

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ФАКУЛЬТЕТ ПРИКЛАДНОЙ  
МАТЕМАТИКИ И ИНФОРМАТИКИ  
Кафедра компьютерных технологий и систем**

**Отчет  
о прохождении преддипломной практики**

Ковалевского Сергея Александровича  
студента 4 курса,

специальность «Информатика»

Руководитель практики:  
Пилипчук Людмила Андреевна  
канд. физ.-мат. наук, доцент

Минск, 2021

## Оглавление

Введение.....	3
Глава 1. Необходимые определения и теоремы.....	4
Глава 2. Опора мультиграфа для разреженной части системы.....	9
Глава 3. Теоретические свойства мультиграфов.....	10
Глава 4. Задача о расположении специальных программируемых устройств (сенсоров) в узлах мультиграфа.....	12
Глава 5. Алгоритмические и структурные свойства построения оптимального решения задачи оценки потоков на ненаблюдаемой части мультиграфа ....	14
Глава 6. Основные моменты реализации алгоритма.....	17
Заключение.....	20
Список литературы.....	21
Приложение А	
Листинг 1.....	22
Листинг 2.....	26
Листинг 3.....	32
Приложение В	
Листинг 1.....	34
Листинг 2.....	38

## Введение

Будем рассматривать разреженные линейные недоопределенные системы с вложенной сетевой структурой. Их структура наследуется из задач где применяются неоднородные сетевые потоки с переменными внешними потоками узлов. Одно из приложений рассматриваемых недоопределенных систем — задача о расположении сенсоров для мультиграфа. Задача заключается в расположении минимального количества сенсоров в узлах мультиграфа для определения величины потока на дугах и переменных интенсивностей узлов для всего мультиграфа. Исследование ранга разреженной матрицы основано на конструктивной теории декомпозиции разреженных СЛАУ.

Данная задача имеет широкое практическое применение, пример которого будет рассмотрен далее.

Для реализации алгоритмов, описанных в работе используется система компьютерной алгебры Wolfram Mathematica. Данная система компьютерной алгебры включает в себя возможность производить символьные вычисления, что позволяет построить общее решение для рассматриваемой разреженной СЛАУ и, используя возможности системы по упрощению логических выражений, проверить и продемонстрировать корректность построенного общего решения, а также наглядно продемонстрировать построенные вспомогательные структуры. Более того, данная система компьютерной алгебры предоставляет широкие возможности по оптимизации кода, такие как автоматическая параллелизация вычислений и компиляция часто используемых модулей, что позволяет значительно увеличить скорость вычислений.

## Глава 1. НЕОБХОДИМЫЕ ОПРЕДЕЛЕНИЯ И ТЕОРЕМЫ

Пусть  $G = (I, U)$  — конечный ориентированный связный мультиграф без циклов с множеством узлов  $I$  и множеством дуг  $U$ ,  $|U| \gg |I|$ . Пусть  $K$  ( $|K| < \infty$ ) — множество различных потоков, проходящих через сеть  $G$ . Полагаем  $K = \{1, \dots, |K|\}$ . Зафиксируем связную сеть, соответствующую конкретному потоку  $k \in K$ :  $G^k = (I^k, U^k)$ ,  $I^k \subseteq I$ ,  $U^k = \{(i, j)^k : (i, j) \in \hat{U}^k\}$ ,  $\hat{U}^k \in U$  — множество дуг  $G$ , по которым проходит поток  $k \in K$ . Кроме того, для каждой муьтидуги  $(i, j) \in U$  введем множество  $K(i, j) = \{k \in K : (i, j)^k \in U^k\}$ , которое содержит все потоки, проходящие по муьтидуге  $(i, j)$ .

Рассмотрим следующую разреженную недоопределенную СЛАУ:

$$\sum_{j \in I_i^+(U^k)} x_{ij}^k - \sum_{j \in I_i^-(U^k)} x_{ji}^k = \begin{cases} a_i^k, & i \in I^k \setminus I_k^*, \\ x_i^k \text{sign}(i), & i \in I_k^*. \end{cases}, k \in K, \quad (1)$$

$$\sum_{k \in K} \left( \sum_{(i,j)^k \in U^k} \lambda_{ij}^{k,p} x_{ij}^k \right) + \sum_{k \in K} \sum_{i \in I_k^*} \lambda_i^{k,p} x_i^k = \beta_p, p = \overline{1, q}. \quad (2)$$

Здесь  $I_i^+(U^k) = \{j \in I^k : (i, j)^k \in U^k\}$ ,  $I_i^-(U^k) = \{j \in I^k : (j, i)^k \in U^k\}$ ,  $x_{ij}^k$  — величина потока вдоль дуги  $(i, j)^k$ . Узлы  $i^k \in I_k^*$ ,  $k \in K$  (обозначаем  $i$  если  $k$  определено контекстом) называются динамическими (или базисными) — узлы с переменной интенсивностью  $x_i^k$ ,

$$\text{sign}(i^k) = \begin{cases} 1, & i^k \in I_{k+}^*, \\ -1, & i^k \in I_{k-}^*; \end{cases}$$

$$I_{k+}^*, I_{k-}^* \subseteq I_k^*, I_{k+}^* \cap I_{k-}^* = \emptyset, a_i^k, \lambda_{ij}^{k,p}, \lambda_i^{k,p}, \beta_p \in \mathbb{R}$$

Матрица системы (1) - (2) имеет блочную структуру вида:

$$A = \begin{bmatrix} M & B \\ Q & T \end{bmatrix}$$

Здесь  $M$  — матрица с блочно-диагональной структурой размера  $\sum_{k \in K} |I^k| \times \sum_{k \in K} |U^k|$ , т.к. каждый ее блок представляет из себя матрицу размера  $|I^k| \times |U^k|$  — матрицу инцидентности сети  $G^k = (I^k, U^k)$ ,  $k \in K$ , условно запишем  $M = M_1 \oplus \dots \oplus M_{|K|}$ , где  $M_k, k \in K$  — блоки матрицы  $M$ ;  $Q$  — матрица размера  $q \times \sum_{k \in K} |U^k|$  с элементами  $\lambda_{ij}^{k,p}, (i, j) \in U, k \in K(i, j), p = \overline{1, q}$ . Матрица  $B$  — разреженная матрица с блочно-диагональной структурой размера  $\sum_{k \in K} |I^k| \times \sum_{k \in K} |I_k^*|$ , т.к. каждый ее блок представляет из себя матрицу размера  $|I^k| \times |I_k^*|$  матрицы сети  $G^k, k \in K$ , условно запишем  $B = B_1 \oplus \dots \oplus B_{|K|}$ .

Матрица  $B_k$  колонке  $j$  имеет ненулевой элемент равный  $-sign(i)$  в строке  $i$  для всех  $j \in I_k^*$ , все остальные элементы равны нулю,  $i \in I^k, k \in K$ .  $T$  — матрица размера  $q \times \sum_{k \in K} |I_k^*|$  и состоит из элементов  $\lambda_i^{k,p}, i \in I_k^*, k \in K, p = \overline{1, q}$ . В [1], [3] рассматриваются СЛАУ с включенной сетевой структурой. Их структура унаследована из задач о гомогенном сетевом потоке с узлами переменной интенсивности. В [4] исследуются разреженные линейные системы с фрактальными матрицами. В этой работе рассматривается алгоритм декомпозиции разреженных систем для мультиграфа, которые используются в задаче расположения сенсоров.

Считаем, что  $\sum_{k \in K} |I^k| + q < \sum_{k \in K} |U^k| + \sum_{k \in K} |I_k^*|$ .

**Теорема 1.** Ранг матрицы  $[M_k \ B_k]$  системы (1) для фиксированного  $k \in K$  и графа  $G^k, I_k^* \neq \emptyset$  равен  $|I^k| + 1$ .

Характеристический вектор цикла, характеристический вектор цепи с учетом направления с привязкой к узлу, характеристический вектор цепи с учетом направления с привязкой к дуге для заданного  $k \in K$  составлены в соответствии с правилами [1].

**Теорема 2.** Характеристический вектор цикла, характеристический вектор цепи с учетом направления с привязкой к узлу, характеристический вектор цепи с учетом направления с привязкой к дуге для заданного  $k \in K$  удовлетворяют системе (3)[1,2].

$$\sum_{j \in I_i^+(U^k)} x_{ij}^k - \sum_{j \in I_i^-(U^k)} x_{ji}^k = \begin{cases} 0, & i \in I^k \setminus I_k^*, \\ x_i^k sign(i), & i \in I_k^*. \end{cases}, k \in K. \quad (3)$$

**Теорема 3.** Любое решение системы 3 для фиксированного  $k \in K$  есть линейная комбинация характеристических векторов [1].

**Определение 1.** Будем называть упорядоченную пару множеств  $R = \{U_R^k, I_R^{*k}, k \in K\}$ ,  $U_R^k \subseteq U^k$  и  $I_R^{*k} \subseteq I_k^*$  опорой для мультиграфа  $G$  и системы (1) если для него система

$$\sum_{j \in I_i^+(\tilde{U}^k)} x_{ij}^k - \sum_{j \in I_i^-(\tilde{U}^k)} x_{ji}^k = \begin{cases} 0, & i \in I^k \setminus \tilde{I}_k^*, \\ x_i^k sign(i), & i \in \tilde{I}_k^*. \end{cases}, k \in K. \quad (4)$$

при  $U^k = U_R^k, I_k^* = I_R^{*k}$  имеет единственное тривиальное решение, но при этом для всех следующих пар множеств система (4) имеет нетривиальные решения:

$$\tilde{R} = \{\tilde{U}^k, \tilde{I}_k^*, k \in K\},$$

$$\begin{aligned}
\tilde{U}^{k_0} &= U_R^{k_0} \cup (i_0, j_0)^{k_0}, (i_0, j_0)^{k_0} \in (U^{k_0} \setminus U_R^{k_0}); \\
\tilde{U}^k &= U_R^k, k \in (K \setminus k_0); \tilde{I}_k^* = I_R^{*k}, k \in K \\
\tilde{R} &= \{\tilde{U}^k, \tilde{I}_k^*, k \in K\}, \\
\tilde{U}^k &= U_R^k, k \in K; \tilde{I}_{k_0}^* = (I_R^{*k_0} \cup \{i_0\}), i_0 \in (I_{k_0}^* \setminus I_R^{*k_0}); \tilde{I}_k^* = I_R^{*k}, k \in (K \setminus k_0).
\end{aligned}$$

Для подмножества дуг  $U_1 \subseteq U$  введем понятие множества ницедентных узлов как

$$I(U_1) = \{i \in I : (i, j) \in U_1 \vee (j, i) \in U_1\}.$$

Для каждого  $k \in K$  построим лес из  $\tilde{t}_k$  деревьев  $T^{k, t_k} = \{I_T^{k, t_k}, U_T^{k, t_k}\}$ , где  $I_T^{k, t_k} = I(U_T^{k, t_k})$ ,  $U_T^{k, t_k} \subseteq U_R^k$ ,  $t_k = \overline{1, \tilde{t}_k}$ , таких, что каждое дерево  $T^{k, t_k}$  содержит ровно один узел  $u_{t_k} \in I_R^{*k}$ ,  $\forall t_k = \overline{1, \tilde{t}_k}$ ,  $k \in K$ .

Для каждого  $k \in K$  построим следующие множества:

$$U_R^k = \bigcup_{t_k=1}^{\tilde{t}_k} U_T^{k, t_k}, \quad I_R^{*k} = \left\{ u_{t_k} : t_k = \overline{1, \tilde{t}_k} \right\}$$

**Теорема 4.** (Критерий опоры сети)

Упорядоченная пара множеств  $R = \{U_R^k, I_R^{*k}, k \in K\}$ ,  $U_R^k \subseteq U^k$ ,  $I_R^{*k} \subseteq I_k^*$  является опорой мультиграфа  $G$  для системы (1) тогда и только тогда, когда выполняются следующие условия:

- 1) Каждая компонента связности  $T^{k, t_k} = \{I(U_T^{k, t_k}), U_T^{k, t_k}\}$ ,  $t_k = \overline{1, \tilde{t}_k}$  является деревом;
- 2) Множество узлов леса  $T^{k, t_k} = \{I(U_T^{k, t_k}), U_T^{k, t_k}\}$ ,  $t_k = \overline{1, \tilde{t}_k}$  покрывает все узлы графа  $G^k = (I^k, U^k)$ , то есть выполняется:

$$\bigcup_{t_k=1}^{\tilde{t}_k} I(U_T^{k, t_k}) = I^k;$$

- 3) Каждая компонента связности  $T^{k, t_k} = \{I(U_T^{k, t_k}), U_T^{k, t_k}\}$ ,  $t_k = \overline{1, \tilde{t}_k}$  содержит ровно один базисный узел из  $I_R^{*k}$ , то есть выполняется:

$$|I_R^{*k} \cap I(U_T^{k, t_k})| = 1, \quad \forall t_k = \overline{1, \tilde{t}_k}.$$

После того, как опора  $R = \{U_R^k, I_R^{*k}, k \in K\}$  мультиграфа  $G$  для системы (1) построена, можно рассмотреть несколько структур, которые могут быть получены путем добавления одной неопорной дуги к опоре  $R$ .

**Определение 2.** Характеристическим вектором, порожденным дугой  $(\tau, \rho)^k \in (U^k \setminus U_R^k)$  называется вектор  $\delta^k(\tau, \rho) = (\delta_{ij}^k(\tau, \rho), (i, j)^k \in U^k; \delta_i^k(\tau, \rho), i \in I_k^*)$ , построенный по следующим правилам для фиксированного  $k$ :

1) Если множество  $U_R^k \cup \{(\tau, \rho)^k\}$  имеет цикл  $L_k = \{I_L^k, U_L^k\}$ , то  $\delta^k(\tau, \rho)$  — характеристический вектор этого цикла и дуга  $(\tau, \rho)^k$  выбирается как определяющая направление обхода цикла.

2) Если в множестве  $U_R^k \cup \{(\tau, \rho)^k\}$  содержится цепь  $C_k = I_C^k, U_C^k$ , соединяющая узлы  $u$  и  $v \in I_R^{*k}$ , то  $\delta^k(\tau, \rho)$  — характеристический вектор цепи и дуга  $(\tau, \rho)^k$  выбирается как определяющая направление обхода цепи  $C$ .

**Определение 3.** Характеристическим вектором, порожденным узлом  $\gamma \in (I^k \setminus I_R^{*k})$  называется вектор  $\delta^k(\gamma) = (\delta_{ij}^k(\gamma), (i, j)^k \in U^k; \delta_i^k(\gamma), i \in I_k^*)$ , построенный как характеристический вектор цепи для фиксированного  $k$ , которая соединяет узлы  $\gamma$  и  $v \in I_R^{*k}$ , причем направление цепи выбирается таким образом, что  $\gamma$  — ее начало.

**Теорема 5.** Общее решение системы (1) для фиксированного  $k \in K$  может быть представлено в виде:

$$x_{ij}^k = \sum_{(\tau, \rho)^k \in (U^k \setminus U_R^k)} x_{\tau\rho}^k \delta_{ij}^k(\tau, \rho) + \sum_{\gamma \in I_k^* \setminus I_R^{*k}} x_{\gamma}^k \delta_{ij}^k(\gamma) + \tilde{x}_{ij}^k, \quad (i, j)^k \in U_R^k; \quad (5)$$

$$x_i^k = \sum_{(\tau, \rho)^k \in (U^k \setminus U_R^k)} x_{\tau\rho}^k \delta_i^k(\tau, \rho) + \sum_{\gamma \in I_k^* \setminus I_R^{*k}} x_{\gamma}^k \delta_i^k(\gamma) + \tilde{x}_i^k, \quad i \in I_R^{*k}; \quad (6)$$

где  $\tilde{x}^k = (\tilde{x}_{ij}^k, (i, j) \in U^k; \tilde{x}_i^k, i \in I_k^*)$  — частное решение неоднородной системы (1), то есть вектор  $(x_{ij}^{*k}, x_i^{*k})$  определенный как:

$$x_{ij}^{*k} = \sum_{(\tau, \rho)^k \in (U^k \setminus U_R^k)} x_{\tau\rho}^k \delta_{ij}^k(\tau, \rho) + \sum_{\gamma \in I_k^* \setminus I_R^{*k}} x_{\gamma}^k \delta_{ij}^k(\gamma), \quad (i, j)^k \in U_R^k;$$

$$x_i^{*k} = \sum_{(\tau, \rho)^k \in (U^k \setminus U_R^k)} x_{\tau\rho}^k \delta_i^k(\tau, \rho) + \sum_{\gamma \in I_k^* \setminus I_R^{*k}} x_{\gamma}^k \delta_i^k(\gamma), \quad i \in I_R^{*k};$$

является решением однородной системы (3).

Доказательство этой теоремы для фиксированного  $k$  можно найти в [1,3], общее неоднородное решение системы (1) находится как сумма общего решения однородной системы (3) и любого частного решения неоднородной системы (1).

Заметим, что теорема 5 и формулы (5)-(6) справедливы для любого частного решения  $\tilde{x}^k = (\tilde{x}_{ij}^k, (i, j) \in U^k; \tilde{x}_i^k, i \in I_k^*)$  системы (1), таким образом можно наложить ограничение на это

частное решение, например, положить:

$$\tilde{x}_{\tau\rho}^k = 0, (\tau, \rho) \in U^k \setminus U_R^k;$$

$$\tilde{x}_{\gamma}^k = 0, \gamma \in I_k^* \setminus I_R^{*k}$$

Далее будем использовать формулы (5)-(6), подразумевая, что частное решение  $\tilde{x}^k$  построено в соответствии со введенными ограничениями.



## Глава 2. ОПОРА МУЛЬТИГРАФА ДЛЯ РАЗРЕЖЕННОЙ ЧАСТИ СИСТЕМЫ

Пусть  $R = \{U_R^k, I_R^{*k}, k \in K\}$ ,  $U_R^k \subseteq U^k$ ,  $I_R^{*k} \subseteq I_k^*$  — опора мультиграфа  $G$  для системы (1). Выберем произвольно множества  $W = \{U_W^k, I_W^{*k}, k \in K\}$ ,  $|W| = q$ ,  $U_W^k \subseteq U^k \setminus U_R^k$ ,  $I_W^{*k} \subseteq I_k^* \setminus I_R^{*k}$ . Если подставить общее решение системы (1), которое имеет вид (5)-(6) в систему (2), система (2) примет следующий вид:

$$\sum_{k \in K} \left( \sum_{(\tau, \rho)^k \in U^k \setminus U_R^k} \Lambda_{\tau\rho}^{k,p} x_{\tau\rho}^k \right) + \sum_{k \in K} \left( \sum_{\gamma \in I_k^* \setminus I_R^{*k}} \Lambda_{\gamma}^{k,p} x_{\gamma}^k \right) = A_p, p = \overline{1, q}. \quad (7)$$

Где

$$\begin{aligned} \Lambda_{\tau\rho}^{k,p} &= \lambda_{\tau\rho}^{k,p} + \sum_{(i,j)^k \in U_R^k} \lambda_{ij}^{k,p} \delta_{ij}^k(\tau, \rho) + \sum_{i \in I_R^{*k}} \lambda_i^{k,p} \delta_i^k(\tau, \rho), \\ \Lambda_{\gamma}^{k,p} &= \lambda_{\gamma}^{k,p} + \sum_{(i,j)^k \in U_R^k} \lambda_{ij}^{k,p} \delta_{ij}^k(\gamma) + \sum_{i \in I_R^{*k}} \lambda_i^{k,p} \delta_i^k(\gamma), \\ A_p &= \beta_p - \sum_{k \in K} \sum_{(i,j)^k \in U_R^k} \lambda_{ij}^{k,p} \tilde{x}_{ij}^k - \sum_{k \in K} \sum_{i \in I_R^{*k}} \lambda_i^{k,p} \tilde{x}_i^k \end{aligned}$$

В системе (7) отделим переменные, относящиеся ко множеству  $W$ , после чего получим (8)

$$\begin{aligned} \sum_{k \in K} \sum_{(\tau, \rho)^k \in U_W^k} \Lambda_{\tau\rho}^{k,p} x_{\tau\rho}^k + \sum_{k \in K} \sum_{\gamma \in I_W^{*k}} \Lambda_{\gamma}^{k,p} x_{\gamma}^k &= A_p - \sum_{k \in K} \sum_{(\tau, \rho)^k \in U^k \setminus (U_W^k \cup U_R^k)} \Lambda_{\tau\rho}^{k,p} x_{\tau\rho}^k - \\ &- \sum_{k \in K} \sum_{\gamma \in I_k^* \setminus (I_W^{*k} \cup I_R^{*k})} \Lambda_{\gamma}^{k,p} x_{\gamma}^k, \quad p = \overline{1, q} \end{aligned} \quad (8)$$

### Глава 3. ТЕОРЕТИЧЕСКИЕ СВОЙСТВА МУЛЬТИГРАФОВ

Определение 4. Будем называть опорой мультиграфа  $G$  для систем (1)-(2) такую упорядоченную пару множеств  $Z = \{U_Z^k, I_Z^{*k}, k \in K\}$ ,  $U_Z^k \subseteq U^k$ ,  $I_Z^{*k} \subseteq I_k^*$ , что для  $\tilde{Z} = \{\tilde{U}^k, \tilde{I}_k^*, k \in K\}$ , при  $\tilde{U}^k = U_Z^k$ ,  $\tilde{I}_k^* = I_Z^{*k}$  система

$$\sum_{j \in I_i^+(\tilde{U}^k)} x_{ij}^k - \sum_{j \in I_i^-(\tilde{U}^k)} x_{ji}^k = \begin{cases} 0, & i \in I^k \setminus \tilde{I}_k^*, \\ x_i^k \text{sign}(i), & i \in \tilde{I}_k^*; \end{cases}, k \in K; \quad (9)$$

$$\sum_{k \in K} \left( \sum_{(i,j)^k \in \tilde{U}^k} \lambda_{ij}^{k,p} x_{ij}^k \right) + \sum_{k \in K} \sum_{i \in \tilde{I}_k^*} \lambda_i^{k,p} x_i^k = 0, p = \overline{1, q}$$

имеет только тривиальное решение, но система (9) имеет нетривиальное решение хотя бы для одной из следующих пар  $\tilde{Z}$ :

1.  $\tilde{Z} = \{\tilde{U}^k, \tilde{I}_k^*, k \in K\}$ , при  $\tilde{U}^{k_0} = U_Z^{k_0} \cup \{(i_0, j_0)^{k_0}\}$ ,  $(i_0, j_0)^{k_0} \in U^{k_0} \setminus U_Z^{k_0}$ ;  
 $\tilde{U}^k = U_Z^k$ , при  $k \in K \setminus \{k_0\}$ ;  
 $\tilde{I}_k^* = I_Z^{*k}$ ,  $k \in K$ ;
2.  $\tilde{Z} = \{\tilde{U}^k, \tilde{I}_k^*, k \in K\}$  при  $\tilde{U}^k = U_Z^k$ ,  $k \in K$ ;  
 $\tilde{I}_{k_0}^* = I_Z^{*k_0} \cup \{i_0\}$ ,  $i_0 \in I_{k_0}^* \setminus I_Z^{*k_0}$ ;  
 $\tilde{I}_k^* = I_Z^{*k}$ ,  $k \in K \setminus \{k_0\}$ .

Теорема 5. (Критерий опоры мультиграфа)

Пара множеств  $Z = \{U_Z^k, I_Z^{*k}, k \in K\}$ ,  $U_Z^k \subseteq U^k$ ,  $I_Z^{*k} \subseteq I_k^*$  является опорой мультиграфа  $G = \{I, U\}$  для систем (1)-(2) тогда и только тогда, когда выполняются следующие условия:

1. Пара множеств  $Z = \{U_Z^k, I_Z^{*k}, k \in K\}$  может быть разделена на 2 пары множеств:  $R = \{U_R^k, I_R^{*k}, k \in K\}$  и  $W = \{U_W^k, I_W^{*k}, k \in K\}$ , такие, что  $R \cup W = Z$  и  $R \cap W = \emptyset$  и при этом  $R$  — опора мультиграфа  $G$  для системы (1);
2.  $|W| = q$ , где  $q$  — количество линейно независимых уравнений в системе (2), или ранг системы (2), то есть  $q = \text{rank}(2)$ ;
3. матрица  $D$  системы (8), элементами которой являются детерминанты  $\Lambda_{\tau\rho}^{k,p}, \Lambda_{\gamma}^{k,p}$  структур, порожденных дугами и узлами пары  $W$ , является невырожденной.

Таким образом введено понятие опоры мультиграфа для систем (1)-(2). Из теоремы 5 следует, что для построения опоры мультиграфа  $G = \{I, U\}$   $Z = \{U_Z^k, I_Z^{*k}, k \in K\}$  необходимо сначала построить опору мультиграфа  $G$  для системы (1)  $R = \{U_R^k, I_R^{*k}, k \in K\}$ , которая образует лес деревьев, который покрывает все узлы из множества  $I^k$ ,  $\forall k \in K$  и каждая компонента связности этого леса деревьев включает ровно один узел из множества  $I_R^{*k}$ . Далее произвольно выбирается  $W = \{U_W^k, I_W^{*k}, k \in K\}$ ,  $|W| = q$  и  $N = \{U_N^k, I_N^{*k}, k \in K\}$ ,  $U_N^k = U^k \setminus (U_R^k \cup U_W^k)$ ,  $I_N^{*k} = I_k^* \setminus (I_R^{*k} \cup I_W^{*k})$  и вычисляются детерминанты элементов путем добавления элементов этих множеств к опоре  $R$  и поиска циклов или цепей.

## Глава 4. ЗАДАЧА О РАСПОЛОЖЕНИИ СПЕЦИАЛЬНЫХ ПРОГРАММИРУЕМЫХ УСТРОЙСТВ (СЕНСОРОВ) В УЗЛАХ МУЛЬТИГРАФА

Рассмотрим конечный связный ориентированный мультиграф  $G = \{I, U\}$ , и пусть для фиксированного  $k \in K$  орграф  $G^k = \{I^k, U^k\}$  симметричный, то есть если  $(i, j)^k \in U^k$ , то  $(j, i)^k \in U^k$ . Заметим, что  $G^k$  — орграф, величина потока на дугах  $(i, j)^k$  и  $(j, i)^k$  в общем случае будет различна. Чтобы подчеркнуть это различие, мы будем рассматривать  $G^k = I^k, U^k$  как двунаправленный орграф.

Величину потока будем рассматривать как функцию  $x : U \rightarrow R$ , которая удовлетворяет следующей системе:

$$\sum_{j \in I_i^+(U^k)} x_{ij}^k - \sum_{j \in I_i^-(U^k)} x_{ji}^k = \begin{cases} 0, & i \in I^k \setminus I_k^*, \\ x_i^k, & i \in I_k^*. \end{cases}, k \in K, \quad (10)$$

где  $I_k^* \subseteq I^k, k \in K$  — множество узлов с переменными интенсивностями,  $x_i^k$  — переменная интенсивность узла  $i \in I_k^*$ . Если переменная интенсивность  $x_i^k$  узла  $i$  положительная, то узел  $i$  — источник, если  $x_i^k$  — отрицательная, то узел  $i$  — сток. Для переменных интенсивностей должно выполняться условие баланса:

$$\sum_{i \in I_k^*} x_i^k = 0, \quad \forall k \in K.$$

Из теоремы 1 следует, что если  $I_k^* \neq \emptyset$ , то ранг системы (10) для связного орграфа  $G^k = (I^k, U^k)$  для фиксированного  $k \in K$  будет равен количеству узлов  $G^k$ , то есть  $\text{rank}(10) = |I^k|$ .

Пусть для того, чтобы получить информацию о величине потока на дугах  $(i, j)^k \in U^k$  и переменных интенсивностях узлов  $i \in I_k^*, k \in K$ , необходимо поместить "сенсор" на соответствующий узел. Узлы орграфа, на которых были размещены сенсоры, будем называть отслеживаемыми (monitored), а множество всех отслеживаемых узлов обозначим  $M$ , то есть  $M = \bigcup_{k \in K} M^k, M^k \subseteq I^k, k \in K$ . Предполагается, что если узел  $i^k$  — отслеживаемый, то известны величины потоков на всех входящих в него и исходящих из него дугах то есть

$$x_{ij}^k = f_{ij}^k, j \in I_i^+(U^k), \quad x_{ji}^k = f_{ji}^k, j \in I_i^-(U^k), \quad i \in M^k, \quad k \in K. \quad (11)$$

Если множество отслеживаемых узлов  $M$  содержит узлы  $i^k \in I_k^*$ , то нам также известны их переменные интенсивности, то есть

$$x_i^k = f_i^k, \quad i \in M^k \cap I_k^*, \quad k \in K. \quad (12)$$

Рассмотрим узел  $i \in I^k$  сети  $G^k = (I^k, U^k)$ . пусть для каждой исходящей дуги  $(i, j)^k \in U^k$  известна действительная величина  $p_{ij}^k \in (0, 1]$ , которая отражает, какая часть величины всего исходящего потока на узле  $i$   $\sum_{j \in I_i^+(U^k)} x_{ij}^k$  идет по дуге  $(i, j)^k$ . То есть

$$x_{ij}^k = p_{ij}^k \sum_{\tilde{j} \in I_i^+(U^k)} x_{i\tilde{j}}^k.$$

Таким образом, если  $|I_i^+(U^k)| > 1$  для узла  $i \in I^k$ , то величины потоков на всех исходящих дугах могут быть выражены через величину потока на одной из них, например выберем дугу  $(i, \tilde{j})^k \in U^k, \tilde{j} \in I_i^+(U^k)$ , тогда все  $x_{ij}^k, j \in I_i^+(U^k)$  можно записать в виде:

$$x_{ij}^k = \frac{p_{ij}^k}{p_{i\tilde{j}}^k} x_{i\tilde{j}}^k, j \in I_i^+(U^k). \quad (13)$$

Заметим, что формула (13) справедлива и при  $j = \tilde{j}$ , тогда  $x_{ij}^k = \frac{p_{ij}^k}{p_{i\tilde{j}}^k} x_{i\tilde{j}}^k = x_{ij}^k$ .

Таким образом можно сформулировать задачу о расположении сенсоров: какое минимальное число отслеживаемых узлов необходимо для того, чтобы система (10) имела единственное решение.

Иначе говоря необходимо построить множество

$$M = \bigcup_{k \in K} M^k,$$

такое, что  $|M| \rightarrow \min$  и система (10) с учетом (11)-(12), а также (13) имеет единственное решение.

Отметим, что данная задача имеет широкое применение в различных областях, позволяя минимизировать затраты на установку оборудования для оценки потока. Например, если представить дорожную сеть некоторого города в качестве сети  $G$ , где транспортные развязки образуют узлы, а дороги — дуги, въезды, выезды и парковки считать внешним потоком, а в качестве различных типов потока задать грузовой, пассажирский, легковой, пешеходный транспорт, то полученный граф будет разреженный, так как в большинстве случаев планировка дорог планарна, и степень большинства развязок не превосходит 4 (перекрестки). Таким образом решив задачу о расположении сенсоров для такого мультиграфа можно с минимальными затратами покрыть город сенсорами и получить возможность оценить интенсивность транспортного потока, что в свою очередь позволит произвести оптимизацию в сильно загруженных местах.

## Глава 5. АЛГОРИТМИЧЕСКИЕ И СТРУКТУРНЫЕ СВОЙСТВА ПОСТРОЕНИЯ ОПТИМАЛЬНОГО РЕШЕНИЯ ЗАДАЧИ ОЦЕНКИ ПОТОКОВ НА НЕНАБЛЮДАЕМОЙ ЧАСТИ МУЛЬТИГРАФА

Рассмотрим систему (10). Подставим в нее известные величины потока, то есть для узлов

$$i \in M, \quad M = \bigcup_{k \in K} M^k$$

$$x_{ij}^k = f_{ij}^k, j \in I_i^+(U^k), \quad x_{ji}^k = f_{ji}^k, j \in I_i^-(U^k), \quad i \in M^k, \quad k \in K$$

считаем, что величины  $f_{ij}^k$  известны.

Если  $|I_i^+(U^k)| > 1$  для узла  $i \in I^k$ , то величины потоков на всех исходящих дугах могут быть выражены через величину потока на одной из них, например выберем дугу  $(i, \tilde{j})^k \in U^k, \tilde{j} \in I_i^+(U^k)$ , тогда все  $x_{ij}^k, j \in I_i^+(U^k)$  можно записать в виде:

$$x_{ij}^k = \frac{p_{ij}^k}{p_{i\tilde{j}}^k} x_{i\tilde{j}}^k, j \in I_i^+(U^k). \quad (14)$$

Подставим (14) в систему (10),  $i \in I^k, k \in K$ .

Теперь удалим из графов  $G^k = (I^k, U^k), k \in K$  все дуги и узлы, для которых величина потока и переменные интенсивности уже определены. Таким образом получим новый мультиграф  $\bar{G} = (\bar{I}, \bar{U})$ , состоящий из графов  $\bar{G}^k = (\bar{I}^k, \bar{U}^k), k \in K$ , причем  $\bar{G}^k = (\bar{I}^k, \bar{U}^k)$  — в общем случае несвязный граф, относящийся к соответствующему типу потока  $k \in K$ . Для каждой мультидуги  $(i, j) \in \bar{U}$  введем множество  $\bar{K}(i, j) = \{k \in K : (i, j)^k \in \bar{U}^k\}$  содержащее все типы потоков, проходящие по этой мультидуге.

Новый мультиграф  $\bar{G}$  несвязный, то есть можно выделить в нем компоненты связности, некоторые из которых могут не содержать ни одного из узлов с переменной интенсивностью  $i \in \bar{I}_k^*$ , где  $\bar{I}_k^*$  — множество узлов с переменной интенсивностью полученного графа  $\bar{G}^k, k \in K$ .

Если  $|I_i^+(\bar{U}^k)| > 1$  для узла  $i \in \bar{I}^k$ , то величины потоков на всех исходящих дугах могут быть выражены через величину потока на одной из них, например выберем дугу  $(i, \tilde{j})^k \in U^k, \tilde{j} \in I_i^+(\bar{U}^k)$ , тогда все  $x_{ij}^k, j \in I_i^+(\bar{U}^k)$  можно записать

в виде:

$$x_{ij}^k = \frac{p_{ij}^k}{p_{i\tilde{j}}^k} x_{i\tilde{j}}^k, j \in I_i^+(\bar{U}^k). \quad (15)$$

При этом система (10) с учетом (15) для нового мультиграфа  $\bar{G}$  примет следующий вид:

$$\sum_{j \in I_i^+(\bar{U}^k)} x_{ij}^k - \sum_{j \in I_i^-(\bar{U}^k)} x_{ji}^k = \begin{cases} a_i^k, & i \in \bar{I}^k \setminus \bar{I}_k^*, \\ x_i^k + b_i, & i \in \bar{I}_k^*; \end{cases}, k \in K; \quad (16)$$

$$\sum_{(i,j) \in \bar{U}} \sum_{k \in \bar{K}(i,j)} \lambda_{ij}^{k,p} x_{ij}^k = 0, p = \overline{1, q},$$

где  $a_i^k, b_i, \bar{\lambda}_{ij}^{k,p} = const$ .

Приведем алгоритм построения множества

$$\bar{I} = \bigcup_{k \in K} I^k \setminus M_k^*,$$

то есть множества узлов нового мультиграфа  $\bar{G}$ :

1. Построить разрез  $CC(M) = \bigcup_{k \in K} CC(M^k)$  для множества отслеживаемых узлов  $M = \bigcup_{k \in K} M^k$ .
2. Найти узлы множества  $I(CC(M))$ .
3. Построить множество  $M_k^+ = I(CC(M^k)) \setminus M^k, k \in K$ .
4. Сформировать множества  $M_k^* = M^k \cup M_k^+$  и  $\bar{I}^k = I^k \setminus M_k^*, k \in K$ .

Неизвестные переменные в системе (16) — это величины потоков на исходящих дугах из узлов  $i \in \bar{I}^k, k \in K$ , а также переменные интенсивности  $x_i^k$  узлов  $i \in \bar{I}_k^*, k \in K$  нового мультиграфа  $\bar{G}$ . Этот новый мультиграф  $\bar{G}$  состоит из компонент связности. Если фиксированная компонента связности нового мультиграфа  $\bar{G}$  содержит узлы из множества  $\bar{I}_k^*$ , то, по теореме 1, с помощью теоремы 4 можно найти ранг матрицы системы (16), так как система (16) — частный случай системы (1)-(2) для этой компоненты связности.

Если фиксированная компонента связности нового мультиграфа  $\bar{G} = (\bar{I}, \bar{U})$  не содержит узлов с переменной интенсивностью из  $\bar{I}_k^*$ , то мы используем теорию декомпозиции, описанную в [2] для этой компоненты связности.

Система (16) имеет единственное решение для заданного  $M$  тогда и только тогда, когда ранг матрицы этой системы совпадает с количеством неизвестных в ней.



## Глава 6. ОСНОВНЫЕ МОМЕНТЫ РЕАЛИЗАЦИИ АЛГОРИТМА

Рассмотрим алгоритм построения опоры графа с точки зрения программной реализации.

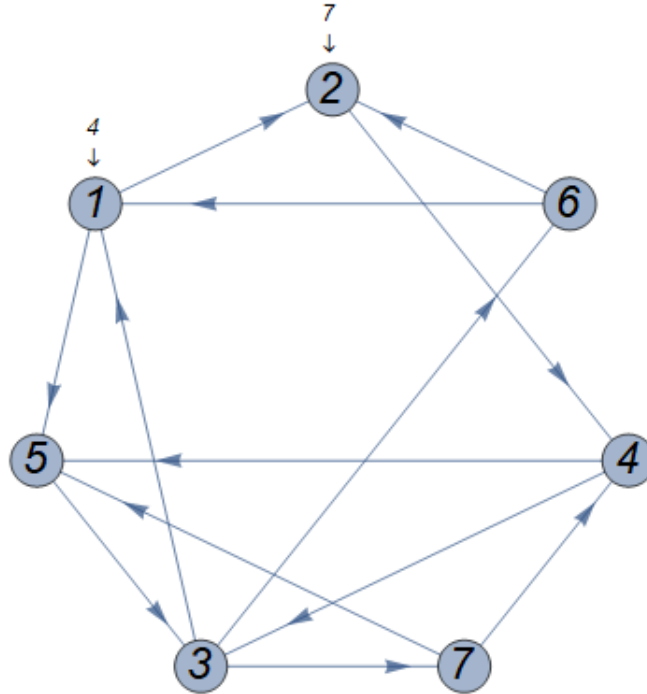


Рисунок 1 — Пример рассматриваемого графа  $G$

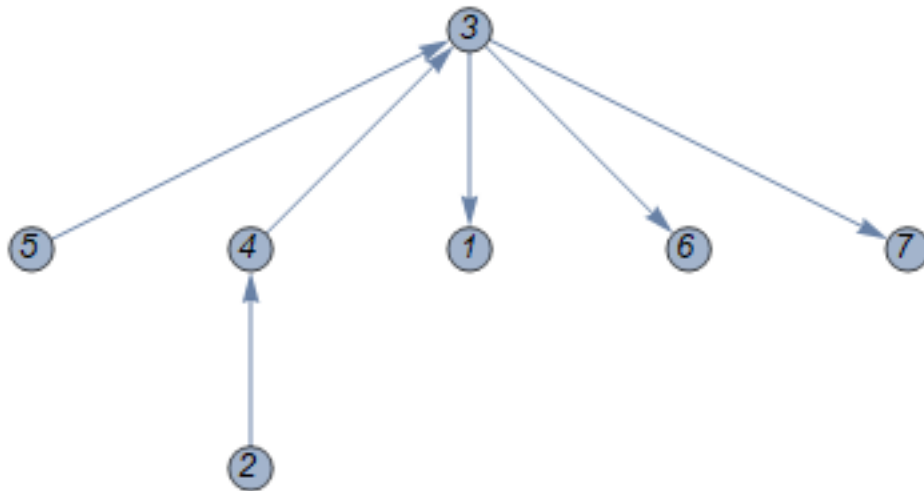


Рисунок 2 — Пример опоры графа  $G$

Также пусть  $G$  — орграф с однородным потоком и с постоянным внешним потоком. В случае постоянного внешнего потока опора графа совпадает с остовом и является деревом. Например для представленного на рисунке 1 графа  $G$  опора может иметь вид как на рисунке 2. Для хранения таких деревьев в памяти

компьютера выберем один из узлов  $r \in I^T$  в качестве корня и временно изменим направления всех дуг так, чтобы они были направлены от корня к листьям и так получим  $T^*$ .

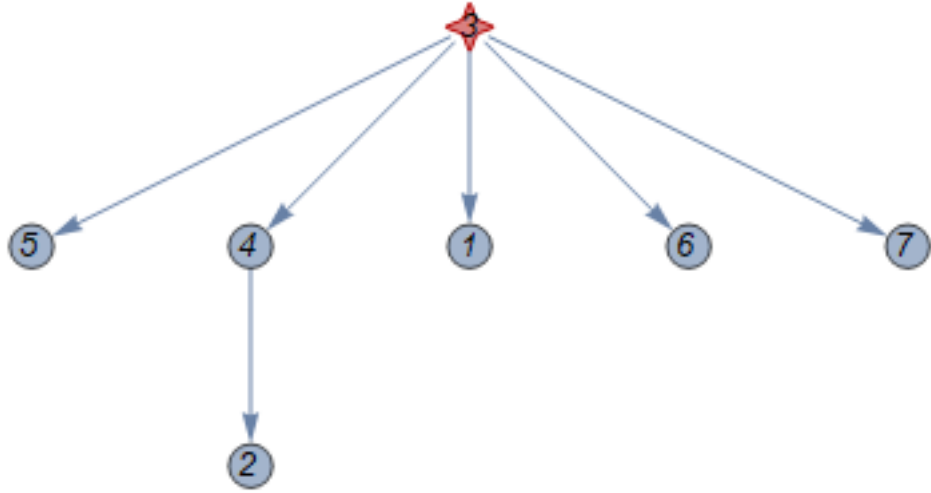


Рисунок 3 — Пример корневого дерева  $(T^*, r)$

Теперь пара  $(T, r)$  образует корневое дерево. На рисунке 3 показан пример такого корневого дерева, корень помечен как звезда.

Для представления корневого дерева в памяти воспользуемся корневыми структурами.

	1	2	3	4	5	6	7
<i>pred</i>	3	4	0	3	3	3	3
<i>dir</i>	1	-1	0	-1	-1	1	1
<i>depth</i>	1	2	0	1	1	1	1
<i>d</i>	6	3	5	1	4	7	2

Таблица 1 — Пример корневых структур для дерева  $T$

В таблице 1 приведены корневые структуры для рассматриваемого дерева. Здесь *pred* — список предков, в котором в ячейке, соответствующей номеру узла  $v \in I^T$  записан номер предыдущего узла из цепи  $C(r, v)$ , для всех  $v \in I^T \setminus \{r\}$ . У корневого узла предок не определен, так что  $pred[r] = 0$ . *dir* — список направлений,  $dir[v] = 1$  если  $(pred[v], v) \in U^T$  и  $-1$ , если  $(v, pred[v]) \in U^T$ ,  $dir[r] ::= 0$ .

Для описания дерева  $T$  приведенных структур достаточно, однако для эффективной работы с ними необходимо добавить 2 вспомогательных структуры: *depth* и *d*. *depth* — список глубин, для каждого узла хранит расстояние от него до корня. *d* — список династического обхода дерева, или его топологическая сортировка.

Алгоритм построения остова и корневых структур приведен в модуле *buildTree* Листинга A1.

Для вычисления характеристических векторов дуг используется алгоритм нахождения наименьшего общего предка (least common ancestor, lca). Утверждается, что при добавлении дуги к корневому дереву образуется ровно 1 цикл, и этот цикл состоит из добавленной дуги  $(u, v) \in U^T$ ,  $C(u, lca(u, v))$  и  $C(v, lca(u, v))$ . Таким образом вычисление  $\delta_{uv}$  сводится к вычислению  $lca(u, v)$  и подъему от каждого из концов дуги к их  $lca$ . Т.к. предварительно построен список высот, то нахождение  $lca$  может быть выполнено за  $O(|I|)$  операций подъема, и сложность вычисления всех характеристических векторов составляет  $O(|I| * (|U| - |I|))$  операций, а в силу разреженности графа  $G$  имеем  $|U| \leq c|I|$ , где  $c \ll |I|$  — некоторая константа, и тогда  $O(|I| * (|U| - |I|)) = O(|I|^2)$ .

Вычисление определителей циклов происходит по приведенным выше формулам и использует алгоритмы перемножения разреженных матриц.

Рассмотрим теперь случай графа  $G = (I, U)$  с переменным внешним потоком. В таком случае опора представляет собой лес деревьев. Однако задача построения такой опоры может быть сведена к предыдущей. Для этого добавим к графу  $G$  фиктивный узел  $i'$  и дуги для каждого узла с переменным внешним потоком в (из) вершину  $i'$ . Внешний поток в узле  $i'$  постоянный и может быть найден из глобального уравнения баланса и уравнения баланса для узла  $i'$ . Таким образом граф  $G$  преобразован в граф  $G'$  с постоянным внешним потоком и для него можно построить опору как было рассмотрено ранее. Если теперь из полученной опоры удалить фиктивный узел, то дерево распадется на лес, который и будет опорой графа  $G$ .

Заметим, что для удобства работы фиктивную вершину можно и не удалять, тогда характеристики вершин с переменным внешним потоком будут совпадать с характеристиками соответствующих фиктивных дуг, как показано в листинге B1-B2.

## Заключение

В ходе выполнения данной работы рассмотрена разреженная недоопределенная система линейных алгебраических уравнений и соответствующий ей мультиграф  $G = (I, U)$ . Для этих СЛАУ и мультиграфа приведены необходимые теоремы и определения и рассмотрены соответствующие теоретические свойства рассматриваемого мультиграфа. После этого введена соответствующая математическая модель и сформулирована задача о расположении специальных программируемых устройств (сенсоров) в узлах мультиграфа. Для сформулированной задачи приведены алгоритмические и структурные свойства построения оптимального решения. В итоге рассматриваемая задача сводится к решению одной из уже известных задач.

В системе компьютерной алгебры Wolfram Mathematica создан пользовательский пакет "FlowSolwer.m", в котором реализованы программные модули и базовые функции для построения и работы с опорой мультиграфа (мультисети) для случая однородного потока (смотри Листинг A1) и для случая переменного внешнего потока (смотри Листинг B1). Для описания опоры мультиграфа используются расширенные корневые структуры описания деревьев, включающие базовые структуры, список династического обхода и список глубин для уменьшения асимптотической сложности алгоритмов работы с ними. В пакете также реализованы модули для построения частного решения неоднородной системы уравнений баланса, общего решения однородной системы уравнений баланса, общего решения неоднородной системы уравнений баланса, построения характеристических векторов для дополнительной системы, и вычисления детерминантов циклов, а также построения общего решения для системы вида (1)-(2). Все перечисленные алгоритмы реализованы для случая однородного потока. Их работа продемонстрирована в Листинге A2 на примере графа, описанного в файле "gr.txt", содержание которого можно найти в Листинге A3.

## Список Литературы

1. L. A. Pilipchuk, Y. V. Malakhouskaya, D. R. Kincaid, and M. Lai, East-West J. of Mathematics 4, No 2, (2002), 191-201.
2. Пилипчук Л. А. Линейные неоднородные задачи потокового программирования: учеб.-метод. пособие / Л. А. Пилипчук. - Минск: БГУ, 2009. - 222 с.
3. L. A. Pilipchuk, A. S. Pilipchuk, Y. H. Pesheva, International Journal of Pure and Applied Mathematics (IJPAM) 54, No 2, (2009), 193-205.
4. I. V. Romanovski, L. A. Pilipchuk, Computing 76, (2006), 353-357.
5. L. Bianco, G. Confessore, M. Gentili, Combinatorial Aspects of the Sensor Location Problem. Annals of Operation Research 144, 1, (2006), 201-234.
6. Пилипчук, Л. А. Разреженные недоопределенные системы линейных алгебраических уравнений / Л. А. Пилипчук. – Минск : БГУ, 2012. – 260 с.

## Листинг 1.

```
BeginPackage["FlowSolver`"]
```

```
readGraph[file_,dir_] := Module[{
  fn=FileNameJoin[{dir,file}],
  stream,imod,umod,u,b
},
  stream=OpenRead[fn];
  imod=Read[stream,{Word,Number}][[2]];
  umod=Read[stream,{Word,Number}][[2]];
  u=#[ [1] ] <->#[ [2] ] &/@ReadList[stream,Expression,umod];
  b=ConstantArray[0,imod];
  (b[[Read[StringToStream[StringTake[#1,{5,-3}]],Number]]]=#2)&
  @@@ReadList[stream,{Word,Number},imod];
  {Graph[u,VertexSize->Medium, VertexLabels->{x_:>Placed[{x,Style[{
    {{
      {-b[[x]]},
      {"↑"}
    }, b[[x]]<0},
    {{
      {b[[x]]},
      {"↓"}
    }, b[[x]]>0},
    {"", True}
  } //TableForm ,Medium]],{Center,Above}]],
  VertexLabelStyle->Directive[Black,Italic,24],
  EdgeShapeFunction-> GraphElementData[{"Arrow"}],
  GraphLayout->"CircularEmbedding"],b}
]
```

```
buildTree::usage="build a tree based on graph g with a specified root"
```

```
getDepth[x_,depth_] := {
  {depth[[x]], x>0},
  {-1, True}
}
```

```
edge::usage="ret edge from i to j with specified direction"
```

```
edge[i_,j_,dir_] := {
  {i->j, dir<0},
  {j->i, dir>0},
  {i<->j, dir==0}
}
```

```

buildTree[g_?GraphQ,root_]:=Module[{
  rt={},
  pred=ConstantArray[0,VertexCount[g]],
  dir=ConstantArray[0,VertexCount[g]],
  depth=ConstantArray[0,VertexCount[g]],
  d=ConstantArray[root,VertexCount[g]],
  curD=0,lastVis=root,edgeN=ConstantArray[0,VertexCount[g]]
},
  DepthFirstScan[UndirectedGraph[g],root,{
    "PrevisitVertex"->((depth[[#]]:=getDepth[pred[[#]],depth]+1;d[[lastVis]]=#;
    lastVis=#)&),"FrontierEdge"->((pred[[Last[#]]]:=First[#];
    dir[[Last[#]]]={{
      {1, MemberQ[EdgeList[g],First[#]↔Last[#]]},
      {-1, True}
    }};
    edgeN[[Last[#]]]=Position[EdgeList[g],
    edge[First[#],Last[#],-dir[[Last[#]]]][[1,1]]];
    )&}}];
    {pred,dir,depth,d,root,edgeN}
  ]

```

```

treeQ::usage="tests if the structure t is a tree"
treeQ[t_List]:=Length[t]==6&&Length[t[[1]]]==Length[t[[2]]]&&Length[t[[1]]]==Lengt

```

```

pred[t_?treeQ]:=t[[1]]
dir[t_?treeQ]:=t[[2]]
depth[t_?treeQ]:=t[[3]]
getDepth[v_,t_?treeQ]:=getDepth[v,depth[t]]
d[t_?treeQ]:=t[[4]]
root[t_?treeQ]:=t[[5]]
reverceD[t_?treeQ]:=Module[{
  rd=ConstantArray[0,t//dir//Length]
},rd[[t//root]]=NestWhile[(rd[[d[t][[#]]]]=#;d[t][[#]]&),
t//root,(d[t][[#]]!=(t//root))&];rd]
tableForm[t_]:=TableForm[t[[1];4]],
TableHeadings->{"pred","dir","depth","d"},t//pred//Length//Range}
uNb[g_?GraphQ,t_?treeQ]:=EdgeList[g,τ↔ρ;/pred[t][[τ]]!=ρ&&pred[t][[ρ]]!=τ]

```

```

path[t_?treeQ,v_]:=NestList[pred[t][[#]]&,v,getDepth[v,t]]
alignDepth=Compile[{{vert1,_Integer},
{vert2,_Integer},{pred,_Integer,1},{depth,_Integer,1}},
NestWhile[pred[[#]]&,vert1,getDepth[#,depth]>getDepth[vert2,depth]&]
];

```

```

lcmHelper=Compile[{{vert1,_Integer},{vert2,_Integer},{pred,_Integer,1},{depth,_Integer}},
Module[{v1=vert1,v2=vert2},
v1=alignDepth[v1,v2,pred,depth];
v2=alignDepth[v2,v1,pred,depth];

NestWhile[{pred[#[[1]]],pred[#[[2]]]}&,List[v1,v2],#[[1]]!=#[[2]]&][[1]]
],{{alignDepth[_,_,_,_],_Integer}}
];
lcm[t_?treeQ,vert1_,vert2_]:=lcmHelper[vert1,vert2,pred[t],depth[t]]
pathLen[t_?treeQ,v1_,v2_]:=getDepth[v1,t]+getDepth[v2,t]-2*getDepth[lcm[v1,v2],t]
subTree[t_?treeQ,v_]:=NestWhileList[d[t][#[[1]]]&,v,getDepth[d[t][#[[1]]],t]>getDepth[v,t]&
getLeafs[t_?treeQ]:=Cases[Range[pred[t]//Length],x_/;getDepth[x,t]>=getDepth[d[t][

```

```

partSolve::usage=
"ret list of ruelles with part solve for graph g
with outside flow b and base tree, left part symbols are x"
partSolve[g_?GraphQ,b_List,tree_?treeQ,x_]:=Module[{xed,t=tree,rd,last=0},
rd=reverced[t];
xed=ConstantArray[0,g//VertexCount];
last=NestWhile[{xed[#[[1]]]+=-dir[t][#[[1]]]×b[#[[1]]];
xed[pred[t][#[[1]]]+=-dir[t][pred[t][#[[1]]]×dir[t][#[[1]]]×xed[#[[1]]];
rd[#[[1]]]&,rd[t//root]],(pred[t][#[[1]]]!=(t//root))&];
(xed[#[[1]]]+=-dir[t][#[[1]]]×b[#[[1]]])&[last];
Print[rd];
((Subscript[x,#[[1]]->0]&/@(uNb[g,t]))∪((Subscript[x,edge[#[[1]],pred[t][#[[1]]],
dir[t][#[[1]]]->xed[#[[1]]]&/@((g//VertexList)~Complement~{t//root})))
partSolve[g_?GraphQ,b_List,rootV_,x_]:=partSolve[g,b,buildTree[g,rootV],x]

```

```

Subscript[δ, τ→ρ][g_?GraphQ,t_?treeQ]:=Module[
{λ=lcm[t,τ,ρ],δ=ConstantArray[0,g//VertexCount]},
NestWhile[{δ[#[[1]]]=dir[t][#[[1]]];pred[t][#[[1]]]&,τ,#[[1]]!=λ&];
NestWhile[{δ[#[[1]]]=-dir[t][#[[1]]];pred[t][#[[1]]]&,ρ,#[[1]]!=λ&];
((Subscript[x,edge[#[[1]],pred[t][#[[1]]],dir[t][#[[1]]]->δ[#[[1]]])&
/@(g//VertexList))~Join~{Subscript[x,τ→ρ->1,Subscript[x,_->0]}
]

```

```

δ2h=Compile[{{l,_Integer,1},{pr,_Integer,1},{dep,_Integer,1},
{direct,_Integer,1},{nums,_Integer,1},{n,_Integer}},
Module[{λ,δ=SparseArray[{}],n,τ=1[[1]],ρ=1[[2]],j=1[[3]]},
λ=lcmHelper[τ,ρ,pr,dep];
NestWhile[{δ[nums[#[[1]]]=direct[#[[1]]];pr[#[[1]]]&,τ,#[[1]]!=λ&];
NestWhile[{δ[nums[#[[1]]]=-direct[#[[1]]];pr[#[[1]]]&,ρ,#[[1]]!=λ&];
δ[[j]]=1;
δ},Parallelization->False, RuntimeAttributes->{Listable}
]

```



```

δ2[{τ_,ρ_,j_},t_,n_]:=Module[{λ,δ=SparseArray[{},n]},
λ=lcm[t,τ,ρ];
NestWhile[(δ[t[[6,#]]]=dir[t][[#]];pred[t][[#]]&,τ,#!=λ&];
NestWhile[(δ[t[[6,#]]]=-dir[t][[#]];pred[t][[#]]&,ρ,#!=λ&];
δ[[j]]=1;
δ]

```

```

alignJ=Compile[{{j,_Integer},{l,_Integer,2},{ed,_Integer,1}},
NestWhile[(#1)&,j,(l[[#]]!=ed)&]];
δ1[g_?GraphQ,t_?treeQ]:=Module[
{λ,unb=uNb[g,t],unb1,tmp,δ,τ,ρ,ed,j=1},
δ=SparseArray[{},{unb//Length,g//EdgeCount}];
tmp={#[[1]],#[[2]]}&/@EdgeList[g];
unb1=Map[(ed=#;j=alignJ[j,tmp,{ed[[1]],ed[[2]]}];{#[[1]],#[[2]],j})&,unb];
δ2[#,t,g//EdgeCount]&/@unb1//SparseArray]

```

```

eqSystem[g_?GraphQ]:=Fold[ReplacePart[#1,{(#2//First)->#1[[#2//First]]-
Subscript[x,#2],(#2//Last)->#1[[#2//Last]]+Subscript[x,#2]}]&,
ConstantArray[0,g//VertexCount],g//EdgeList]

```

```

solveAll[g_?GraphQ,t_?treeQ]:=Module[{xs=Subscript[x,#]&/(g//EdgeList)},
Cases[(MapThread[#1->#2&,{xs,Parallelize[ParallelMap[Subscript[x,#]&,
uNb[g,t]].δMatr]+(ParallelMap[Subscript[χ,#]&,(g//EdgeList)]/.ps)}]],
Except[x_->x_]]]

```

```

setPred[t_?treeQ,i_,val_]:=ReplacePart[t,{1,i}->val];
setDir[t_?treeQ,i_,val_]:=ReplacePart[t,{2,i}->val];
setDepth[t_?treeQ,i_,val_]:=ReplacePart[t,{3,i}->val];
setD[t_?treeQ,i_,val_]:=ReplacePart[t,{4,i}->val];

```

```

EndPackage[]

```

## Листинг 2

```

In[1]:= ClearAll["Global`*"]
SetDirectory[NotebookDirectory[]];
Needs["FlowSolver`"]

In[4]:= readGraph1[file_, dir_] := Module[{
    fn = FileNameJoin[{dir, file}],
    stream, imod, umod, u, b, q,  $\lambda$ ,  $\alpha$ 
},
    stream = OpenRead[fn];
    imod = Read[stream, {Word, Number}][[2]];
    umod = Read[stream, {Word, Number}][[2]];
    u = #[[1]]  $\leftrightarrow$  #[[2]] & /@ ReadList[stream, Expression, umod];
    b = ConstantArray[0, imod];
    (b[[Read[StringToStream[StringTake[#1, {5, -3}]], Number]]] = #2) & @@@
    ReadList[stream, {Word, Number}, imod];
    q = Read[stream, {Word, Number}][[2]];

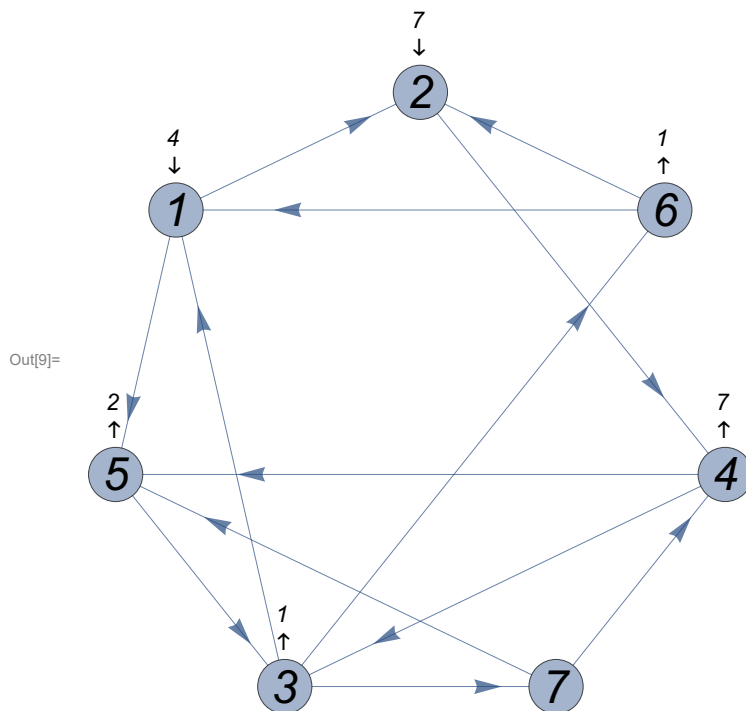
     $\lambda$  = Reap[Do[Do[Sow[Read[stream, {Word, Number}][[2]], j], {j, q}], {i, umod}]];
     $\alpha$  = ReadList[stream, {Word, Number}, q][[All, 2]];
    {Graph[u, VertexSize -> Medium, VertexLabels -> {x_ :> Placed[{x, Style[{
        {{
            {-b[[x]]},
            {"↑"}
        }, b[[x]] < 0},
        {{
            {b[[x]]},
            {"↓"}
        }, b[[x]] > 0},
        {"", True}
    } // TableForm, Medium]], {Center, Above}]],
        VertexLabelStyle -> Directive[Black, Italic, 24], EdgeShapeFunction ->
        GraphElementData[{"Arrow"}], GraphLayout -> "CircularEmbedding"], b,  $\lambda$ ,  $\alpha$ 
}
]

```

```

In[7]:= {g, b, λ, α} = readGraph1["gr.txt", NotebookDirectory[]];
λ = λ[[2]];
g

```



```

In[10]:= buildt = Timing[t = buildTree[g, 3];][[1]]
TableForm[t[[1 ;; 4]],
  TableHeadings → {"pred", "dir", "depth", "d"}, t // pred // Length // Range]

```

Out[10]= 0.

Out[11]/TableForm=

	1	2	3	4	5	6	7
pred	5	6	0	2	3	1	4
dir	-1	1	0	1	-1	-1	-1
depth	2	4	0	5	1	3	6
d	6	4	5	7	1	2	3

```

In[12]:= pst = Timing[ps = partSolve[g, b, t, x]][[1]]

```

{5, 6, 7, 2, 3, 1, 4}

Out[12]= 0.

```
In[13]:= "partSolve:"
{"x"#, "=", x̃#} & /@ (g // EdgeList) /. ps // TableForm
```

```
Out[13]= partSolve:
```

```
Out[14]//TableForm=
```

```

x1→2    =    0
x1→5    =    3
x2→4    =    7
x3→1    =    0
x3→6    =    0
x3→7    =    0
x4→3    =    0
x4→5    =    0
x5→3    =    1
x6→1    =   -1
x6→2    =    0
x7→5    =    0
x7→4    =    0

```

```
In[15]:= syst = Timing[eq = eqSystem[g];] [[1]]
"equations test:"
notOdnorEq = MapThread[#1 == -#2 &, {eq, b}];
odnorEq = (# == 0) & /@ eq;
(notOdnorEq /. x → x̃) /. ps
```

```
Out[15]= 0.
```

```
Out[16]= equations test:
```

```
Out[19]= {True, True, True, True, True, True, True}
```

```
In[20]:= matrt = Timing[δMatr = δ1[g, t];] [[1]]
```

```
Out[20]= 0.
```

```
In[25]:= solAt = Timing[s = solveAll[g, t]];
solAt[[1]]
s // TableForm
```

```
Out[26]= 0.03125
```

```
Out[27]//TableForm=
```

```

x1→5 → 3 + x3→1 + x3→6 + x3→7 - x4→3 - x4→5 - x7→5
x2→4 → 7 - x3→7 + x4→3 + x4→5 + x7→5
x5→3 → 1 + x3→1 + x3→6 + x3→7 - x4→3
x6→1 → -1 + x1→2 + x3→6 + x3→7 - x4→3 - x4→5 - x7→5
x6→2 → -x1→2 - x3→7 + x4→3 + x4→5 + x7→5
x7→4 → x3→7 - x7→5

```

```
In[29]:= dopEq = MapThread[#1 == #2 &, {Flatten[λ. {x# & /@ EdgeList[g]}T], α}];
dopEq // TableForm
```

```

9 x1→2 - 6 x1→5 - 5 x2→4 + 10 x3→1 + 3 x3→6 + 3 x3→7 + 5 x4→5 - 4 x5→3 +
  3 x6→1 - 10 x6→2 + 8 x7→4 - 7 x7→5 == 7
8 x1→2 - 7 x1→5 - 4 x2→4 + 10 x3→1 + 3 x3→6 + 6 x3→7 + 10 x4→3 +
  8 x4→5 + x5→3 + 5 x6→1 + 2 x6→2 + 10 x7→4 + x7→5 == -1
-9 x1→2 - 4 x1→5 + 3 x2→4 - 6 x3→7 - x4→3 + 4 x4→5 + 2 x5→3 +
  7 x6→1 + x6→2 - 4 x7→4 - 7 x7→5 == -4

```

```
In[40]:=  $\Lambda = \lambda.\delta\text{Matr}^T$ ;
"cycle det's:"
Grid[ $\Lambda$ , Frame  $\rightarrow$  All]
```

Out[41]= cycle det's:

Out[42]=

22	0	-4	19	-8	-7	-27
11	4	2	17	9	8	-9
-3	-2	5	-9	-2	5	-2

```
In[43]:= "Uc="
          Uc = {1, 2, 3}
          "Unc="
          Unc = {4, 5, 6, 7}
```

Out[43]= U<sub>c</sub> =

Out[44]= {1, 2, 3}

Out[45]= U<sub>nc</sub> =

Out[46]= {4, 5, 6, 7}

```
In[47]:=  $\Delta_c = \Lambda[[\text{All}, U_c]]$ ;
           $\Delta_{nc} = \Lambda[[\text{All}, U_{nc}]]$ ;
          " $\Delta_c$ ="
           $\Delta_c$  // MatrixForm
```

Out[49]=  $\Delta_c$  =

Out[50]//MatrixForm=

$$\begin{pmatrix} 22 & 0 & -4 \\ 11 & 4 & 2 \\ -3 & -2 & 5 \end{pmatrix}$$

```
In[51]:= "det( $\Delta_c$ )="
          Det[ $\Delta_c$ ]
```

Out[51]= det( $\Delta_c$ ) =

Out[52]= 568

```
In[53]:= "UT="
          utind = Cases[t[[6]],  $\xi_$  /;  $\xi \neq 0$ ];
          UT = EdgeList[g][[utind]]
```

Out[53]= U<sub>T</sub> =

Out[55]= {1  $\leftrightarrow$  5, 6  $\leftrightarrow$  2, 2  $\leftrightarrow$  4, 5  $\leftrightarrow$  3, 6  $\leftrightarrow$  1, 7  $\leftrightarrow$  4}

```
In[56]:= "UNb="
          UNb = uNb[g, t]
```

Out[56]= U<sub>Nb</sub> =

Out[57]= {1  $\leftrightarrow$  2, 3  $\leftrightarrow$  1, 3  $\leftrightarrow$  6, 3  $\leftrightarrow$  7, 4  $\leftrightarrow$  3, 4  $\leftrightarrow$  5, 7  $\leftrightarrow$  5}

```

In[58]:= A =  $\alpha - \lambda \cdot \{\tilde{x}_{\#} \& /@ \text{EdgeList}[g]\}^T / . \text{ps};$ 
"А="
A // MatrixForm

Out[59]= A=

Out[60]//MatrixForm=

$$\begin{pmatrix} 67 \\ 52 \\ -8 \end{pmatrix}$$


In[64]:=  $\beta = A - \Delta_{nc} \cdot \{x_{\#} \& /@ U_{Nb}[[U_{nc}]]\}^T;$ 
"β="
β // TableForm

Out[65]= β=

Out[66]//TableForm=

$$\begin{array}{l} 67 - 19 x_{3 \leftrightarrow 7} + 8 x_{4 \leftrightarrow 3} + 7 x_{4 \leftrightarrow 5} + 27 x_{7 \leftrightarrow 5} \\ 52 - 17 x_{3 \leftrightarrow 7} - 9 x_{4 \leftrightarrow 3} - 8 x_{4 \leftrightarrow 5} + 9 x_{7 \leftrightarrow 5} \\ -8 + 9 x_{3 \leftrightarrow 7} + 2 x_{4 \leftrightarrow 3} - 5 x_{4 \leftrightarrow 5} + 2 x_{7 \leftrightarrow 5} \end{array}$$


In[67]:= "решаем уравнение  $\Delta_c x_c = \beta$ :"
xc = LinearSolve[ $\Delta_c$ , β]

Out[67]= решаем уравнение  $\Delta_c x_c = \beta$ :

Out[68]=  $\left\{ \left\{ \frac{1}{71} (237 - 56 x_{3 \leftrightarrow 7} + 19 x_{4 \leftrightarrow 3} + 3 x_{4 \leftrightarrow 5} + 94 x_{7 \leftrightarrow 5}) \right\}, \right.$ 

$$\left\{ \frac{1}{568} (1713 - 1299 x_{3 \leftrightarrow 7} - 1546 x_{4 \leftrightarrow 3} - 771 x_{4 \leftrightarrow 5} - 941 x_{7 \leftrightarrow 5}) \right\},$$


$$\left. \left\{ \frac{1}{284} (457 + 117 x_{3 \leftrightarrow 7} - 150 x_{4 \leftrightarrow 3} - 431 x_{4 \leftrightarrow 5} + 151 x_{7 \leftrightarrow 5}) \right\} \right\}$$


In[69]:= xcp = MapThread[x#1 → #2 &, {UNb[[Uc]], Flatten[xc]}]

Out[69]=  $\left\{ x_{1 \leftrightarrow 2} \rightarrow \frac{1}{71} (237 - 56 x_{3 \leftrightarrow 7} + 19 x_{4 \leftrightarrow 3} + 3 x_{4 \leftrightarrow 5} + 94 x_{7 \leftrightarrow 5}), \right.$ 

$$x_{3 \leftrightarrow 1} \rightarrow \frac{1}{568} (1713 - 1299 x_{3 \leftrightarrow 7} - 1546 x_{4 \leftrightarrow 3} - 771 x_{4 \leftrightarrow 5} - 941 x_{7 \leftrightarrow 5}),$$


$$x_{3 \leftrightarrow 6} \rightarrow \frac{1}{284} (457 + 117 x_{3 \leftrightarrow 7} - 150 x_{4 \leftrightarrow 3} - 431 x_{4 \leftrightarrow 5} + 151 x_{7 \leftrightarrow 5}) \left. \right\}$$


In[70]:= "общее решение:"
xsol = (s /. xcp) ~Join~ xcp;
xsol // TableForm

Out[70]= общее решение:
```

$$\begin{aligned}
x_{1 \rightarrow 5} &\rightarrow 3 + x_{3 \rightarrow 7} - x_{4 \rightarrow 3} - x_{4 \rightarrow 5} + \frac{1}{568} (1713 - 1299 x_{3 \rightarrow 7} - 1546 x_{4 \rightarrow 3} - 771 x_{4 \rightarrow 5} - 941 x_{7 \rightarrow 5}) - x_{7 \rightarrow 5} + \\
&\quad \frac{1}{284} (457 + 117 x_{3 \rightarrow 7} - 150 x_{4 \rightarrow 3} - 431 x_{4 \rightarrow 5} + 151 x_{7 \rightarrow 5}) \\
x_{2 \rightarrow 4} &\rightarrow 7 - x_{3 \rightarrow 7} + x_{4 \rightarrow 3} + x_{4 \rightarrow 5} + x_{7 \rightarrow 5} \\
x_{5 \rightarrow 3} &\rightarrow 1 + x_{3 \rightarrow 7} - x_{4 \rightarrow 3} + \frac{1}{568} (1713 - 1299 x_{3 \rightarrow 7} - 1546 x_{4 \rightarrow 3} - 771 x_{4 \rightarrow 5} - 941 x_{7 \rightarrow 5}) \\
&\quad + \frac{1}{284} (457 + 117 x_{3 \rightarrow 7} - 150 x_{4 \rightarrow 3} - 431 x_{4 \rightarrow 5} + 151 x_{7 \rightarrow 5}) \\
x_{6 \rightarrow 1} &\rightarrow -1 + x_{3 \rightarrow 7} - x_{4 \rightarrow 3} - x_{4 \rightarrow 5} - x_{7 \rightarrow 5} + \frac{1}{71} (237 - 56 x_{3 \rightarrow 7} + 19 x_{4 \rightarrow 3} + 3 x_{4 \rightarrow 5} + 94 x_{7 \rightarrow 5}) \\
&\quad + \frac{1}{284} (457 + 117 x_{3 \rightarrow 7} - 150 x_{4 \rightarrow 3} - 431 x_{4 \rightarrow 5} + 151 x_{7 \rightarrow 5}) \\
x_{6 \rightarrow 2} &\rightarrow -x_{3 \rightarrow 7} + x_{4 \rightarrow 3} + x_{4 \rightarrow 5} + \frac{1}{71} (-237 + 56 x_{3 \rightarrow 7} - 19 x_{4 \rightarrow 3} - 3 x_{4 \rightarrow 5} - 94 x_{7 \rightarrow 5}) + x_{7 \rightarrow 5} \\
x_{7 \rightarrow 4} &\rightarrow x_{3 \rightarrow 7} - x_{7 \rightarrow 5} \\
x_{1 \rightarrow 2} &\rightarrow \frac{1}{71} (237 - 56 x_{3 \rightarrow 7} + 19 x_{4 \rightarrow 3} + 3 x_{4 \rightarrow 5} + 94 x_{7 \rightarrow 5}) \\
x_{3 \rightarrow 1} &\rightarrow \frac{1}{568} (1713 - 1299 x_{3 \rightarrow 7} - 1546 x_{4 \rightarrow 3} - 771 x_{4 \rightarrow 5} - 941 x_{7 \rightarrow 5}) \\
x_{3 \rightarrow 6} &\rightarrow \frac{1}{284} (457 + 117 x_{3 \rightarrow 7} - 150 x_{4 \rightarrow 3} - 431 x_{4 \rightarrow 5} + 151 x_{7 \rightarrow 5})
\end{aligned}$$

```
In[73]:= "eq test:"
Simplify[dopEq /. s /. xcp]
```

```
Out[73]= eq test:
```

```
Out[74]= {True, True, True}
```

### Листинг 3

```
/*||*/ 7
/*|U|*/ 13
{1,2}
{1,5}
{2,4}
{3,1}
{3,6}
{3,7}
{4,3}
{4,5}
{5,3}
{6,1}
{6,2}
{7,5}
{7,4}
/*b_1*/ 4
/*b_2*/ 7
/*b_3*/ -1
/*b_4*/ -7
/*b_5*/ -2
/*b_6*/ -1
/*b_7*/ 0
/*q*/ 3
/*lambda_1_2_1*/ 9
/*lambda_1_2_2*/ 8
/*lambda_1_2_3*/ -9
/*lambda_1_5_1*/ -6
/*lambda_1_5_2*/ -7
/*lambda_1_5_3*/ -4
/*lambda_2_4_1*/ -5
/*lambda_2_4_2*/ -4
/*lambda_2_4_3*/ 3
/*lambda_3_1_1*/ 10
/*lambda_3_1_2*/ 10
/*lambda_3_1_3*/ 0
/*lambda_3_6_1*/ 3
/*lambda_3_6_2*/ 3
/*lambda_3_6_3*/ 0
/*lambda_3_7_1*/ 3
/*lambda_3_7_2*/ 6
/*lambda_3_7_3*/ -6
/*lambda_4_3_1*/ 0
/*lambda_4_3_2*/ 10
/*lambda_4_3_3*/ -1
/*lambda_4_5_1*/ 5
/*lambda_4_5_2*/ 8
/*lambda_4_5_3*/ 4
/*lambda_5_3_1*/ -4
```



```

/*lambda_5_3_2*/ 1
/*lambda_5_3_3*/ 2
/*lambda_6_1_1*/ 3
/*lambda_6_1_2*/ 5
/*lambda_6_1_3*/ 7
/*lambda_6_2_1*/ -10
/*lambda_6_2_2*/ 2
/*lambda_6_2_3*/ 1
/*lambda_7_5_1*/ -7
/*lambda_7_5_2*/ 1
/*lambda_7_5_3*/ -7
/*lambda_7_4_1*/ 8
/*lambda_7_4_2*/ 10
/*lambda_7_4_3*/ -4
/*alpha_1*/ 7
/*alpha_2*/ -1
/*alpha_3*/ -4

```

Листинг 1.

```

BeginPackage["FlowSolver`"]

readGraph[file_, dir_] := Module[{
  fn = FileNameJoin[{dir, file}],
  stream, imod, umod, u, b
},
  stream = OpenRead[fn];
  imod = Read[stream, {Word, Number}] [[2]];
  umod = Read[stream, {Word, Number}] [[2]];
  u = #[[1]] <-> #[[2]] &/@ ReadList[stream, Expression, umod];
  b = ConstantArray[0, imod];
  (b[[Read[StringToStream[StringTake[#, {5, -3}]], Number]]] = #2) &
  @@@ ReadList[stream, {Word, Number}, imod];
  {Graph[u, VertexSize -> Medium, VertexLabels -> {x_ :> Placed[{x, Style[{
    {
      {-b[[x]]},
      {"^"}
    }, b[[x]] < 0},
    {
      {b[[x]]},
      {"v"}
    }, b[[x]] > 0},
    {"", True}
  } // TableForm, Medium]], {Center, Above}]],
  VertexLabelStyle -> Directive[Black, Italic, 24],
  EdgeShapeFunction -> GraphElementData[{"Arrow"}],
  GraphLayout -> "CircularEmbedding"], b}
]

buildTree::usage = "build a tree based on graph g with a specified root"

getDepth1[x_, depth_] := {
  {depth[[x]], x > 0},
  {-1, True}
}

edge::usage = "ret edge from i to j with specified direction"
edge[i_, j_, dir_] := {
  {i -> j, dir < 0},
  {j -> i, dir > 0},
  {i -> j, dir == 0}
}

```

```

buildTree[g_?GraphQ,root_]:=Module[{
ng,
  rt={},
  tmp,
  pred=ConstantArray[0,VertexCount[g]+1],
  dir=ConstantArray[0,VertexCount[g]+1],
  depth=ConstantArray[0,VertexCount[g]+1],
  d=ConstantArray[VertexCount[g]+1,VertexCount[g]+1],
  curD=0,lastVis=0,edgeN=ConstantArray[0,VertexCount[g]+1],
  lroot=VertexCount[g]+1
},
ng=EdgeAdd[VertexAdd[g,lroot],(lroot↔#)&/@root];
BreadthFirstScan[UndirectedGraph[ng],lroot,{
  "PrevisitVertex"->((If[#==0,,depth[[#]]=getDepth1[pred[[#]],depth]+1;])&),
  "FrontierEdge"->((pred[[Last[#]]]=First[#];
  dir[[Last[#]]]={{
    {1,MemberQ[EdgeList[ng],First[#]↔Last[#]]},
    {-1,True}
  }});
tmp=Position[EdgeList[ng],
edge[First[#],Last[#],-dir[[Last[#]]]]];
edgeN[[Last[#]]=If[Length[tmp]==0,0,tmp[[1,1]]];
AppendTo[rt,First[#]↔Last[#]]&)}];
DepthFirstScan[UndirectedGraph[Graph[rt]],lroot,{"PrevisitVertex"->((If[lastVis==0,,
  {{pred,dir,depth,d,lroot,edgeN,Graph[rt,VertexSize->Medium,VertexLabels->Placed[
    (*{pred,dir,depth,d,root,edgeN}*)
  ]

```

```

treeQ::usage="tests if the structure t is a tree"
treeQ[t_List]:=Length[t]==7&&Length[t[[1]]]==Length[t[[2]]]&&Length[t[[1]]]==Length

```

```

pred[t_]:=t[[1]]
dir[t_]:=t[[2]]
depth[t_]:=t[[3]]
getDepth[v_,t_?treeQ]:=getDepth1[v,depth[t]]
d[t_]:=t[[4]]
root[t_]:=t[[5]]
reverced[t_]:=Module[{
rd=ConstantArray[0,t//dir//Length]
},rd[[t//root]]=NestWhile[(rd[[d[t][[#]]]]==#;d[t][[#]])&,
t//root,(d[t][[#]]!=(t//root))&];rd]
tableForm[t_]:=TableForm[t[[1;;4]],
TableHeadings->{{"pred","dir","depth","d"},t//pred//Length//Range}]
uNb[g_?GraphQ,t_]:=EdgeList[g,t↔ρ_/;pred[t][[τ]]!=ρ&&pred[t][[ρ]]!=τ]

```

```

path[t_,v_]:=NestList[pred[t][[#]]&,v,getDepth[v,t]]
alignDepth=Compile[{{vert1,_Integer},
{vert2,_Integer},{pred,_Integer,1},{depth,_Integer,1}},
NestWhile[pred[[#]]&,vert1,getDepth1[#,depth]>getDepth1[vert2,depth]&]
];

```

```

lcmHelper=Compile[{{vert1,_Integer},{vert2,_Integer},{pred,_Integer,1},{depth,_Integer}},
Module[{v1=vert1,v2=vert2},
v1=alignDepth[v1,v2,pred,depth];
v2=alignDepth[v2,v1,pred,depth];

NestWhile[{pred[[#][1]]},pred[[#][2]]]&,List[v1,v2],#[[1]]!=#[[2]]&][[1]]
],{{alignDepth[_,_,_],_Integer}}
];
lcm[t_,vert1_,vert2_]:=lcmHelper[vert1,vert2,pred[t],depth[t]]
pathLen[t_,v1_,v2_]:=getDepth[v1,t]+getDepth[v2,t]-2*getDepth[lcm[v1,v2],t]
subTree[t_,v_]:=NestWhileList[d[t][[#]]&,v,getDepth[d[[#]],t]>getDepth[v,t]&]
getLeafs[t_]:=Cases[Range[pred[t]//Length],x_/;getDepth[x,t]>=getDepth[d[t][[x]],t]]

```

```

partSolve::usage=
"ret list of rullles with part solve for graph g
with outside flow b and base tree, left part symbols are x"
partSolve[g_?GraphQ,b_List,tree_,x_]:=Module[{xed,t=tree,rd,last=0},
rd=reverced[t];
xed=ConstantArray[0,g//VertexCount];
last=NestWhile[(xed[[#]]+=-dir[t][[#]]×b[[#]]];
xed[[pred[t][[#]]]+=-dir[t][[pred[t][[#]]]]×dir[t][[#]]×xed[[#]]];
rd[[#]]&,rd[[t//root]],(pred[t][[#]]!=(t//root))&];
(xed[[#]]+=-dir[t][[#]]×b[[#]])&[last];

((Subscript[x,#]->0)&/@(uNb[g,t]))∪((Subscript[x,edge[#,pred[t][[#]]],
dir[t][[#]]]->xed[[#]])&/@(g//VertexList)~Complement~{t//root})]
partSolve[g_?GraphQ,b_List,rootV_,x_]:=partSolve[g,b,buildTree[g,rootV],x]

```

```

Subscript[δ, τ→ρ][g_?GraphQ,t_]:=Module[
{λ=lcm[t,τ,ρ],δ=ConstantArray[0,g//VertexCount]},
NestWhile[(δ[[#]]=dir[t][[#]];pred[t][[#]]&,τ,#!=λ&];
NestWhile[(δ[[#]]=-dir[t][[#]];pred[t][[#]]&,ρ,#!=λ&];
((Subscript[x,edge[#,pred[t][[#]],dir[t][[#]]]->δ[[#]])&
/@@(g//VertexList))~Join~{Subscript[x,τ→ρ]->1,Subscript[x,_]->0}
]

```

```

δ2h=Compile[{{l,_Integer,1},{pr,_Integer,1},{dep,_Integer,1},
{direct,_Integer,1},{nums,_Integer,1},{n,_Integer}},
Module[{λ,δ=SparseArray[{}],n,τ=1[[1]],ρ=1[[2]],j=1[[3]]},
λ=lcmHelper[τ,ρ,pr,dep];
NestWhile[(δ[[nums[[#]]]]=direct[[#]];pr[[#]]&,τ,#!=λ&];
NestWhile[(δ[[nums[[#]]]]=-direct[[#]];pr[[#]]&,ρ,#!=λ&];
δ[[j]]=1;
δ],Parallelization->False, RuntimeAttributes -> {Listable}
]

```

```

 $\delta 2[\{\tau\_,\rho\_,\mathbf{j}\_ \},\mathbf{t}\_,\mathbf{n}\_] := \text{Module}[\{\lambda,\delta = \text{SparseArray}[\{\},\mathbf{n}\}\},
\lambda = \text{lcm}[\mathbf{t},\tau,\rho];
\text{NestWhile}[(\delta[[\mathbf{t}[[6,\#]]]] = \text{dir}[\mathbf{t}][[\#]];\text{pred}[\mathbf{t}][[\#]]) \&,\tau,\# \neq \lambda \&];
\text{NestWhile}[(\delta[[\mathbf{t}[[6,\#]]]] = -\text{dir}[\mathbf{t}][[\#]];\text{pred}[\mathbf{t}][[\#]]) \&,\rho,\# \neq \lambda \&];
\delta[[\mathbf{j}]] = 1;
\delta]$ 
```

```

alignJ = Compile[{{j,_Integer},{1,_Integer,2},{ed,_Integer,1}},
NestWhile[(\# + 1) \&,j,(1[[\#]] \neq ed) \&];
 $\delta 1[\mathbf{g\_?GraphQ},\mathbf{t\_}] := \text{Module}[\{\lambda,\text{unb} = \text{uNb}[\mathbf{g},\mathbf{t}],\text{unb1},\text{tmp},\tau,\rho,\text{ed},\mathbf{j} = 1\},

\text{tmp} = \{\#[[1]],\#[[2]]\} \& / @ \text{EdgeList}[\mathbf{g}];
\text{unb1} = \text{Map}[(\text{ed} = \#;\mathbf{j} = \text{alignJ}[\mathbf{j},\text{tmp},\{\text{ed}[[1]],\text{ed}[[2]]\}];\{\#[[1]],\#[[2]],\mathbf{j}\}) \&,\text{unb}];

 $\delta 2[\#, \mathbf{t}, \mathbf{g} // \text{EdgeCount}] \& / @ \text{unb1} // \text{SparseArray}]$$ 
```

```

eqSystem[\mathbf{g\_?GraphQ}] := Fold[ReplacePart[\#1,\{(\#2//First) -> \#1[(\#2//First)] -
Subscript[x,\#2],(\#2//Last) -> \#1[(\#2//Last)] + Subscript[x,\#2]\}]\&,
ConstantArray[0,\mathbf{g} // \text{VertexCount}],\mathbf{g} // \text{EdgeList}]

```

```

solveAll[\mathbf{g\_?GraphQ},\mathbf{t\_}] := \text{Module}[\{\mathbf{x} = \text{Subscript}[x,\#] \& / @ (\mathbf{g} // \text{EdgeList})\},
Cases[(MapThread[\#1 -> \#2 \&,\{\mathbf{x},\text{Parallelize}[\text{ParallelMap}[\text{Subscript}[x,\#] \&,
\text{uNb}[\mathbf{g},\mathbf{t}]] . \delta \text{Matr}] + (\text{ParallelMap}[\text{Subscript}[\tilde{x},\#] \& ,(\mathbf{g} // \text{EdgeList})] /. ps)\}],
Except[x_->x_]]]

```

```

setPred[\mathbf{t\_},\mathbf{i\_},\text{val\_}] := \text{ReplacePart}[\mathbf{t},\{1,\mathbf{i}\} -> \text{val}];
setDir[\mathbf{t\_},\mathbf{i\_},\text{val\_}] := \text{ReplacePart}[\mathbf{t},\{2,\mathbf{i}\} -> \text{val}];
setDepth[\mathbf{t\_},\mathbf{i\_},\text{val\_}] := \text{ReplacePart}[\mathbf{t},\{3,\mathbf{i}\} -> \text{val}];
setD[\mathbf{t\_},\mathbf{i\_},\text{val\_}] := \text{ReplacePart}[\mathbf{t},\{4,\mathbf{i}\} -> \text{val}];

```

```

EndPackage[]

```

## Листинг 2

```

In[4]:= ClearAll["Global`*"]
SetDirectory[NotebookDirectory[]];
Needs["FlowSolver`"]

In[*]:= readGraph1[file_, dir_] := Module[{
    fn = FileNameJoin[{dir, file}],
    stream, imod, umod, u, b, q,  $\lambda$ ,  $\alpha$ 
},
    stream = OpenRead[fn];
    imod = Read[stream, {Word, Number}][[2]];
    umod = Read[stream, {Word, Number}][[2]];
    u = #[[1]]  $\leftrightarrow$  #[[2]] & /@ ReadList[stream, Expression, umod];
    b = ConstantArray[0, imod];
    (b[[Read[StringToStream[StringTake[#1, {5, -3}]], Number]]] = #2) & @@@
    ReadList[stream, {Word, Expression}, imod];
    q = Read[stream, {Word, Number}][[2]];

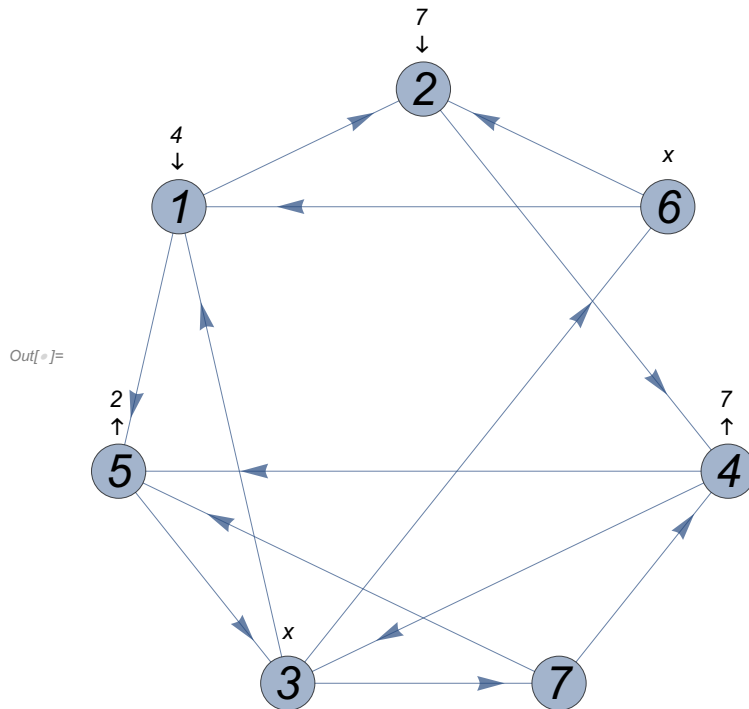
     $\lambda$  = Reap[Do[Do[Sow[Read[stream, {Word, Number}][[2]], j], {j, q}], {i, umod}]];
     $\alpha$  = ReadList[stream, {Word, Number}, q][[All, 2]];
    {Graph[u, VertexSize -> Medium, VertexLabels -> {xx_ :> Placed[{xx, Style[{ {
        {x, SameQ[b[[xx]], x}], {{
            {-b[[xx]]},
            {"^"}
        }, b[[xx]] < 0},
        {{
            {b[[xx]]},
            {"v"}
        }, b[[xx]] > 0},
        {"", True}
    } // TableForm, Medium]], {Center, Above}]],
        VertexLabelStyle -> Directive[Black, Italic, 24], EdgeShapeFunction ->
        GraphElementData[{"Arrow"}], GraphLayout -> "CircularEmbedding"], b,  $\lambda$ ,  $\alpha$ 
}
]

```

```
In[ ]:= {g, b, λ, α} = readGraph1["gr.txt", NotebookDirectory[]];
λ = λ[[2]]
```

**g**

```
Out[ ]:= {{9, -6, -5, 10, 3, 3, 0, 5, -4, 3, -10, -7, 8},
{8, -7, -4, 10, 3, 6, 10, 8, 1, 5, 2, 1, 10}, {-9, -4, 3, 0, 0, -6, -1, 4, 2, 7, 1, -7, -4}}
```



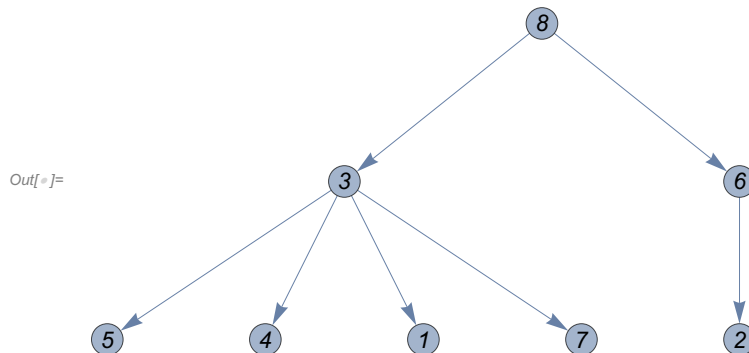
```
In[ ]:= buildt = Timing[{t, g} = buildTree[g, {6, 3}]]][[1]]
TableForm[t[[1 ;; 4]],
TableHeadings → {"pred", "dir", "depth", "d"}, t // pred // Length // Range]]
```

Out[ ]:= 0.015625

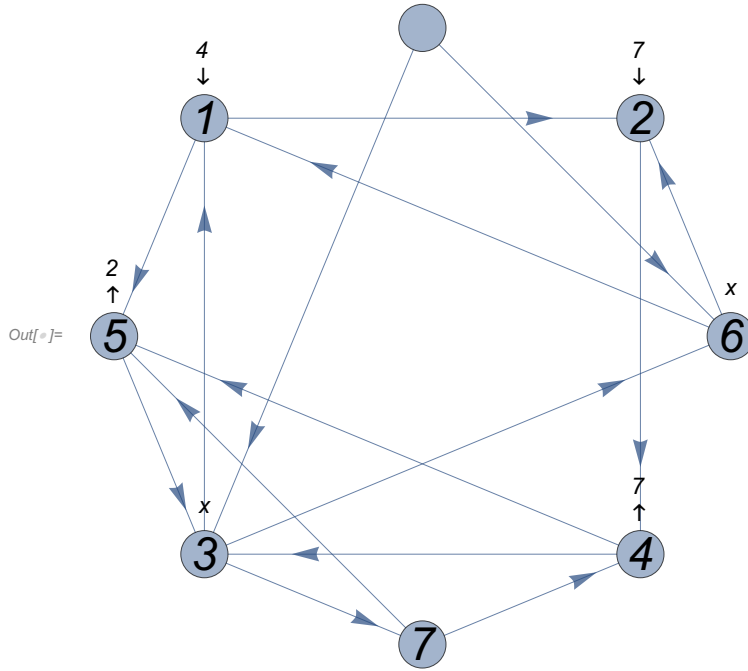
Out[ ]//TableForm=

	1	2	3	4	5	6	7	8
pred	3	6	8	3	3	8	3	0
dir	1	1	1	-1	-1	1	1	0
depth	2	2	1	2	2	1	2	0
d	7	8	5	1	4	2	6	3

```
In[ ]:= t[[7]]
```



In[\*]:= g



In[\*]:= matrt = Timing[ $\delta$ Matr =  $\delta$ 1[g, t]];

TableForm[ $\delta$ Matr,  
TableHeadings  $\rightarrow$  {uNb[g, t],  $\delta$ If[#[[1]] == VertexCount[g], #[[2]], #] & /@ EdgeList[g]}]

Out[\*]//TableForm=

	$\delta_{1 \rightarrow 2}$	$\delta_{1 \rightarrow 5}$	$\delta_{2 \rightarrow 4}$	$\delta_{3 \rightarrow 1}$	$\delta_{3 \rightarrow 6}$	$\delta_{3 \rightarrow 7}$	$\delta_{4 \rightarrow 3}$	$\delta_{4 \rightarrow 5}$	$\delta_{5 \rightarrow 3}$	$\delta_{6 \rightarrow 1}$	$\delta_{7 \rightarrow 2}$
1 $\rightarrow$ 2	1	0	0	1	0	0	0	0	0	0	0
1 $\rightarrow$ 5	0	1	0	1	0	0	0	0	1	0	0
2 $\rightarrow$ 4	0	0	1	0	0	0	1	0	0	0	0
3 $\rightarrow$ 6	0	0	0	0	1	0	0	0	0	0	0
4 $\rightarrow$ 5	0	0	0	0	0	0	-1	1	1	0	0
6 $\rightarrow$ 1	0	0	0	-1	0	0	0	0	0	1	0
7 $\rightarrow$ 5	0	0	0	0	0	1	0	0	1	0	0
7 $\rightarrow$ 4	0	0	0	0	0	1	1	0	0	0	0

In[\*]:= dopEq = MapThread[ $\#1 == \#2 \&$ ,  
{Flatten[ $\lambda.$ { $x_{\#}$  & /@ EdgeList[g, a\_  $\rightarrow$  \_ /; a  $\neq$  VertexCount[g]}]}<sup>T</sup>],  $\alpha$ ];  
dopEq // TableForm

Out[\*]//TableForm=

9  $x_{1 \rightarrow 2}$  - 6  $x_{1 \rightarrow 5}$  - 5  $x_{2 \rightarrow 4}$  + 10  $x_{3 \rightarrow 1}$  + 3  $x_{3 \rightarrow 6}$  + 3  $x_{3 \rightarrow 7}$  + 5  $x_{4 \rightarrow 5}$  - 4  $x_{5 \rightarrow 3}$  + 3  $x_{6 \rightarrow 1}$  - 10  $x_{6 \rightarrow 2}$  + 8  $x_{7 \rightarrow 4}$  - 7  $x_{7 \rightarrow 5}$  =  
8  $x_{1 \rightarrow 2}$  - 7  $x_{1 \rightarrow 5}$  - 4  $x_{2 \rightarrow 4}$  + 10  $x_{3 \rightarrow 1}$  + 3  $x_{3 \rightarrow 6}$  + 6  $x_{3 \rightarrow 7}$  + 10  $x_{4 \rightarrow 3}$  + 8  $x_{4 \rightarrow 5}$  +  $x_{5 \rightarrow 3}$  + 5  $x_{6 \rightarrow 1}$  + 2  $x_{6 \rightarrow 2}$  + 10  $x_{7 \rightarrow 4}$  +  
- 9  $x_{1 \rightarrow 2}$  - 4  $x_{1 \rightarrow 5}$  + 3  $x_{2 \rightarrow 4}$  - 6  $x_{3 \rightarrow 7}$  -  $x_{4 \rightarrow 3}$  + 4  $x_{4 \rightarrow 5}$  + 2  $x_{5 \rightarrow 3}$  + 7  $x_{6 \rightarrow 1}$  +  $x_{6 \rightarrow 2}$  - 4  $x_{7 \rightarrow 4}$  - 7  $x_{7 \rightarrow 5}$  == -4

In[\*]:=  $\Lambda = \lambda. (\delta$ Matr[[All, 1 ;; -3]])<sup>T</sup>;  
"cicle det's:"  
Grid[ $\Lambda$ , Frame  $\rightarrow$  All]

Out[\*]= cicle det's:

Out[\*]=

29	0	-15	3	1	-7	-8	11
16	4	8	3	-1	-5	8	26
-10	-2	3	0	7	7	-11	-11



```

In[*]:= "Uc="
      Uc = {1, 2, 3}
      "Unc="
      Unc = {4, 5, 6, 7, 8}

Out[*]:= Uc=

Out[*]:= {1, 2, 3}

Out[*]:= Unc=

Out[*]:= {4, 5, 6, 7, 8}

In[*]:= Δc = Δ[All, Uc];
      Δnc = Δ[All, Unc];
      "Δc="
      Δc // MatrixForm

Out[*]:= Δc=

Out[*]//MatrixForm=

$$\begin{pmatrix} 29 & 0 & -15 \\ 16 & 4 & 8 \\ -10 & -2 & 3 \end{pmatrix}$$


In[*]:= "det (Δc) ="
      Det[Δc]

Out[*]:= det (Δc) =

Out[*]:= 692

In[*]:= "UT="
      utind = Cases[t[[6]], ξ- /; ξ ≠ 0];
      UT = EdgeList[g][[utind]]

Out[*]:= UT=

Out[*]:= {3 ↔ 1, 6 ↔ 2, 8 ↔ 3, 4 ↔ 3, 5 ↔ 3, 8 ↔ 6, 3 ↔ 7}

In[*]:= "UNb="
      UNb = uNb[g, t]

Out[*]:= UNb=

Out[*]:= {1 ↔ 2, 1 ↔ 5, 2 ↔ 4, 3 ↔ 6, 4 ↔ 5, 6 ↔ 1, 7 ↔ 5, 7 ↔ 4}

In[*]:= ps = X̃# → X̃# & /@ EdgeList[g]

Out[*]:= {X̃1→2 → X̃1→2, X̃1→5 → X̃1→5, X̃2→4 → X̃2→4, X̃3→1 → X̃3→1, X̃3→6 → X̃3→6,
      X̃3→7 → X̃3→7, X̃4→3 → X̃4→3, X̃4→5 → X̃4→5, X̃5→3 → X̃5→3, X̃6→1 → X̃6→1,
      X̃6→2 → X̃6→2, X̃7→5 → X̃7→5, X̃7→4 → X̃7→4, X̃8→6 → X̃8→6, X̃8→3 → X̃8→3}

In[*]:= A = α - λ. {X̃# & /@ EdgeList[g, a- ↔ _ /; a ≠ VertexCount[g]]} T /. ps;
      "A="
      A // MatrixForm

Out[*]:= A=

Out[*]//MatrixForm=

$$\begin{pmatrix} 7 - 9 \tilde{x}_{1 \rightarrow 2} + 6 \tilde{x}_{1 \rightarrow 5} + 5 \tilde{x}_{2 \rightarrow 4} - 10 \tilde{x}_{3 \rightarrow 1} - 3 \tilde{x}_{3 \rightarrow 6} - 3 \tilde{x}_{3 \rightarrow 7} - 5 \tilde{x}_{4 \rightarrow 5} + 4 \tilde{x}_{5 \rightarrow 3} - 3 \tilde{x}_{6 \rightarrow 1} + 10 \tilde{x}_{6 \rightarrow 2} - 8 \tilde{x}_{7 \rightarrow 4} \\ -1 - 8 \tilde{x}_{1 \rightarrow 2} + 7 \tilde{x}_{1 \rightarrow 5} + 4 \tilde{x}_{2 \rightarrow 4} - 10 \tilde{x}_{3 \rightarrow 1} - 3 \tilde{x}_{3 \rightarrow 6} - 6 \tilde{x}_{3 \rightarrow 7} - 10 \tilde{x}_{4 \rightarrow 3} - 8 \tilde{x}_{4 \rightarrow 5} - \tilde{x}_{5 \rightarrow 3} - 5 \tilde{x}_{6 \rightarrow 1} - 2 \tilde{x}_{6 \rightarrow 2} - 1 \\ -4 + 9 \tilde{x}_{1 \rightarrow 2} + 4 \tilde{x}_{1 \rightarrow 5} - 3 \tilde{x}_{2 \rightarrow 4} + 6 \tilde{x}_{3 \rightarrow 7} + \tilde{x}_{4 \rightarrow 3} - 4 \tilde{x}_{4 \rightarrow 5} - 2 \tilde{x}_{5 \rightarrow 3} - 7 \tilde{x}_{6 \rightarrow 1} - \tilde{x}_{6 \rightarrow 2} + 4 \tilde{x}_{7 \rightarrow 4} + 7 \tilde{x}_7 \end{pmatrix}$$


```

```
In[*]:=  $\beta = A - \Delta c . \{x_{\#} \& /@ U_{Nb}[[U_{nc}]]\}^T;$ 
"β="
β // TableForm
```

```
Out[*]:= β=
```

```
Out[*]//TableForm=
```

$$\begin{aligned} & 7 - 3 x_{3 \rightarrow 6} - x_{4 \rightarrow 5} + 7 x_{6 \rightarrow 1} - 11 x_{7 \rightarrow 4} + 8 x_{7 \rightarrow 5} - 9 \tilde{x}_{1 \rightarrow 2} + 6 \tilde{x}_{1 \rightarrow 5} + 5 \tilde{x}_{2 \rightarrow 4} - 10 \tilde{x}_{3 \rightarrow 1} - 3 \tilde{x}_{3 \rightarrow 6} - 3 \tilde{x}_{3 \rightarrow 7} - 5 \tilde{x}_{4 \rightarrow 3} \\ & - 1 - 3 x_{3 \rightarrow 6} + x_{4 \rightarrow 5} + 5 x_{6 \rightarrow 1} - 26 x_{7 \rightarrow 4} - 8 x_{7 \rightarrow 5} - 8 \tilde{x}_{1 \rightarrow 2} + 7 \tilde{x}_{1 \rightarrow 5} + 4 \tilde{x}_{2 \rightarrow 4} - 10 \tilde{x}_{3 \rightarrow 1} - 3 \tilde{x}_{3 \rightarrow 6} - 6 \tilde{x}_{3 \rightarrow 7} - 10 \tilde{x}_{4 \rightarrow 3} \\ & - 4 - 7 x_{4 \rightarrow 5} - 7 x_{6 \rightarrow 1} + 11 x_{7 \rightarrow 4} + 11 x_{7 \rightarrow 5} + 9 \tilde{x}_{1 \rightarrow 2} + 4 \tilde{x}_{1 \rightarrow 5} - 3 \tilde{x}_{2 \rightarrow 4} + 6 \tilde{x}_{3 \rightarrow 7} + \tilde{x}_{4 \rightarrow 3} - 4 \tilde{x}_{4 \rightarrow 5} - 2 \tilde{x}_{5 \rightarrow 3} - 7 \tilde{x}_6 \end{aligned}$$

```
In[*]:= "решаем уравнение  $\Delta_c x_c = \beta$ :"
xc = LinearSolve[Δc, β]
```

```
Out[*]:= решаем уравнение  $\Delta_c x_c = \beta$ :
```

$$\begin{aligned} Out[*] = & \left\{ \left\{ \frac{1}{346} \left( -37 - 87 x_{3 \rightarrow 6} - 209 x_{4 \rightarrow 5} - 37 x_{6 \rightarrow 1} - 214 x_{7 \rightarrow 4} + \right. \right. \right. \\ & 322 x_{7 \rightarrow 5} + 24 \tilde{x}_{1 \rightarrow 2} + 309 \tilde{x}_{1 \rightarrow 5} + 40 \tilde{x}_{2 \rightarrow 4} - 290 \tilde{x}_{3 \rightarrow 1} - 87 \tilde{x}_{3 \rightarrow 6} + 48 \tilde{x}_{3 \rightarrow 7} - \\ & \left. \left. 120 \tilde{x}_{4 \rightarrow 3} - 310 \tilde{x}_{4 \rightarrow 5} - 19 \tilde{x}_{5 \rightarrow 3} - 327 \tilde{x}_{6 \rightarrow 1} + 80 \tilde{x}_{6 \rightarrow 2} - 142 \tilde{x}_{7 \rightarrow 4} + 293 \tilde{x}_{7 \rightarrow 5} \right) \right\}, \\ & \left\{ \frac{1}{692} \left( 1055 + 573 x_{3 \rightarrow 6} + 3369 x_{4 \rightarrow 5} + 2093 x_{6 \rightarrow 1} - 2146 x_{7 \rightarrow 4} - 5712 x_{7 \rightarrow 5} - 2592 \tilde{x}_{1 \rightarrow 2} - \right. \right. \\ & 3097 \tilde{x}_{1 \rightarrow 5} + 524 \tilde{x}_{2 \rightarrow 4} + 1910 \tilde{x}_{3 \rightarrow 1} + 573 \tilde{x}_{3 \rightarrow 6} - 2070 \tilde{x}_{3 \rightarrow 7} + 158 \tilde{x}_{4 \rightarrow 3} + \\ & \left. \left. 3032 \tilde{x}_{4 \rightarrow 5} + 495 \tilde{x}_{5 \rightarrow 3} + 4003 \tilde{x}_{6 \rightarrow 1} - 682 \tilde{x}_{6 \rightarrow 2} - 234 \tilde{x}_{7 \rightarrow 4} - 4137 \tilde{x}_{7 \rightarrow 5} \right) \right\}, \\ & \left\{ \frac{1}{346} \left( -233 - 99 x_{3 \rightarrow 6} - 381 x_{4 \rightarrow 5} - 233 x_{6 \rightarrow 1} - 160 x_{7 \rightarrow 4} + 438 x_{7 \rightarrow 5} + 254 \tilde{x}_{1 \rightarrow 2} + \right. \right. \\ & 459 \tilde{x}_{1 \rightarrow 5} - 38 \tilde{x}_{2 \rightarrow 4} - 330 \tilde{x}_{3 \rightarrow 1} - 99 \tilde{x}_{3 \rightarrow 6} + 162 \tilde{x}_{3 \rightarrow 7} - 232 \tilde{x}_{4 \rightarrow 3} - \\ & \left. \left. 484 \tilde{x}_{4 \rightarrow 5} - 129 \tilde{x}_{5 \rightarrow 3} - 563 \tilde{x}_{6 \rightarrow 1} - 76 \tilde{x}_{6 \rightarrow 2} - 90 \tilde{x}_{7 \rightarrow 4} + 405 \tilde{x}_{7 \rightarrow 5} \right) \right\} \end{aligned}$$

```
In[*]:= xcp = MapThread[x_{#1} → #2 &, {U_{Nb}[[U_c]], Flatten[xc]}]
```

$$\begin{aligned} Out[*] = & \left\{ x_{1 \rightarrow 2} \rightarrow \frac{1}{346} \left( -37 - 87 x_{3 \rightarrow 6} - 209 x_{4 \rightarrow 5} - 37 x_{6 \rightarrow 1} - 214 x_{7 \rightarrow 4} + \right. \right. \\ & 322 x_{7 \rightarrow 5} + 24 \tilde{x}_{1 \rightarrow 2} + 309 \tilde{x}_{1 \rightarrow 5} + 40 \tilde{x}_{2 \rightarrow 4} - 290 \tilde{x}_{3 \rightarrow 1} - 87 \tilde{x}_{3 \rightarrow 6} + 48 \tilde{x}_{3 \rightarrow 7} - \\ & \left. \left. 120 \tilde{x}_{4 \rightarrow 3} - 310 \tilde{x}_{4 \rightarrow 5} - 19 \tilde{x}_{5 \rightarrow 3} - 327 \tilde{x}_{6 \rightarrow 1} + 80 \tilde{x}_{6 \rightarrow 2} - 142 \tilde{x}_{7 \rightarrow 4} + 293 \tilde{x}_{7 \rightarrow 5} \right) \right\}, \\ & x_{1 \rightarrow 5} \rightarrow \frac{1}{692} \left( 1055 + 573 x_{3 \rightarrow 6} + 3369 x_{4 \rightarrow 5} + 2093 x_{6 \rightarrow 1} - 2146 x_{7 \rightarrow 4} - 5712 x_{7 \rightarrow 5} - \right. \\ & 2592 \tilde{x}_{1 \rightarrow 2} - 3097 \tilde{x}_{1 \rightarrow 5} + 524 \tilde{x}_{2 \rightarrow 4} + 1910 \tilde{x}_{3 \rightarrow 1} + 573 \tilde{x}_{3 \rightarrow 6} - 2070 \tilde{x}_{3 \rightarrow 7} + \\ & \left. \left. 158 \tilde{x}_{4 \rightarrow 3} + 3032 \tilde{x}_{4 \rightarrow 5} + 495 \tilde{x}_{5 \rightarrow 3} + 4003 \tilde{x}_{6 \rightarrow 1} - 682 \tilde{x}_{6 \rightarrow 2} - 234 \tilde{x}_{7 \rightarrow 4} - 4137 \tilde{x}_{7 \rightarrow 5} \right) \right\}, \\ & x_{2 \rightarrow 4} \rightarrow \frac{1}{346} \left( -233 - 99 x_{3 \rightarrow 6} - 381 x_{4 \rightarrow 5} - 233 x_{6 \rightarrow 1} - 160 x_{7 \rightarrow 4} + 438 x_{7 \rightarrow 5} + 254 \tilde{x}_{1 \rightarrow 2} + \right. \\ & 459 \tilde{x}_{1 \rightarrow 5} - 38 \tilde{x}_{2 \rightarrow 4} - 330 \tilde{x}_{3 \rightarrow 1} - 99 \tilde{x}_{3 \rightarrow 6} + 162 \tilde{x}_{3 \rightarrow 7} - 232 \tilde{x}_{4 \rightarrow 3} - \\ & \left. \left. 484 \tilde{x}_{4 \rightarrow 5} - 129 \tilde{x}_{5 \rightarrow 3} - 563 \tilde{x}_{6 \rightarrow 1} - 76 \tilde{x}_{6 \rightarrow 2} - 90 \tilde{x}_{7 \rightarrow 4} + 405 \tilde{x}_{7 \rightarrow 5} \right) \right\} \end{aligned}$$

In[ ]:= "общее решение:"

**xsol = ((s /. xcp) ~ Join ~ xcp);**

**xsol // TableForm**

Out[ ]:= общее решение:

Out[ ]//TableForm=

**Join[s,**

$$\left\{ x_{1 \leftrightarrow 2} \rightarrow \frac{1}{346} \left( -37 - 87 x_{3 \leftrightarrow 6} - 209 x_{4 \leftrightarrow 5} - 37 x_{6 \leftrightarrow 1} - 214 x_{7 \leftrightarrow 4} + 322 x_{7 \leftrightarrow 5} + 24 \tilde{x}_{1 \leftrightarrow 2} + 309 \tilde{x}_{1 \leftrightarrow 5} + 40 \tilde{x}_{2 \leftrightarrow 4} - 290 \tilde{x}_{3 \leftrightarrow 1} - 87 \tilde{x}_{3 \leftrightarrow 6} + 48 \tilde{x}_{3 \leftrightarrow 7} - 120 \tilde{x}_{4 \leftrightarrow 3} - 310 \tilde{x}_{4 \leftrightarrow 5} - 19 \tilde{x}_{5 \leftrightarrow 3} - 327 \tilde{x}_{6 \leftrightarrow 1} + 80 \tilde{x}_{6 \leftrightarrow 2} - 142 \tilde{x}_{7 \leftrightarrow 4} + 293 \tilde{x}_{7 \leftrightarrow 5} \right),$$

$$x_{1 \leftrightarrow 5} \rightarrow \frac{1}{692} \left( 1055 + 573 x_{3 \leftrightarrow 6} + 3369 x_{4 \leftrightarrow 5} + 2093 x_{6 \leftrightarrow 1} - 2146 x_{7 \leftrightarrow 4} - 5712 x_{7 \leftrightarrow 5} - 2592 \tilde{x}_{1 \leftrightarrow 2} - 3097 \tilde{x}_{1 \leftrightarrow 5} + 524 \tilde{x}_{2 \leftrightarrow 4} + 1910 \tilde{x}_{3 \leftrightarrow 1} + 573 \tilde{x}_{3 \leftrightarrow 6} - 2070 \tilde{x}_{3 \leftrightarrow 7} + 158 \tilde{x}_{4 \leftrightarrow 3} + 3032 \tilde{x}_{4 \leftrightarrow 5} + 495 \tilde{x}_{5 \leftrightarrow 3} + 4003 \tilde{x}_{6 \leftrightarrow 1} - 682 \tilde{x}_{6 \leftrightarrow 2} - 234 \tilde{x}_{7 \leftrightarrow 4} - 4137 \tilde{x}_{7 \leftrightarrow 5} \right),$$

$$x_{2 \leftrightarrow 4} \rightarrow \frac{1}{346} \left( -233 - 99 x_{3 \leftrightarrow 6} - 381 x_{4 \leftrightarrow 5} - 233 x_{6 \leftrightarrow 1} - 160 x_{7 \leftrightarrow 4} + 438 x_{7 \leftrightarrow 5} + 254 \tilde{x}_{1 \leftrightarrow 2} + 459 \tilde{x}_{1 \leftrightarrow 5} - 38 \tilde{x}_{2 \leftrightarrow 4} - 330 \tilde{x}_{3 \leftrightarrow 1} - 99 \tilde{x}_{3 \leftrightarrow 6} + 162 \tilde{x}_{3 \leftrightarrow 7} - 232 \tilde{x}_{4 \leftrightarrow 3} - 484 \tilde{x}_{4 \leftrightarrow 5} - 129 \tilde{x}_{5 \leftrightarrow 3} - 563 \tilde{x}_{6 \leftrightarrow 1} - 76 \tilde{x}_{6 \leftrightarrow 2} - 90 \tilde{x}_{7 \leftrightarrow 4} + 405 \tilde{x}_{7 \leftrightarrow 5} \right) \right]$$

In[ ]:= "eq test:"

**Simplify[dopEq /. s /. xcp]**