нахождения минимума, то значения  $c_{ij}$  берутся с отрицательным знаком; оптимальные назначения, или проблема выбора. Уменьшение порожнего пробега сократит количество автомобилей для перевозок, увеличив их производительность; решение задач с помощью метода запрещения перевозок. Используется в том случае, если груз от некоторого поставщика по каким-то причинам не может быть отправлен одному из потребителей. Данное ограничение можно учесть, присвоив соответствующей клетке достаточно большое значение стоимости, тем самым в эту клетку не будут производиться перевозки.

Таким образом, важность решения данной задачи несомненна.

## Список литературы

- [1] Юдин Д.Б., Гольштейн Е.Г. Задачи и методы линейного программирования; Москва: Советское радио, 1969 - 736с.
- [2] Канторович Л.В., Горстко А.Б. Математическое оптимальное программирование в экономике; Москва: Знание, 1968 - 96с.
- [3] Палий И.А., Линейное программирование. Учебное пособие; Москва: Эксмо, 2008 - 256с.
- [4] Костевич Л.С. Математическое программирование. Информационные технологии оптимальных решений; Минск: Новое знание, 2003 - 424с.
- [5] Акулич И.Л. Математическое программирование в примерах и задачах; Санкт-Петербург: Лань, 2011 - 352с.
- [6] Данциг Д. Линейное программирование, его обобщения и применения; Москва: Прогресс, 1966 - 602с.
- [7] Гасс С. Линейное программирование; Москва: Физматгиз, 1961 - 303с.
- [8] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ, 3-е издание; Москва: «Вильямс», 2013. - 1328 с.