

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра дискретной математики и алгоритмики

РОМАНЧУК Марина Павловна

ЗАДАЧА МОДЕЛИРОВАНИЯ И ОЦЕНКИ ТРАНСПОРТНЫХ
ПОТОКОВ И ЕЕ ПРИЛОЖЕНИЯ

Магистерская диссертация
специальность 1-31 80 09 «Прикладная математика и информатика»

Научный руководитель
Пилипчук Людмила Андреевна
кандидат физ.-мат. наук, доцент

Допущена к защите
"___" _____ 2023 г.
Зав. кафедрой дискретной
математики и алгоритмики
_____ В.М. Котов
доктор физ.-мат. наук, профессор

Минск, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ	5
АГУЛЬНАЯ ХАРАКТЕРИСТИКА РАБОТЫ	6
SUMMARY	7
1 СВОЙСТВА БАЗИСА ПРОСТРАНСТВА РЕШЕНИЙ И ФУНДАМЕНТАЛЬНЫЕ ОПЕРАЦИИ С КОРНЕВЫМИ ДЕРЕВЬЯМИ	8
1.1 О свойствах фундаментальных (базисных) циклов	8
1.2 Математическая модель задачи о кратчайших путях из узла s в достижимые узлы графа	9
1.3 Фундаментальные операции с корневыми деревьями	10
1.4 Фундаментальные операции с ориентированным графом	13
1.5 Формирование списка дуг, входящих в узлы графа	14
1.6 Условия оптимальности опорного потока	15
1.7 Преобразование корневых деревьев	15
1.8 Основные свойства фундаментального разреза	17
1.9 Алгоритм построения множества дуг фундаментального разреза	19
2 ОЦЕНКА ПОТОКОВ НА НЕНАБЛЮДАЕМОЙ ЧАСТИ СЕТИ ПУТЕМ ИСПОЛЬЗОВАНИЯ РАЗРЕЖЕННЫХ СИСТЕМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ИДЕНТИФИКАЦИИ СЕНСОРНЫХ КОНФИГУРАЦИЙ УЗЛОВ	21
2.1 Идентификация сенсорных конфигураций узлов	21
2.2 Недоопределенная разреженная линейная система	22
2.3 Переопределенная разреженная линейная система	28
2.4 Оптимальное решение задачи оценки потока	32
3 ИССЛЕДОВАНИЕ ПРОЦЕССОВ ОЦЕНКИ НЕОДНОРОДНОГО ПОТОКА В МУЛЬТИСЕТЯХ	37
3.1 Моделирование процессов оценки мультипотока	37
3.2 Примеры локализации сенсоров в узлах мультиграфа	39
4 ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧИ	41
4.1 Генерация данных и решения задачи оценки потоков на ненаблюдаемой части сети	41
4.2 Сравнение матричного метода и метода декомпозиции для решения задачи оценки потоков в сетях	41
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	45
ПРИЛОЖЕНИЕ А	47
ПРИЛОЖЕНИЕ Б	49
ПРИЛОЖЕНИЕ В	57

ВВЕДЕНИЕ

Трудно представить жизнь в современном обществе без интенсивных транспортных потоков, которые, с одной стороны, обеспечивают экономическую стабильность, с другой стороны - отрицательно сказываются на экологической ситуации в мире. Разработка путей управления потоками позволит обеспечить баланс между их пользой и негативным влиянием.

Для того чтобы грамотно управлять потоками, необходимо спроектировать стратегию расположения специальных программируемых устройств (сенсоров). В современном мире разработаны различные технологии, которые дают возможность благодаря установке сенсоров и видеокамер отслеживать небольшие участки дорог, что в целом позволяет контролировать потоки в больших сетях. В то же время одной из задач является минимизация числа сенсоров в узлах сети для сбора частичной информации о функции потока. Отследить все участки сети не представляется возможным, поэтому анализируются данные отдельных отслеживаемых частей сети и на основе полученных результатов дается оценка транспортного потока на ненаблюдаемой части. Для моделирования процессов оценки потоков на ненаблюдаемой части можно использовать недоопределенную систему линейных алгебраических уравнений, в которых неизвестные дуговые и внешние потоки представлены переменными, а условия баланса образуют совокупность уравнений. Численные значения потоков на ненаблюдаемой части исходной сети можно найти путем решения такой системы.

Особые сложности вызывает моделирование процесса транспортного потока в мегаполисах из-за интенсивного использования транспортных средств (может достигать нескольких десятков тысяч), следовательно, величины дугового и внешнего потока обычно лежат в широком диапазоне. Очевидна неэффективность использования стратегий полного перебора узлов для решения задачи минимизации количества элементов множества обозреваемых узлов, потому что они требуют значительных вычислительных затрат. Для оптимизации управления транспортными потоками в больших городах видится целесообразным построение алгоритмических, структурных и технологических подсистем с матрицами полного или неполного рангов с различными типами разреженности в синтезе с современными достижениями в области инновационных технологий разреженного матричного и сетевого анализа, алгоритмической теории графов, теоретической информатики.

В работе рассматриваются фундаментальные операции с корневыми деревьями, алгоритм построения фундаментального разреза, исследуется возможность применения метода декомпозиции, позволяющего разделить крупную задачу на несколько задач меньших размерностей и применить

новейшие программные инструменты для построения численных решений разреженных систем линейных алгебраических уравнений.

Результатом исследований является разработанное на языке R программное обеспечение, позволяющее находить оптимальное решение в задаче расположения сенсоров.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 62 с., 23 рис., 12 таблиц, 3 приложения, 19 источников.

Ключевые слова: РАЗРЕЖЕННАЯ СИСТЕМА ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ, ФУНДАМЕНТАЛЬНЫЙ ЦИКЛ, КОРНЕВОЕ ДЕРЕВО, ФУНДАМЕНТАЛЬНЫЙ РАЗРЕЗ, ОПОРНЫЙ ПОТОК, ХАРАКТЕРИСТИЧЕСКИЕ ВЕКТОРЫ, ДЕКОМПОЗИЦИЯ, СЕНСОР, ОБОЗРЕВАЕМЫЙ УЗЕЛ, МУЛЬТИПОТОК, МУЛЬТИГРАФ.

Объект исследования – модели и методы оценки потоков, алгоритмы и технологии решения разреженных систем линейных алгебраических уравнений.

Цель работы – разработка и применение эффективных алгоритмов и технологий в задачах математического моделирования процессов оценки потоков, решение задач оптимального расположения сенсоров для оценки транспортных потоков в сетях.

Методы исследования – методы разреженного численного анализа, линейной алгебры, аппарата теории графов, теории алгоритмов, теории потоков, методы оптимизации.

В результате приведен пример построения оптимального решения задачи оценки потока на ненаблюдаемой части сети с минимальной мощностью множества обозреваемых узлов. Смоделирован процесс оценки мультипотока и рассмотрен пример локализации специальных программируемых устройств в узлах мультиграфа. На языке программирования R реализован алгоритм декомпозиции для решения задачи оценки транспортных потоков.

Область применения – решение разреженных систем линейных алгебраических уравнений больших размерностей методом декомпозиции, решение линейных неоднородных задач потокового программирования, решение задачи оценки транспортных потоков.

АГУЛЬНАЯ ХАРАКТЭРЫСТЫКА РАБОТЫ

Магістарская дысертацыя, 62 с., 23 мал., 12 табліц, 3 дадаткі, 19 крыніц.

Ключавыя словы: РАЗРЭДЖАНАЯ СІСТЭМА ЛІНЕЙНЫХ АЛГЕБРАІЧНЫХ УРАЎНЕННЯЎ, ФУНДАМЕНТАЛЬНЫ ЦЫКЛ, КАРНЯВОЕ ДРЭВА, ФУНДАМЕНТАЛЬНЫ РАЗРЭЗ, АПОРНЫ ПАТОК, ХАРАКТЭРЫСТЫЧНЫЯ ВЕКТАРЫ, ДЭКАМПАЗІЦЫЯ, СЭНСАР, МУЛЬЦПАТОК, МУЛЬЦІГРАФ.

Аб’ект даследавання – мадэлі і метады ацэнкі патокаў, алгарытмы і тэхналогіі рашэння разрэджаных сістэм лінейных алгебраічных ураўненняў.

Мэта работы – распрацоўка і прымяненне эфектыўных алгарытмаў і тэхналогій у задачах матэматычнага мадэліравання працэсаў ацэнкі патокаў, рашэнне задач аптымальнага размяшчэння сэнсараў для ацэнкі транспартных патокаў у сетках.

Метады даследавання – метады разрэджанага лікавага аналізу, лінейнай алгебры, апарата тэорыі графаў, тэорыі алгарытмаў, тэорыі патокаў, метады аптымізацыі.

У выніку прыведзены прыклад пабудовы аптымальнага рашэння задачы ацэнкі патокаў на неназіральнай частцы сеткі з мінімальнай колькасцю элементаў мноства аглядаемых вузлаў. Змадэліраваны працэс ацэнкі мультыпатоку і разгледжан прыклад лакалізацыі спецыяльных праграмаваных устройстваў у вузлах мультыграфу. На мове праграмавання R рэалізаваны алгарытм дэкампазіцыі для рашэння задачы ацэнкі транспартных патокаў.

Галіна прымянення – рашэнне разрэджаных сістэм лінейных алгебраічных ураўненняў вялікіх размернасцей метадам дэкампазіцыі, рашэнне лінейных неаднародных задач патокавага праграмавання, рашэнне задачы ацэнкі транспартных патокаў.

SUMMARY

Master thesis, 62 p., 23 img., 12 tables, 3 appendices, 19 sources.

Keywords: SPARSE SYSTEM OF LINEAR ALGEBRAIC EQUATIONS, FUNDAMENTAL CYCLE, ROOT TREE, FUNDAMENTAL CUT, REFERENCE FLOW, CHARACTERISTIC VECTORS, DECOMPOSITION, SENSOR, OBSERVABLE NODE, MULTIFLOW, MULTIGRAPH.

The object of study – models and methods for estimating flows, algorithms and techniques for solving sparse systems of linear algebraic equations.

The purpose of the work – development and application of effective algorithms and technologies in mathematical modeling of flow estimation processes, solving the problems of optimal location of sensors to estimate traffic flows in networks.

The research methods – methods of sparse numerical analysis, linear algebra, graph theory, theory of algorithms, flow theory, optimization methods.

As a result – an example of constructing an optimal solution to the flow estimation problem on the unobservable part of the network with the minimum power of the set with observable nodes. The process of multiflow estimation is simulated and an example of localization of special programmable devices in the nodes of the multigraph is considered. A decomposition algorithm for solving the problem of estimating traffic flows is implemented in the programming language R.

Scope of application – solution of sparse systems of linear algebraic equations of large dimensions by decomposition method, solving linear inhomogeneous problems of stream programming, solving the problem of estimating traffic flows.

ГЛАВА 1

СВОЙСТВА БАЗИСА ПРОСТРАНСТВА РЕШЕНИЙ И ФУНДАМЕНТАЛЬНЫЕ ОПЕРАЦИИ С КОРНЕВЫМИ ДЕРЕВЬЯМИ

1.1 О свойствах фундаментальных (базисных) циклов

Для определения свойств фундаментального цикла рассмотрим связный ориентированный граф $G = (V, E)$. Пусть $\bar{G} = (V, E_0)$ – его произвольное покрывающее дерево (остов). Приведем определение фундаментального цикла.

Определение 1.1 *Добавление любой дуги $(\tau, \rho) \in E_H = E \setminus E_0$ к покрывающему дереву $\bar{G} = (V, E_0)$ создает ровно один цикл $C(\tau, \rho)$, который называется фундаментальным (базисным) циклом в графе G , порожденным дугой (τ, ρ) , относительно исходного покрывающего дерева.*

Исходя из того, что $|E_H| = |E| - |E_0| = m - n + 1$, где числа m, n обозначают мощности множеств E и V соответственно, а число дуг дерева $|E_0| = n - 1$, получаем, что граф G содержит $\nu = m - n + 1$ фундаментальных (базисных) циклов и изменение покрывающего дерева не меняет их число. При условии удаления из базисного цикла $C(\tau, \rho)$ любой дуги (i, j) мы получаем некоторое покрывающее дерево.

Определение 1.2 *Число $\nu = \nu(G) = m - n + 1$ называется цикломатическим числом графа G .*

Для графа G , состоящего из p связных компонент, цикломатическое равно $\nu = \nu(G) = m - n + p$.

Определение 1.3 *Число $\rho = \rho(G) = |V| - 1$ называется рангом графа G .*

Для графа G , состоящего из p связных компонент, ранг равен $\rho = \rho(G) = |V| - p = n - p$. Таким образом, для любого графа G выполняется равенство $\nu(G) + \rho(G) = |E|$.

Будем считать дугу $(\tau, \rho) \in E_H$ прямой, исходя из этого введем направление обхода в каждом базисном цикле $C(\tau, \rho)$, порожденном дугой (τ, ρ) . Введем множества $C^+(\tau, \rho)$, содержащее прямые дуги цикла (совпадают по направлению с дугой (τ, ρ)), и $C^-(\tau, \rho)$, содержащее обратные дуги цикла (не совпадают с направлением дуги (τ, ρ)).

1.2 Математическая модель задачи о кратчайших путях из узла s в достижимые узлы графа

Для каждой дуги (i, j) конечного ориентированного графа $G = (V, E)$ установим стоимости $c_{i,j}$ перемещения по дуге. Для расчета значения длины пути $L_{s,t} = \{v_0, v_1, \dots, v_k\}$ из узла $s = v_0 \in V$ в узел $t = v_k \in V$ будем использовать формулу

$$c(L_{s,t}) \stackrel{def}{=} \sum_{i=1}^k c_{v_{i-1}, v_i}. \quad (1.1)$$

Предположим, что наиболее коротким путем из узла s в узел t является путь $L_{s,t} = \{v_0, v_1, \dots, v_k\}$, где $s = v_0, t = v_k$, тогда для любого значения $i : 0 < i < k$ путь $L_{s,v_i} = \{v_0, \dots, v_i\}$ также будет иметь наименьшую длину.

Задача о кратчайших путях из узла $s \in V$ в узлы $i \in V \setminus \{s\}$ графа G может быть представлена следующим образом:

$$\sum_{(i,j) \in E} c_{i,j} x_{i,j} \rightarrow \min, \quad (1.2)$$

$$\sum_{j \in V_i^+(E)} x_{i,j} - \sum_{j \in V_i^-(E)} x_{j,i} = \begin{cases} n - 1, & i = s, \\ -1, & i \in V \setminus \{s\}, \end{cases} \quad (1.3)$$

где $c_{i,j}$ – стоимость перемещения по дуге (i, j) единицы потока $x_{i,j}$ из узла i в узел j .

$$0 \leq x_{i,j} \leq n, (i, j) \in E, x_{i,j} \in \mathbb{N}, \quad (1.4)$$

где $V_i^+(E) = \{j \in V : (i, j) \in E\}$, $V_i^-(E) = \{j \in V : (j, i) \in E\}$, $x_{i,j}$ – величина дугового потока для дуги (i, j) .

Дуговой поток $x_{i,j}$ не превышает $n - 1$ единиц потока для любой дуги. Ограничения принимают вид:

$$x_{i,j} \geq 0, (i, j) \in E, x_{i,j} \in \mathbb{N}. \quad (1.5)$$

Определение 1.4 Прямоугольная матрица называется абсолютно унимодулярной, если все ее миноры принимают значения из множества чисел $\{-1, 0, +1\}$.

Из абсолютной унимодулярности матрицы инцидентности графа G следует, что оптимальный поток x имеет целочисленное значение.

Математическая модель задачи о кратчайшем пути представлена в виде (1.2) - (1.3), (1.5). Задача состоит в минимизации стоимости $\sum_{(i,j) \in E} c_{ij} x_{ij}$ потока величины $n - 1$, исходящего из узла $s \in V$ в узел $\forall i \in V \setminus \{s\}$, где n – число узлов графа G , достижимых из узла s (достижимы все узлы графа).

1.3 Фундаментальные операции с корневыми деревьями

Определение 1.5 Поток $x = (x_{i,j}, (i,j) \in E)$ называется допустимым решением (*feasible solution*) задачи (1.2) – (1.3), (1.5), если выполняются ограничения (1.3), (1.5).

Выберем начальный допустимый поток $x = (x_{i,j}, (i,j) \in E)$. Пусть покрывающее дерево $G_0 = (V, E_0)$ включает все узлы множества V графа G , тогда пару $\{x, G_0\}$ условно назовем опорным потоком.

Определение 1.6 Множество линейно независимых векторов, каждый из которых является решением однородной системы, порожденной из системы (1.3), называется базисом пространства решений.

Эффективные алгоритмы построения характеристических векторов для разреженной системы с матрицей инцидентности графа представлены в [1, 2].

Характеристические векторы $\delta(\tau, \rho) = (\delta_{i,j}^{\tau, \rho}, (i,j) \in E)$, порожденные дугами $(\tau, \rho) \in E \setminus E_0$, составляют базис пространства решений однородной системы, порожденной из системы (1.3).

Опишем структуры данных, необходимые для фундаментальных операций над элементами корневых деревьев. [6]

Предположим, что существует единственный путь из узла $s \in V$ в любой узел $i \in V \setminus \{s\}$. Построим биективное отображение номеров узлов $i \in V \setminus \{s\}$ на соответствующие элементы $\{pred(i), \forall i \in V \setminus \{s\}\}$, где $pred(i)$ обозначает предка для узла i в графе G , $pred(s) = 0$. Полученное дерево $G_0 = (V, E_0)$ называется корневым деревом графа G с корнем в узле s .

Обозначим через $pred[i]$ элемент биективного отображения узла i в корневом дереве и составим список $\{pred[i]\}$ для всех узлов i из множества V . Если $pred[i] = 0$, то узел i является корнем дерева.

Обозначим $d[i]$ – направление дуги в корневом дереве G_0 . Значения могут быть вычислены следующим образом:

$$d[i] = \begin{cases} 0, & i = s, \\ 1, & (pred[i], i) \in E, \\ -1, & (i, pred[i]) \in E, \end{cases} \quad (1.6)$$

где s – корень дерева. Сформируем список направлений $\{d[i]\}$ для всех узлов i множества V .

Пусть уровень узла i (т.е. длина в дугах от узла i до корня в корневом дереве) обозначается $depth[i]$. Составим список значений для всех узлов i множества V .

С целью определения элемента династического обхода корневого дерева составим список $\{thread[i], i \in V\}$ для всех узлов i множества V . Для

нахождения элементов династического обхода с корнем в узле i прибегнем к построению последовательности узлов поддереза

$$i, thread[i], thread[thread[i]], \dots, thread[j].$$

Будем идти по описанной последовательности, пока соблюдается условие

$$depth[thread[j]] > depth[i].$$

Уровень каждого предшествующего узла не превышает уровень каждого следующего узла построенной последовательности $i, thread[i], \dots, thread[j]$.

Теорема 1.1 Пусть задан список $\{pred[i], i \in V\}$ и выполняется следующее соотношение:

$$pred[i] = \begin{cases} 0, & i = s, \\ |pred[i]|, & i \in V \setminus \{s\}, \end{cases} \quad (1.7)$$

где $|pred[i]|$ – элемент биективного отображения узла i корневого дерева, а s – узел, в котором находится корень заданного корневого дерева $G_0 = (V, G_0)$. Корневое дерево G_0 с корнем в узле s совпадает с деревом достижимости узлов графа G из узла s . Тогда для любого узла $i \in V \setminus \{s\}$ величина потока $x_{pred[i], i}$ дуги $(pred[i], i) \in G_0$ равняется количеству узлов поддереза с корнем в узле V .

Пусть вектор $x = (x_{i,j}, x_{i,j} \in E)$ – начальное допустимое решение задачи (1.2) - (1.3), (1.5). Значения $x_{i,j}$ для всех дуг множества $E \setminus E_0$ будем считать нулевыми.

Рассмотрим теорему о нахождении узлов поддереза с корнем в узле i .

Теорема 1.2 Величина дугового потока $x_{pred[i]}$ дуги $(pred[i], i) \in E_0$ совпадает с количеством узлов поддереза с корнем в узле i . [1]

Доказательство теоремы о нахождении узлов поддереза с корнем в узле i приведено в [1]. Оценка числа операций равна $O(k)$, где k – число узлов поддереза.

Приведем алгоритм 1.1 процедуры определения единственного пути в корневом дереве из узла i в корень.

Число операций равно $O(k)$, где k - число узлов пути, соединяющего узел i с корнем дерева.

Алгоритм 1.1 Псевдокод процедуры определения узлов единственного пути в корневом дереве от узла i к корню.

- 1: **while** $i \neq root$ **do**
 - 2: $i \leftarrow pred[i]$
 - 3: **end while**
-

Приведем алгоритм 1.2 процедуры идентификации цикла, порожденного дугой (k, l) и дугами дерева.

Число операций составляет $O(k)$, где k - число узлов цикла, образованного добавлением дуги (k, l) к списку дуг дерева.

Алгоритм 1.2 Псевдокод процедуры идентификации цикла, порожденного дугой (i, j) и дугами дерева.

```
1: while  $i \neq j$  do
2:   if  $depth[i] > depth[j]$  then
3:      $i \leftarrow pred[i]$ 
4:   else
5:     if  $depth[j] == depth[i]$  then
6:        $i \leftarrow pred[i]$ 
7:     end if
8:      $j \leftarrow pred[j]$ 
9:   end if
10: end while
```

Приведем алгоритм 1.3 нахождения листьев дерева. Для определения, является ли узел j листом дерева, необходимо сравнить глубину узла следующего по династическому обходу $thread[j]$.

Сложность алгоритма равна $O(n)$, где n - число узлов дерева.

Алгоритм 1.3 Псевдокод процедуры нахождения листьев дерева с использованием структур $\{thread[i], \forall i \in V\}$ и $\{depth[i], \forall i \in V\}$

```
1:  $i \leftarrow thread[root]$ 
2: while  $i \neq root$  do
3:   if  $depth[i] \geq depth[thread[i]]$  then
4:      $add(list, i)$ 
5:   end if
6:    $i \leftarrow thread[i]$ 
7: end while
8: return  $list$ 
```

Приведем алгоритм 1.4 нахождения листьев дерева, основанный на анализе массива предков узлов дерева. На каждом шаге династического обхода определяется, является ли текущий узел предком узла, следующего по династии.

Сложность данного алгоритма аналогично предыдущему и равна $O(n)$, где n - число узлов дерева.

Алгоритм 1.4 Псевдокод процедуры нахождения листьев дерева с использованием структур $\{thread[i], \forall i \in V\}$ и $\{depth[i], \forall i \in V\}$

```

1:  $i \leftarrow thread[root]$ 
2: while  $i \neq root$  do
3:   if  $i \neq pred[thread[i]]$  then
4:      $add(list, i)$ 
5:   end if
6:    $i \leftarrow thread[i]$ 
7: end while
8: return  $list$ 

```

1.4 Фундаментальные операции с ориентированным графом

Рассмотрим граф G с числом узлов $|V| = 7$ и списком ребер $E = \{(1, 3), (2, 5), (2, 6), (3, 4), (4, 2), (5, 1), (5, 4), (5, 6), (6, 1), (6, 4), (7, 3), (7, 5)\}$ (см. рисунок 1.1).

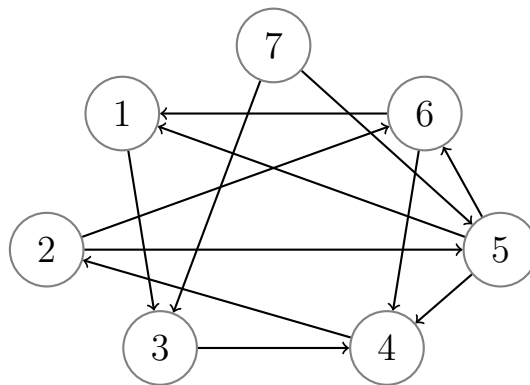


Рисунок 1.1 – Орграф $G = (V, E)$

Приведем список наследников [9] $P_o[i], i = 1, \dots, |V| + 1$. В таблице 1.1 граф представлен списками $o[k]$.

Таблица 1.1 – Представление графа G списками $o[k], k = 1, \dots, |E|$ и $t[k], k = 1, \dots, |E|$, отсортированными по неубыванию списка $o[k], k = 1, \dots, |E|$

k	1	2	3	4	5	6	7	8	9	10	11	12
$o[k]$	1	2	2	3	4	5	5	5	6	6	7	7
$t[k]$	3	5	6	4	2	1	4	6	1	4	3	5

Рассмотрим орграф $G = (V, E)$ с пронумерованными дугами, отсортированными по неубыванию списка $o[k], k = 1, \dots, |E|$ (см. рисунок 1.2).

Количество дуг, исходящих из узла i , равно

$$t[i] = P_o[i + 1] - P_o[i], i \in V.$$

Таблица 1.2 – Представление списка $P_o[i], i \in V$ для нахождения дуг, исходящих из узла i

i	1	2	3	4	5	6	7	8
$P_o[i]$	1	2	4	5	6	9	11	13

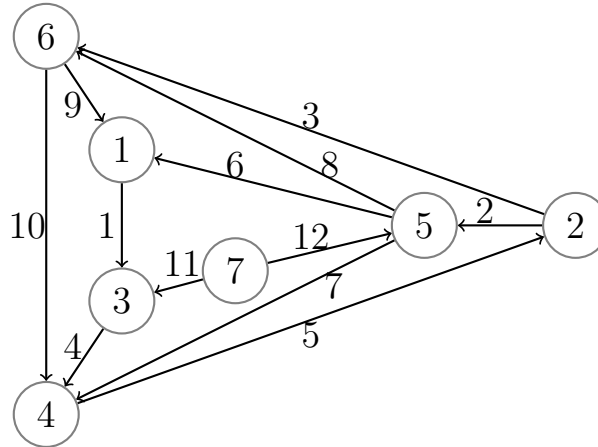


Рисунок 1.2 – Орграф $G = (V, E)$ с пронумерованными дугами, отсортированными по неубыванию списка $o[k], k = 1, \dots, |E|$

Таким образом, число операций в наихудшем случае равно $O(1)$ (вычисление количества дуг, исходящих из узла i , осуществляется согласно [9]).

1.5 Формирование списка дуг, входящих в узлы графа

В таблице 1.3 представлен список номеров дуг, входящих в узлы графа G .

Таблица 1.3 – Представление списка $L_t[k']$ номеров дуг, входящих в узлы графа G

k'	1	2	3	4	5	6	7	8	9	10	11	12
$L_t[k']$	6	9	5	1	11	4	7	10	2	12	3	8
$t[L_t[k']]$	1	1	2	3	3	4	4	4	5	5	6	6

Номера дуг, входящих в узел i , определяются следующим образом:

$$L_t[P_t[i] + j], j = 0, \dots, (P_t[i + 1] - P_t[i] - 1).$$

Если $P_t[i + 1] - P_t[i] - 1 = 0$, то для узла i нет входящих дуг.

Пусть $i = 6$, $P_t[i] = P_t[6] = 11$. Количество дуг $\tilde{t}[i]$, входящих в узел $i = 6$, равно $\tilde{t}[i] = \tilde{t}[6] = P_t[i + 1] - P_t[i] = P_t[7] - P_t[6] = 13 - 11 = 2$. Номера дуг, входящих в узел $i = 6$, равны $L_t[P_t[i] + j] = L_t[P_t[6] + j]$, $j = 0, \dots, (P_t[i + 1] - P_t[i] - 1) = 0, \dots, (P_t[7] - P_t[6] - 1) = 0, \dots, 1$.

Пусть $i = 7$, $P_t[8] - P_t[7] = 13 - 13 = 0$. Это означает, что узел $i = 7$ не содержит входящих дуг.

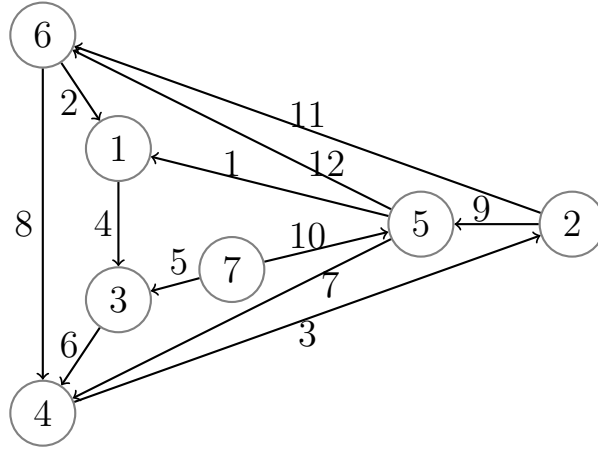


Рисунок 1.3 – Орграф $G = (V, E)$ с пронумерованными дугами, отсортированными по неубыванию списка $t[k], k = 1, \dots, |E|$. Сортировка дуг по неубыванию списка $o[k]$ нарушена

1.6 Условия оптимальности опорного потока

В соответствии с теоремой 1.2 поток считается невырожденным $\{x, E_0\}$ при выполнении условий $x_i, i > 0$ и $x_j, j < n, n = |V|, (i, j) \in E_0$, для потоков дуг корневого дерева. При этом экстремальная задача (1.2), (1.3) - (1.5) будем иметь целочисленные допустимые решения $x = (x_{i,j}, (i, j) \in E)$. Следовательно, опорный поток $\{x, E_0\}$ является невырожденным.

Подсчитаем значение $\Delta_{\tau, \rho}$ – оценку дуги $((\tau, \rho) \in E \setminus E_0)$.

$$\Delta_{\tau, \rho} = c_{\tau, \rho} + \sum_{(i, j) \in E_0} c_{i, j} \delta_{i, j}^{\tau, \rho} = \sum_{(i, j) \in L(\tau, \rho)} c_{i, j} \delta_{i, j}^{\tau, \rho}, (\tau, \rho) \in E \setminus E_0, \quad (1.8)$$

где $(\tau, \rho) \in E \setminus E_0, \delta(\tau, \rho) = (\delta_{i, j}^{\tau, \rho}, (i, j) \in E)$ – вектор базиса пространства решений однородной системы, порожденной из системы (1.3).

Теорема 1.3 Опорный поток $\{x, G_0\}$ является оптимальным тогда и только тогда, когда $\forall (\tau, \rho) \in E \setminus E_0$ выполняются условия

$$\begin{aligned} x_{\tau, \rho} = 0 &\Rightarrow \Delta_{\tau, \rho} > 0 \\ x_{\tau, \rho} > 0 &\Rightarrow \Delta_{\tau, \rho} = 0. \end{aligned} \quad (1.9)$$

Добавление некоторой дуги (τ, ρ) приведет к получению фундаментального цикла графа.

Рассмотрим псевдокод для идентификации базисного цикла $L(\tau, \rho)$, порожденного дугой (τ, ρ) (см. алгоритм 1.5). Число операций алгоритма в наихудшем случае равно $O(k)$, где k - число узлов цикла, образованного дугой (τ, ρ) .

1.7 Преобразование корневых деревьев

Пусть $\overline{G_0} = (V, \overline{E_0})$ – корневое дерево, содержащее дугу $(i_0, j_0) \in E \setminus E_0$, для которой нарушены условия оптимальности. При условии, что дуга $(i_1, j_1) \in$

Алгоритм 1.5 Псевдокод процедуры идентификации фундаментального цикла $L(\tau, \rho)$, порожденного дугой (τ, ρ) , и вычисление оценки $\Delta_{\tau, \rho}$

```

1:  $\Delta_{\tau, \rho} = 0$ ;
2:  $d = \text{depth}[\tau] - \text{depth}[\rho]$ ;
3: while  $\tau \neq \rho$  do
4:   if  $d > 0$  then
5:      $\Delta_{\tau, \rho} = \Delta_{\tau, \rho} + c_{\text{pred}[\tau], \tau} * \delta_{\text{pred}[\tau], \tau}$ ;
6:      $\tau = \text{pred}[\tau]$ ;  $d = d - 1$ ;
7:   else
8:     if  $\text{diff} == 0$  then
9:        $\Delta_{\tau, \rho} = \Delta_{\tau, \rho} + c_{\text{pred}[\tau], \tau} * \delta_{\text{pred}[\tau], \tau}$ ;
10:       $\Delta_{\tau, \rho} = \Delta_{\tau, \rho} + c_{\text{pred}[\rho], \rho} * \delta_{\text{pred}[\rho], \rho}$ ;
11:       $\tau = \text{pred}[\tau]$ ;  $\rho = \text{pred}[\rho]$ ;
12:     else
13:        $\Delta_{\tau, \rho} = \Delta_{\tau, \rho} + c_{\text{pred}[\rho], \rho} * \delta_{\text{pred}[\rho], \rho}$ ;
14:        $\rho = \text{pred}[\rho]$ ;  $d = d + 1$ ;
15:     end if
16:   end if
17: end while

```

$\in E_0$ является выводимой, множество дуг корневого дерева $\overline{G_0}$ равно $\overline{E_0} = E_0 \cup (i_0, j_0) \setminus (i_1, j_1)$.

Теорема 1.4 Дуга, выводимая из корневого дерева G_0 , определяется по формуле

$$(i_1, j_1) = (\text{pred}[j_0], j_0).$$

Для дуг пути $i_1, \text{pred}[i_1], \dots, s$, где s – корень дерева, дуговые потоки уменьшатся на величину $y = x_{\text{pred}[j_0], j_0}$, а значение дуговых потоков на пути $i_0, \text{pred}[i_0], \dots, s$ увеличатся на эту величину. Новый дуговой поток дуги (i_1, j_1) равен нулю и $\text{pred}[j_0] = i_0$.

Рассмотрим процедуру получения корневого дерева $\overline{G_0} = (V, \overline{E_0})$, где $\overline{E_0} = E_0 \cup (i_0, j_0) \setminus (i_1, j_1) = E_0 \cup (i_0, j_0) \setminus (\text{pred}[j_0], j_0)$, из корневого дерева $G_0 = (V, E_0)$. Добавление дуги (i_0, j_0) порождает базисный цикл $L(i_0, j_0)$. Пропустим максимально допустимый поток через цикл $L(i_0, j_0)$. Из теоремы 1.4 следует, что на дуге $(\text{pred}[j_0], j_0) \in L(i_0, j_0)$ достигается нижняя граница дугового потока, т.е. $\overline{x_{\text{pred}[j_0], j_0}} = 0$. Верхняя граница дугового потока не может быть достигнута. Удалим дугу $(i_1, j_1) = (\text{pred}[j_0], j_0)$ из корневого дерева G_0 .

Для корректного перехода к новому корневному дереву $\overline{G_0}$ необходимо изменить некоторые значения структур представления корневого дерева, среди которых списки $\{\text{pred}[i]\}$, $\{\text{depth}[i]\}$, $\{\text{thread}[i]\}$, $i \in V$.

Построим поддерево G_0 с корнем в узле $\text{pred}[j_0] : j, \text{thread}[j], \dots, t$, где $j = \text{pred}[j_0]$. Далее найдем узел $k = j_0$ такой, что $\text{depth}[\text{thread}[j]] \leq \text{depth}[k]$,

и выполним корректировку списка $\{pred[i], i \in V\}$: $pred[j_0] = i_0$. Изменим список $\{thread[i], i \in V\}$ с учетом следующих равенств: $thread[t] = thread[k]$, $thread[k] = j_0$. Для изменения списка $\{depth[i], i \in V\}$ используем условие $l = depth[i_0] - depth[k] + 1$, где l – значение, на которое изменяется глубина.

Пусть $p_{i,j}$ – надежность дуги, определяемая как вероятность того, что $(i, j) \in E$.

Определение 1.7 Величина $R(W) = \prod_{(i,j) \in W} p_{i,j}$ называется надежностью пути W из узла s в некоторый достижимый узел.

В постановке задачи о пути максимальной надежности графа производится поиск пути максимальной надежности для любой вершины i из множества $V \setminus \{s\}$, где $s \in V$.

Путь максимальной надежности от узла s в любой достижимый узел совпадает с кратчайшим путем из узла s в этот узел [1].

1.8 Основные свойства фундаментального разреза

Рассмотрим граф $G = (V, E)$. Пусть $V' \subset V$, $\bar{V}' = V \setminus V'$. Введем множество прямых и обратных дуг разреза.

Определение 1.8 Прямыми дугами разреза будем называть дуги, входящие в множество $CC^+(V') = \{(i, j) \in E \mid i \in V', j \in \bar{V}'\}$, а дуги множества $CC^-(V') = \{(i, j) \in E \mid i \in \bar{V}', j \in V'\}$ – обратными дугами разреза.

Определение 1.9 Объединение множества $CC^+(V')$ прямых дуг и множества $CC^-(V')$ обратных дуг разреза определяется как $CC(V') = CC^+(V') \cup (-CC^-(V'))$ и называется разрезом графа G .

Из определения следует, что $CC^-(V') = CC^+(\bar{V}')$ и $CC(V') = -CC(\bar{V}')$.

Определение 1.10 Если граф $\tilde{G} = (V, E \setminus CC(V'))$, полученный из графа G в результате удаления дуг разреза $CC(V')$, содержит ровно две компоненты связности, то разрез $CC(V')$ называется простым.

Введем множества узлов $V^+(V')$ и $V^-(V')$ следующим образом:

$$\begin{aligned} V^+(V') &= \{j \in V \setminus V' : (i, j) \in CC^+(V')\} \\ V^-(V') &= \{i \in V \setminus V' : (i, j) \in CC^-(V')\}. \end{aligned}$$

Определение 1.11 Числа $|CC^+_i|$, $|CC^-_i|$ называются исходной и входной валентностью узла $i \in V$ соответственно.

Обозначим через $\tilde{\delta}(CC(V')) = (\tilde{\delta}_{i,j}(CC(V')), (i, j) \in E)$ характеристический вектор разреза $CC(V')$ (для краткости будем использовать обозначение $\tilde{\delta}(CC)$). Его координаты равны

$$\tilde{\delta}_{i,j}(CC) = \begin{cases} 0, & (i, j) \notin CC, \\ 1, & (i, j) \in CC^+, \\ -1, & (i, j) \in CC^-. \end{cases}$$

Пусть $(\tau, \rho) \in E_0$ – некоторая дуга покрывающего дерева $\bar{G} = (V, E_0)$, ее удаление приведет к получению графа, содержащего ровно две компоненты связности, являющиеся деревьями. Выберем одну из полученных компонент связности и обозначим множество ее узлов как V' , построим на этом множестве разрез $CC(V')$ графа. Множество дуг построенного разреза содержит порождающую дугу (τ, ρ) , лежащую в множестве дуг покрывающего дерева \bar{G} и дуги, принадлежащие множеству $E \setminus E_0$.

Назовем такой разрез фундаментальным, порожденным дугой $(\tau, \rho) \in E_0$ относительно выбранного покрывающего дерева $\bar{G} = (V, E_0)$, обозначим его как $CC(\tau, \rho)$. Порождающую дугу базисного разреза $CC(\tau, \rho)$ будем всегда считать прямой, исходя из этого можем сформировать множества прямых и обратных дуг разреза $CC^+(\tau, \rho)$, $CC^-(\tau, \rho)$.

Мощность множества E_0 дуг покрывающего дерева \bar{G} совпадает с числом базисных разрезов относительно покрывающего дерева \bar{G} . Это число не зависит от выбранного покрывающего дерева \bar{G} и равно $|V| - 1$, где $|V|$ – количество вершин графа G .

Обозначим через $\tilde{\delta}(\tau, \rho) = (\tilde{\delta}_{i,j}^{\tau,\rho}, (i, j) \in E)$ характеристический вектор базисного разреза $CC(\tau, \rho)$, где $(\tau, \rho) \in E_0$ с координатами

$$\tilde{\delta}_{i,j}^{\tau,\rho} = \begin{cases} 0, & (i, j) \notin CC(\tau, \rho), \\ 1, & (i, j) \in CC^+(\tau, \rho), \\ -1, & (i, j) \in CC^-(\tau, \rho). \end{cases}$$

Введем определение характеристического вектора разреза $CC(V')$ как $\tilde{\delta}(CC(V')) = \tilde{\delta}_{i,j}(CC(V')), (i, j) \in E$.

Случайным образом выберем направление обхода в разрезе $CC(V')$. Сформируем множество $CC^+(V')$ прямых дуг и множество $CC^-(V')$ обратных дуг разреза $CC(V')$, тогда значения элементов характеристического вектора могут быть получены из системы

$$\tilde{\delta}_{i,j}(CC(V')) = \begin{cases} 0, & (i, j) \notin CC(V'), \\ 1, & (i, j) \in CC^+(V'), \\ -1, & (i, j) \in CC^-(V'). \end{cases} \quad (1.10)$$

Теорема 1.5 *Характеристический вектор любого разреза $CC(V')$ может быть представлен следующим образом:*

$$\tilde{\delta}(CC(V')) = \sum_{(i,j) \in E_0} \tilde{\delta}_{i,j}(CC(V')) \tilde{\delta}(i,j), \quad (1.11)$$

где компоненты $\tilde{\delta}_{\rho,\tau}(CC(V'))$ могут быть найдены из системы (1.10).

Теорема 1.6 *Пусть $\bar{G} = (V, E_0)$ – произвольное покрывающее дерево графа $G = (V, E)$. Для него справедливы следующие утверждения:*

1. В множество дуг базисного цикла $C(\tau, \rho)$ входят порождающая его дуга $(\tau, \rho) \in E \setminus E_0$ и дуги (i, j) множества E_0 , которые порождают базисные разрезы $CC(i, j)$, содержащие дугу (τ, ρ) .

2. В свою очередь множество дуг базисного разреза $CC(\tau, \rho)$ содержит порождающую его дугу (τ, ρ) и дуги $(i, j) \in E_0$, которые порождают базисные циклы $C(i, j)$, содержащие дугу (τ, ρ) .

Теорема 1.7 *Выберем в графе $G = (V, E)$ некоторые цикл $C(V')$ и разрез $CC(V')$. Их характеристические векторы $\delta(C)$ и $\tilde{\delta}(CC)$ удовлетворяют соотношению*

$$[\delta(C)]' \tilde{\delta}(CC) = 0. \quad (1.12)$$

1.9 Алгоритм построения множества дуг фундаментального разреза

Рассмотрим задачу принадлежности дуги (τ, ρ) орграфа $G = (V, E)$ множеству дуг фундаментального разреза, порожденного дугой (r, t) , и опишем алгоритм ее решения.

Выберем две произвольные вершины τ и ρ из множества V . Проверим, находится ли дуга, соединяющая указанные узлы, в фундаментальном разрезе $CC(r, t)$. Для начала рекуррентно пройдем по списку предков от узла τ до корневой вершины i дерева леса. Этот путь представим в виде $\tau, pred[\tau], pred[pred[\tau]], \dots, i$. Схожим образом построим путь от вершины ρ . Предположим, что мы попали в узел j – корневую вершину дерева леса, тогда пройденный путь имеет вид $\rho, pred[\rho], pred[pred[\rho]], \dots, j$. В случае совпадения вершин i и j можем утверждать, что дуга (τ, ρ) отсутствует в разрезе $CC(r, t)$, если верно обратное, то считаем, что дуга (τ, ρ) находится в разрезе $CC(r, t)$. Очевидно, что при несовпадении корневых вершин i и j пути к этим узлам от вершин τ и ρ соответственно не пересекаются.

Решим описанную выше задачу для каждой дуги (τ, ρ) графа G и для каждой дуги (r, t) покрывающего дерева $\bar{G} = (V, E_0)$, таким образом мы сможем определить множества дуг фундаментальных разрезов.

Рассмотрим пример построения системы базисных разрезов и их характеристических векторов относительно покрывающего дерева.

Граф $G = (V, E)$, представленный на рисунке 1.4, имеет множество узлов $V = \{1, 2, 3, 4, 5\}$ и множество дуг $E = \{(1, 2), (1, 5), (3, 1), (3, 2), (4, 1), (4, 3), (5, 4)\}$.

Пусть множество дуг его покрывающего дерева $\bar{G} = (V, E_0)$ равно $E_0 = \{(1, 5), (1, 2), (4, 1), (4, 3)\} \in E$.

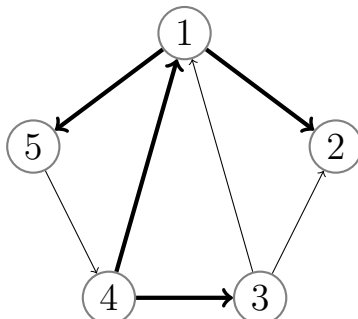


Рисунок 1.4 – Граф $G = (V, E)$ и его покрывающее дерево $\bar{G} = (V, E_0)$

Построим систему фундаментальных разрезов графа G на примере разреза $CC(1, 2)$. Он содержит порождающую его дугу $(1, 2)$ и дугу $(3, 2)$, поскольку корневые вершины деревьев леса, содержащих узлы 3 и 2 равны соответственно 4 и 2, т.е. не совпадают.

Аналогичным образом произведем проверку для всех базисных разрезов и дуг графа.

Система базисных разрезов графа G относительно покрывающего дерева имеет следующий вид:

$$CC(1, 2) = \{(1, 2), (3, 2)\}, CC(1, 5) = \{(1, 5), (5, 4)\},$$

$$CC(4, 1) = \{(4, 1), (3, 1), (3, 2), (5, 4)\}, CC(4, 3) = \{(4, 3), (3, 1), (3, 2)\}.$$

Подсчитаем элементы характеристических векторов базисных разрезов графа G относительно покрывающего дерева \bar{G} . Результаты приведены в таблице 1.4.

Таблица 1.4 – Характеристические векторы базисных разрезов графа G относительно покрывающего дерева \bar{G}

(i, j)	$(1, 2)$	$(1, 5)$	$(3, 1)$	$(3, 2)$	$(4, 1)$	$(4, 3)$	$(5, 4)$
$\tilde{\delta}(\tau, \rho)$	$\tilde{\delta}_{1,2}^{\tau, \rho}$	$\tilde{\delta}_{1,5}^{\tau, \rho}$	$\tilde{\delta}_{3,1}^{\tau, \rho}$	$\tilde{\delta}_{3,2}^{\tau, \rho}$	$\tilde{\delta}_{4,1}^{\tau, \rho}$	$\tilde{\delta}_{4,3}^{\tau, \rho}$	$\tilde{\delta}_{5,4}^{\tau, \rho}$
$\tilde{\delta}(1, 2)$	1	0	0	1	0	0	0
$\tilde{\delta}(1, 5)$	0	1	0	0	0	0	-1
$\tilde{\delta}(4, 1)$	0	0	1	1	1	0	1
$\tilde{\delta}(4, 3)$	0	0	-1	-1	0	1	0

ГЛАВА 2

ОЦЕНКА ПОТОКОВ НА НЕНАБЛЮДАЕМОЙ ЧАСТИ СЕТИ ПУТЕМ ИСПОЛЬЗОВАНИЯ РАЗРЕЖЕННЫХ СИСТЕМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ИДЕНТИФИКАЦИИ СЕНСОРНЫХ КОНФИГУРАЦИЙ УЗЛОВ

2.1 Идентификация сенсорных конфигураций узлов

Для сбора информации о транспортных потоках устанавливают сенсоры в узлах сети. Сделать это во всех узлах не представляется возможным из-за экономических затрат. Чтобы решить поставленную задачу необходимо определить множество обзереваемых узлов, в которых будут расположены специальные программируемые устройства. Они позволят собрать информацию о транспортных потоках на всей сети. Данные, полученные благодаря установленным в обзереваемых узлах сенсорам, позволяют построить специальную разреженную систему линейных алгебраических уравнений. Единственность решения такой системы позволяет обеспечить полный контроль за состоянием сети.

Исследуем разреженную систему линейных алгебраических уравнений. Для этого выберем ориентированный связный двунаправленный граф G с множеством вершин V и множеством дуг E , заданным на декартовом произведении $V \times V$ ($|V| < \infty, |E| < \infty$).

Введем в рассмотрение множества $V_i^+(E) = \{j \in V : (i, j) \in E\}$, $V_i^-(E) = \{j \in V : (j, i) \in E\}$. Обозначим величину потока вдоль дуги (i, j) через $x_{i,j}$, а величину внешнего потока узла $i \in V^*$ через x_i , тогда система уравнений приобретает вид

$$\sum_{j \in V_i^+(E)} x_{i,j} - \sum_{j \in V_i^-(E)} x_{j,i} = \begin{cases} 0, & i \in V \setminus V^*, \\ x_i \cdot \text{sign}(i), & i \in V^*. \end{cases} \quad (2.1)$$

$$\sum_{(i,j) \in E} \lambda_{i,j}^p x_{i,j} + \sum_{i \in V^*} \lambda_i^p x_i = \beta_p, p = \overline{1, q}, \quad (2.2)$$

где $\text{sign}(i) = \begin{cases} 1, & i \in V_+^*, \\ -1, & i \in V_-^*. \end{cases}$

Построим матрицу системы уравнений 2.1 - 2.2. Примем следующие условные обозначения: M – матрица с блочно-диагональной структурой размера $|V| \times |E|$ ориентированного графа $G = (V, E)$; Q – матрица размера $q \times |E|$ с элементами $\lambda_{i,j}^p, (i, j) \in E, p = \overline{1, q}$; разреженную матрицу размера $|V| \times |V^*|$ обозначим B ; матрицу размера $q \times |V^*|$, состоящую из элементов $\lambda_i^p, i \in V^*, p = \overline{1, q}$, обозначим T .

Построенная матрица имеет вид:

$$A = \begin{bmatrix} M & B \\ Q & T \end{bmatrix}. \quad (2.3)$$

Необходимо учесть, что матрица B в столбце j и строке i включает ненулевой элемент, равный $-sign(i) \forall j \in V^*$, все остальные элементы равны нулю.

При условии соблюдения равенства $rank(A) = |V| + q$ и неравенства $|V| + q < |E| + |V^*|$ система 2.1 - 2.2 считается недоопределенной системой линейных алгебраических уравнений. [12]

2.2 Недоопределенная разреженная линейная система

Эффективность метода декомпозиции для определения потоков в сети рассмотрим на примере двунаправленного ориентированного графа G . Численные значения множеств вершин и дуг графа представим равенствами $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $E = \{(1, 2), (2, 1), (1, 5), (5, 1), (2, 8), (8, 2), (3, 4), (4, 3), (3, 8), (8, 3), (4, 7), (7, 4), (8, 5), (5, 8), (7, 6), (6, 7), (2, 5), (5, 2), (6, 4), (4, 6), (9, 6), (6, 9), (9, 5), (5, 9), (9, 2), (2, 9), (3, 5), (5, 3)\}$, $V^* = \{2, 3, 7, 8\}$ (см. рисунок 2.1).

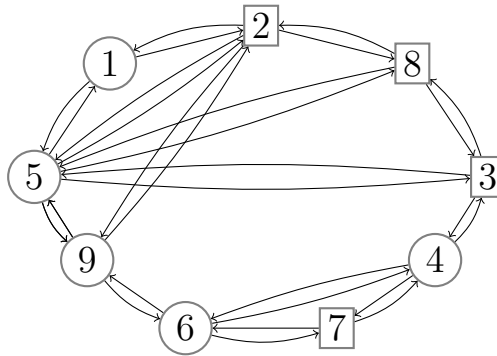


Рисунок 2.1 – Двунаправленный орграф $G = (V, E)$

На рисунке 2.1 узлы с переменным внешним потоком $x_i, i \in V^*$ обозначены квадратами.

Множество обозреваемых узлов обозначим буквой M , придадим ему числовое выражение $\{9\}$. Удалим из графа G все узлы из множества M (см. рисунок 2.2). На множестве M построим разрез $CC(M) = \{(6, 9), (9, 6), (5, 9), (9, 5), (2, 9), (9, 2)\}$ и сформируем множество $V(CC(M)) = \{2, 5, 6, 9\}$, обозначим $M^+ = V(CC(M)) \setminus M = \{2, 5, 6\}$. Объединим множества M^+ и M , условно обозначим полученное множество M^* . Оно содержит значения $\{2, 5, 6, 9\}$. Рассмотрим еще одно множество: $\tilde{V} = V \setminus M^* = \{1, 3, 4, 7, 8\}$ [6].

Чтобы получить информацию о ненаблюдаемой части ориентированного графа G удалим исходящие из узлов множества M^+ дуги. Выполнить

данную операцию нам позволяет наличие информации о значении потока [6]. В результате получим оргграф $\bar{G} = (\bar{V}, \bar{E})$, множества вершин и дуг в котором равны соответственно $\bar{V} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ и $\bar{E} = \{(1, 2), (1, 5), (3, 4), (3, 5), (3, 8), (4, 3), (4, 6), (4, 7), (7, 4), (7, 6), (8, 2), (8, 3), (8, 5)\}$ (см. рисунок 2.3).

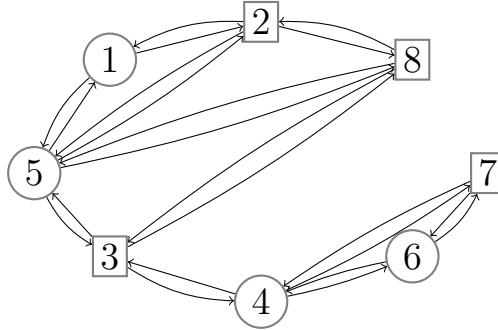


Рисунок 2.2 – Орграф после удаления обозреваемого узла $M = \{9\}$ и соответствующих входящих и исходящих дуг из узла $i = 9$

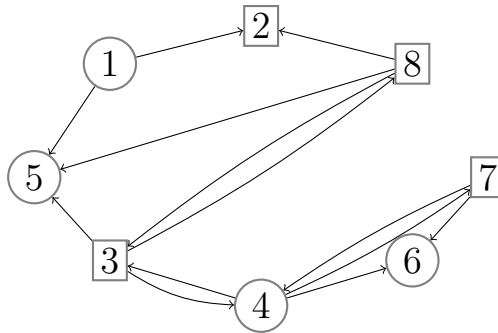


Рисунок 2.3 – Орграф \bar{G} – ненаблюдаемая часть орграфа G

Для построения опоры (остовного дерева) T графа с пустым множеством V^* будем использовать граф \bar{G} . К имеющемуся графу добавим фиктивный узел ξ и дуги $(\xi, 2), (\xi, 3), (\xi, 7), (\xi, 8)$. Получим новый граф $\hat{G} = (\hat{V}, \hat{E})$ (для обозначения опоры полученного орграфа на рисунке 2.4 использованы жирные дуги).

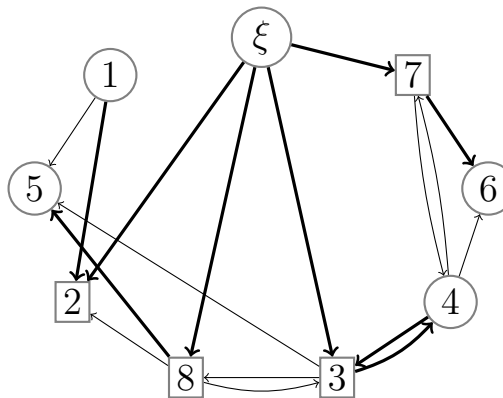


Рисунок 2.4 – Орграф \hat{G} с помеченной опорой для системы 2.1 при $V^* = \emptyset$

Чтобы сохранить в памяти компьютера корневое дерево, построим вспомогательные корневые структуры. [6] Для этого выберем узел ξ в качестве корня. Результаты построения представлены в таблице 2.1.

Таблица 2.1 – Корневые структуры для хранения покрывающего дерева T графа \widehat{G}

i	1	2	3	4	5	6	7	8	ξ
$pred[i]$	2	ξ	ξ	3	8	7	ξ	ξ	0
$dir[i]$	-1	1	1	1	1	1	1	1	0
$depth[i]$	2	1	1	2	2	2	1	1	0

Используя данные таблицы произведем вычисления правой части $a_i, i \in \overline{V}$ орграфа \widehat{G} . Преобразованная система уравнений баланса приобретает следующий вид:

$$\begin{aligned}
-x_{1,2} - x_{1,5} &= -\frac{f_{2,9}p_{2,1}}{p_{2,9}} - \frac{f_{5,9}p_{5,1}}{p_{5,9}}, \\
x_{\xi,2} + x_{1,2} + x_{8,2} &= \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,8} + p_{2,9})}{p_{2,9}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}} - f_{9,2}, \\
x_{1,5} + x_{3,5} + x_{8,5} &= \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - f_{9,5}, \\
x_{\xi,8} + x_{3,8} - x_{8,2} - x_{8,3} - x_{8,5} &= -\frac{f_{2,9}p_{2,8}}{p_{2,9}} - \frac{f_{5,9}p_{5,8}}{p_{5,9}}, \\
x_{\xi,3} - x_{3,4} - x_{3,5} - x_{3,8} + x_{4,3} + x_{8,3} &= -\frac{f_{5,9}p_{5,3}}{p_{5,9}}, \\
x_{3,4} - x_{4,3} - x_{4,6} - x_{4,7} + x_{7,4} &= -\frac{f_{6,9}p_{6,4}}{p_{6,9}}, \\
x_{\xi,7} + x_{4,7} - x_{7,4} - x_{7,6} &= -\frac{f_{6,9}p_{6,7}}{p_{6,9}}, \\
x_{4,6} + x_{7,6} &= \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - f_{9,6}, \\
-x_{\xi,2} - x_{\xi,3} - x_{\xi,7} - x_{\xi,8} &= -f_{2,9} - f_{5,9} - f_{6,9} + f_{9,2} + f_{9,5} + f_{9,6}.
\end{aligned} \tag{2.4}$$

Будем считать нулевыми значения всех неопорных потоков, тогда частное решение $\tilde{x} = (\tilde{x}_{i,j}, (i,j) \in \widehat{E})$ системы 2.4 можно представить как

$$\begin{aligned}
\tilde{x}_{1,5} &\rightarrow 0, \quad \tilde{x}_{3,5} \rightarrow 0, \quad \tilde{x}_{3,8} \rightarrow 0, \quad \tilde{x}_{4,3} \rightarrow 0, \quad \tilde{x}_{4,6} \rightarrow 0, \quad \tilde{x}_{4,7} \rightarrow 0, \\
\tilde{x}_{7,4} &\rightarrow 0, \quad \tilde{x}_{8,2} \rightarrow 0, \quad \tilde{x}_{8,3} \rightarrow 0, \quad \tilde{x}_{1,2} \rightarrow \frac{f_{2,9}p_{2,1}}{p_{2,9}} + \frac{f_{5,9}p_{5,1}}{p_{5,9}}, \\
\tilde{x}_{3,4} &\rightarrow -\frac{f_{6,9}p_{6,4}}{p_{6,9}}, \quad \tilde{x}_{7,6} \rightarrow \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - f_{9,6}, \\
\tilde{x}_{8,5} &\rightarrow \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - f_{9,5}, \\
\tilde{x}_{9,2} &\rightarrow \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,8} + p_{2,9})}{p_{2,9}} - \frac{f_{2,9}p_{2,1}}{p_{2,9}} - \frac{f_{5,9}p_{5,1}}{p_{5,9}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}} - f_{9,2}, \\
\tilde{x}_{9,3} &\rightarrow -\frac{f_{5,9}p_{5,3}}{p_{5,9}} - \frac{f_{6,9}p_{6,4}}{p_{6,9}}, \quad \tilde{x}_{9,7} \rightarrow \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - \frac{f_{6,9}p_{6,7}}{p_{6,9}} - f_{9,6}, \\
\tilde{x}_{9,8} &\rightarrow \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \frac{f_{2,9}p_{2,8}}{p_{2,9}} - \frac{f_{5,9}p_{5,8}}{p_{5,9}} \\
&\quad - f_{9,5}.
\end{aligned} \tag{2.5}$$

Для нахождения общего решения задачи нам необходимо построить характеристические векторы $\delta_{ij}(\tau, \rho)$. При построении будем учитывать дуги, инцидентные фиктивному узлу ξ (см. таблицу 2.3). Система характеристических векторов представляет собой базис пространства решений [6] однородной системы, порожденной системой 2.4. Представим результаты в виде таблиц.

Таблица 2.2 – Характеристические векторы

	$\delta_{1,2}$	$\delta_{1,5}$	$\delta_{8,2}$	$\delta_{3,4}$	$\delta_{4,3}$	$\delta_{3,8}$	$\delta_{8,3}$	$\delta_{4,7}$	$\delta_{7,4}$	$\delta_{8,5}$	$\delta_{7,6}$	$\delta_{4,6}$	$\delta_{3,5}$
(1,5)	-1	1	0	0	0	0	0	0	0	-1	0	0	0
(8,2)	0	0	1	0	0	0	0	0	0	0	0	0	0
(4,3)	0	0	0	1	1	0	0	0	0	0	0	0	0
(3,8)	0	0	0	0	0	1	0	0	0	0	0	0	0
(8,3)	0	0	0	0	0	0	1	0	0	0	0	0	0
(4,7)	0	0	0	1	0	0	0	1	0	0	0	0	0
(7,4)	0	0	0	-1	0	0	0	0	1	0	0	0	0
(4,6)	0	0	0	1	0	0	0	0	0	0	-1	1	0
(3,5)	0	0	0	0	0	0	0	0	0	-1	0	0	1

Таблица 2.3 – Характеристические векторы для дуг, инцидентных ξ

	$\delta_{\xi,2}$	$\delta_{\xi,3}$	$\delta_{\xi,7}$	$\delta_{\xi,8}$
(1,5)	1	0	0	-1
(8,2)	-1	0	0	1
(4,3)	0	0	0	0
(3,8)	0	1	0	-1
(8,3)	0	-1	0	1
(4,7)	0	1	-1	0
(7,4)	0	-1	1	0
(4,6)	0	1	-1	0
(3,5)	0	1	0	-1

$$\begin{aligned}
 x_{1,2} &\rightarrow \frac{f_{2,9}p_{2,1}}{p_{2,9}} + \frac{f_{5,9}p_{5,1}}{p_{5,9}} - x_{1,5}, \\
 x_{3,4} &\rightarrow -\frac{f_{6,9}p_{6,4}}{p_{6,9}} + x_{4,3} + x_{4,6} + x_{4,7} - x_{7,4}, \\
 x_{8,5} &\rightarrow \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - f_{9,5} - x_{1,5} - x_{3,5}, \\
 x_{7,6} &\rightarrow \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - f_{9,6} - x_{4,6}, \\
 x_{\xi,2} &\rightarrow \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,8} + p_{2,9})}{p_{2,9}} - \frac{f_{2,9}p_{2,1}}{p_{2,9}} - \frac{f_{5,9}p_{5,1}}{p_{5,9}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}} \\
 &\quad - f_{9,2} + x_{1,5} - x_{8,2}, \\
 x_{\xi,3} &\rightarrow -\frac{f_{5,9}p_{5,3}}{p_{5,9}} - \frac{f_{6,9}p_{6,4}}{p_{6,9}} + x_{3,5} + x_{3,8} + x_{4,6} + x_{4,7} - x_{7,4} - x_{8,3}, \\
 x_{\xi,7} &\rightarrow \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - \frac{f_{6,9}p_{6,7}}{p_{6,9}} - f_{9,6} - x_{4,6} - x_{4,7} + x_{7,4}, \\
 x_{\xi,8} &\rightarrow \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \frac{f_{2,9}p_{2,8}}{p_{2,9}} \\
 &\quad - \frac{f_{5,9}p_{5,8}}{p_{5,9}} - f_{9,5} - x_{1,5} - x_{3,5} - x_{3,8} + x_{8,2} + x_{8,3}
 \end{aligned}$$

Система дополнительных уравнений [6] имеет вид:

$$\begin{aligned}
 x_{1,2} - \frac{p_{1,5}x_{1,5}}{p_{1,2}} = 0, \quad x_{3,4} - \frac{p_{3,8}x_{3,8}}{p_{3,4}} = 0, \quad x_{3,4} - \frac{p_{3,5}x_{3,5}}{p_{3,4}} = 0, \\
 x_{4,3} - \frac{p_{4,7}x_{4,7}}{p_{4,3}} = 0, \quad x_{4,3} - \frac{p_{4,6}x_{4,6}}{p_{4,3}} = 0, \quad x_{7,4} - \frac{p_{7,6}x_{7,6}}{p_{7,4}} = 0, \\
 x_{8,2} - \frac{p_{8,3}x_{8,3}}{p_{8,2}} = 0, \quad x_{8,2} - \frac{p_{8,5}x_{8,5}}{p_{8,2}} = 0.
 \end{aligned} \tag{2.6}$$

Запишем матрицу Λ системы (2.6):

$$\Lambda = \begin{pmatrix} -\frac{p_{1,5}}{p_{1,2}} - 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{p_{3,8}}{p_{3,4}} & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & 1 & -\frac{p_{3,5}}{p_{3,4}} & 0 \\ 0 & 0 & 1 & 0 & 0 & -\frac{p_{4,7}}{p_{4,3}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{p_{4,6}}{p_{4,3}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{p_{7,6}}{p_{7,4}} & 0 & 0 \\ 0 & 1 & 0 & 0 & -\frac{p_{8,3}}{p_{8,2}} & 0 & 0 & 0 & 0 & 0 \\ \frac{p_{8,5}}{p_{8,2}} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{p_{8,5}}{p_{8,2}} \end{pmatrix}.$$

Поскольку $rank(\Lambda) = 8$, а число неопорных дуг равно 9, то система является недоопределенной. Выберем дуги $\widehat{E}_W = \{(1, 5), (8, 2), (4, 3), (3, 8), (8, 3), (4, 7), (7, 4), (4, 6)\}$, тогда $\widehat{E}_{-T} \setminus \widehat{U}_W = \{(3, 5)\}$.

$$\Lambda_W = \begin{pmatrix} -\frac{p_{1,5}}{p_{1,2}} - 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{p_{3,8}}{p_{3,4}} & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -\frac{p_{4,7}}{p_{4,3}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{p_{4,6}}{p_{4,3}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{p_{7,6}}{p_{7,4}} & 0 & 0 \\ 0 & 1 & 0 & 0 & -\frac{p_{8,3}}{p_{8,2}} & 0 & 0 & 0 & 0 & 0 \\ \frac{p_{8,5}}{p_{8,2}} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

Таким образом, определитель матрицы Λ_W равен

$$\det(\Lambda_W) = \frac{(p_{1,2} + p_{1,5}) p_{3,8} (p_{4,6} p_{4,7} p_{7,4} + p_{4,3} (p_{4,6} p_{7,4} + p_{4,7} (p_{7,4} + p_{7,6}))) p_{8,3}}{p_{1,2} p_{3,4} p_{4,3}^2 p_{7,4} p_{8,2}}.$$

Т.к. $p_{ij} > 0$, то в выражении $\det(\Lambda_W)$ все слагаемые строго положительны, следовательно, $\det(\Lambda_W) \neq 0$, значит, матрица Λ_W невырожденная.

Вычислим правую часть β :

$$\beta = \begin{pmatrix} -\frac{f_{2,9}p_{2,1}}{p_{2,9}} - \frac{f_{5,9}p_{5,1}}{p_{5,9}} \\ \frac{f_{6,9}p_{6,4}}{p_{6,9}} \\ \frac{f_{6,9}p_{6,4}}{p_{6,9}} + \frac{p_{3,5}x_{3,5}}{p_{3,4}} \\ 0 \\ 0 \\ p_{7,6} \left(\frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - f_{9,6} \right) p_{7,4}^{-1} \\ 0 \\ p_{8,5} \left(\frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - f_{9,5} - x_{3,5} \right) \\ \hline p_{8,2} \end{pmatrix} \quad (2.7)$$

Частное решение системы позволили составить системы 2.4, 2.6. Для получения общего решения предложенных систем можно использовать метод декомпозиции, описанный в [6]. Этот путь позволит контролировать потоки на ненаблюдаемой части сети.

2.3 Переопределенная разреженная линейная система

Двунаправленный орграф, представленный на рисунке 2.1, содержит 9 узлов. Попробуем изменить количество обзереваемых узлов либо их конфигурацию. В результате таких преобразований ранг матрицы системы может измениться, тогда система из неопределенной становится переопределенной или имеет единственное решение. Нам необходимо найти решение для переопределенной системы. С этой целью удалим из графа G (рисунок 2.1) узлы из множества $M = \{8, 9\}$ и, соответственно, входящие и исходящие дуги из этих узлов дуги (см. рисунок 2.5).

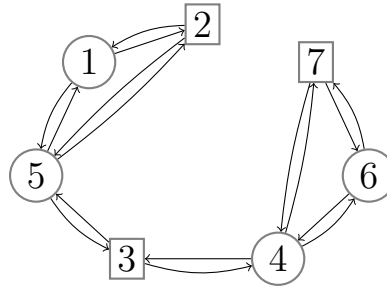


Рисунок 2.5 – Орграф после удаления обзереваемых узлов и соответствующих входящих и исходящих дуг из узлов множества $M = \{8, 9\}$

Имеющаяся информация о потоке на узлах $i \in M^+$ позволяет удалить исходящие из них дуги. Таким образом, получаем орграф \bar{G} , с множеством узлов $\bar{V} = \{1, 2, 3, 4, 5, 6, 7\}$, и множеством дуг $\bar{E} = \{(1, 2), (1, 5), (4, 3), (4, 7), (7, 4), (7, 6), (4, 6)\}$. (см. рисунок 2.6)

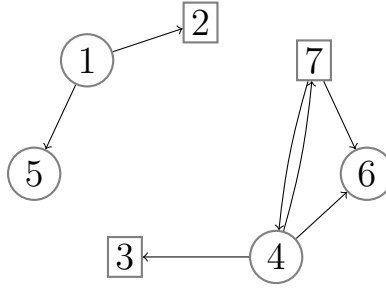


Рисунок 2.6 – Орграф $\bar{G} = (\bar{V}, \bar{E})$, содержащий две компоненты связности

Если орграф $\bar{G} = (\bar{V}, \bar{E})$ дополнить узлом ξ и дугами $(\xi, 2), (\xi, 3), (\xi, 7)$, можно получить новый орграф $\hat{G} = (\hat{V}, \hat{E}), \hat{V} = \bar{V} \cup \{\xi\}, \hat{E} = \bar{E} \cup \{(\xi, 2), (\xi, 3), (\xi, 7)\}$. Это позволяет построить его опору T [5] (на рисунке 2.7 опора полученного орграфа отмечена жирными дугами).

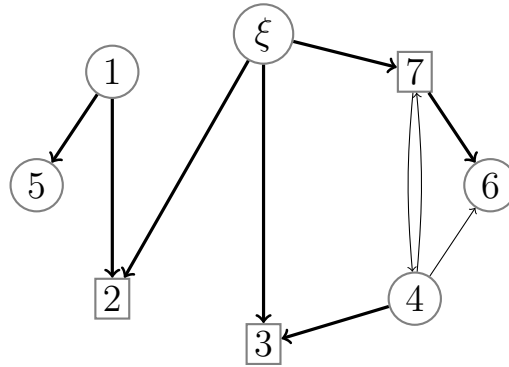


Рисунок 2.7 – Орграф \hat{G} с помеченной опорой для системы (8)

Для построения корневых и вспомогательных структур для опоры орграфа \hat{G} выберем в качестве корня узел ξ . Результаты построения представлены в таблице 2.4.

Таблица 2.4 – Корневые структуры для хранения T орграфа \hat{G}

i	1	2	3	4	5	6	7	ξ
$pred[i]$	2	ξ	ξ	3	1	7	ξ	0
$dir[i]$	-1	1	1	-1	1	1	1	0
$depth[i]$	2	1	1	2	3	2	1	0

Информация, полученная от сенсоров и преобразований системы 2.4 позволяет вычислить значения $a_i, i \in \bar{V}$ для орграфа \hat{G} .

Система уравнений баланса приобретает вид:

$$\begin{aligned}
-x_{1,2} - x_{1,5} &= -\frac{f_{2,8}p_{2,1}}{p_{2,8}} - \frac{f_{2,9}p_{2,1}}{p_{2,9}} + p_{5,1} \left(-\frac{f_{5,8}}{p_{5,8}} - \frac{f_{5,9}}{p_{5,9}} \right), \\
x_{\xi,2} + x_{1,2} &= \frac{f_{2,8}(p_{2,1} + p_{2,5} + p_{2,8})}{p_{2,8}} + \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,9})}{p_{2,9}} - \\
&\quad - \frac{f_{5,8}p_{5,2}}{p_{5,8}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}} - f_{8,2} - f_{9,2}, \\
x_{1,5} &= \frac{f_{5,8}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8})}{p_{5,8}} + \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,9})}{p_{5,9}} - \\
&\quad - \frac{f_{2,8}p_{2,5}}{p_{2,8}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \frac{f_{3,8}p_{3,5}}{p_{3,8}} - f_{8,5} - f_{9,5}, \\
x_{\xi,3} + x_{4,3} &= \frac{f_{3,8}(p_{3,4} + p_{3,5} + p_{3,8})}{p_{3,8}} + p_{5,3} \left(-\frac{f_{5,8}}{p_{5,8}} - \frac{f_{5,9}}{p_{5,9}} \right) - f_{8,3}, \\
-x_{4,3} - x_{4,6} - x_{4,7} + x_{7,4} &= -\frac{f_{3,8}p_{3,4}}{p_{3,8}} - \frac{f_{6,9}p_{6,4}}{p_{6,9}}, \\
x_{\xi,7} + x_{4,7} - x_{7,4} - x_{7,6} &= -\frac{f_{6,9}p_{6,7}}{p_{6,9}}, \\
x_{4,6} + x_{7,6} &= \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - f_{9,6}, \\
-x_{\xi,2} - x_{\xi,3} - x_{\xi,7} &= -f_{2,8} - f_{2,9} - f_{3,8} - f_{5,8} - f_{5,9} - f_{6,9} + f_{8,2} + f_{8,3} + \\
&\quad + f_{8,5} + f_{9,2} + f_{9,5} + f_{9,6}.
\end{aligned} \tag{2.8}$$

Если предположить, что значения неопорных потоков равными 0, то частное решение системы 2.8 можно записать как:

$$\begin{aligned}
&\tilde{x}_{4,6} \rightarrow 0, \tilde{x}_{4,7} \rightarrow 0, \tilde{x}_{7,4} \rightarrow 0, \\
\tilde{x}_{1,2} &\rightarrow f_{8,5} + f_{9,5} - p_{5,1} \left(-\frac{f_{5,8}}{p_{5,8}} - \frac{f_{5,9}}{p_{5,9}} \right) + \frac{f_{2,8}p_{2,1}}{p_{2,8}} + \frac{f_{2,8}p_{2,5}}{p_{2,8}} + \frac{f_{2,9}p_{2,1}}{p_{2,9}} + \\
&+ \frac{f_{2,9}p_{2,5}}{p_{2,9}} + \frac{f_{3,8}p_{3,5}}{p_{3,8}} - \frac{f_{5,8}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8})}{p_{5,8}} - \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,9})}{p_{5,9}}, \\
\tilde{x}_{1,5} &\rightarrow -f_{8,5} - f_{9,5} + \frac{f_{5,8}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8})}{p_{5,8}} + \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,9})}{p_{5,9}} - \\
&\quad - \frac{f_{2,8}p_{2,5}}{p_{2,8}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \frac{f_{3,8}p_{3,5}}{p_{3,8}}, \\
\tilde{x}_{4,3} &\rightarrow \frac{f_{3,8}p_{3,4}}{p_{3,8}} + \frac{f_{6,9}p_{6,4}}{p_{6,9}}, \\
\tilde{x}_{7,6} &\rightarrow \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - f_{9,6},
\end{aligned}$$

$$\begin{aligned}
\tilde{x}_{8,2} &\rightarrow -f_{8,2} - f_{8,5} - f_{9,2} - f_{9,5} + \frac{f_{2,8}(p_{2,1} + p_{2,5} + p_{2,8})}{p_{2,8}} + \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,9})}{p_{2,9}} + \\
&\quad + \frac{f_{5,8}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,8})}{p_{5,8}} + p_{5,1} \left(-\frac{f_{5,8}}{p_{5,8}} - \frac{f_{5,9}}{p_{5,9}} \right) + \\
&\quad + \frac{f_{5,9}(p_{5,1} + p_{5,2} + p_{5,3} + p_{5,9})}{p_{5,9}} - \frac{f_{2,8}p_{2,1}}{p_{2,8}} - \frac{f_{2,8}p_{2,5}}{p_{2,8}} - \frac{f_{2,9}p_{2,1}}{p_{2,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \\
&\quad - \frac{f_{3,8}p_{3,5}}{p_{3,8}} - \frac{f_{5,8}p_{5,2}}{p_{5,8}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}}, \\
\tilde{x}_{8,3} &\rightarrow -f_{8,3} + \frac{f_{3,8}(p_{3,4} + p_{3,5} + p_{3,8})}{p_{3,8}} + p_{5,3} \left(-\frac{f_{5,8}}{p_{5,8}} - \frac{f_{5,9}}{p_{5,9}} \right) - \frac{f_{3,8}p_{3,4}}{p_{3,8}} - \frac{f_{6,9}p_{6,4}}{p_{6,9}}, \\
\tilde{x}_{8,7} &\rightarrow -f_{9,6} + \frac{f_{6,9}(p_{6,4} + p_{6,7} + p_{6,9})}{p_{6,9}} - \frac{f_{6,9}p_{6,7}}{p_{6,9}}
\end{aligned}$$

На основе решения системы представим характеристические векторы $\delta_{ij}(\tau, \rho)$ в виде таблицы 2.5.

Таблица 2.5 – Характеристические векторы

(τ, ρ)	$\delta_{1,2}$	$\delta_{1,5}$	$\delta_{4,3}$	$\delta_{4,7}$	$\delta_{7,4}$	$\delta_{7,6}$	$\delta_{4,6}$	$\delta_{\xi,2}$	$\delta_{\xi,3}$	$\delta_{\xi,7}$
(4,7)	0	0	-1	1	0	0	0	0	1	-1
(7,4)	0	0	1	0	1	0	0	0	-1	1
(4,6)	0	0	-1	0	0	-1	1	0	1	-1

Система дополнительных уравнений имеет вид:

$$x_{1,2} - \frac{p_{1,5}x_{1,5}}{p_{1,2}} = 0, \quad x_{4,3} - \frac{p_{4,7}x_{4,7}}{p_{4,3}} = 0, \quad x_{4,3} - \frac{p_{4,6}x_{4,6}}{p_{4,3}} = 0, \quad x_{7,4} - \frac{p_{7,6}x_{7,6}}{p_{7,4}} = 0 \quad (2.9)$$

Составим матрицу Λ системы (2.9):

$$\Lambda = \begin{pmatrix} 0 & 0 & 0 \\ -\frac{p_{4,7}}{p_{4,3}} - 1 & 1 & -1 \\ -1 & 1 & -\frac{p_{4,6}}{p_{4,3}} - 1 \\ 0 & 1 & \frac{p_{7,6}}{p_{7,4}} \end{pmatrix}.$$

Поскольку ранг матрицы Λ равен 3, что совпадает с количеством небазисных дуг, а значение $q = 4$, то рассматриваемая система является переопределенной. Выберем множество $\widehat{E}_W = \widehat{E}_{-T}$, тогда $\widehat{E}_{-T} \setminus \widehat{E}_W$ является пустым множеством.

Из четырех уравнений системы выбираем 3 линейно независимых, поскольку оставшееся линейно зависит от выбранных и выполняется при условии корректно выбранных коэффициентов p_{ij} и значений f_{ij} .

Из системы не будем использовать первое уравнение, таким образом,

оставшиеся уравнения формируют подматрицу

$$\Lambda_W = \begin{pmatrix} -\frac{p_{4,7}}{p_{4,3}} - 1 & 1 & -1 \\ -1 & 1 & -\frac{p_{4,6}}{p_{4,3}} - 1 \\ 0 & 1 & \frac{p_{7,6}}{p_{7,4}} \end{pmatrix}$$

Определитель этой матрицы равен

$$\det(\Lambda_W) = \frac{p_{4,3} \left(-p_{4,6} - \frac{p_{4,7}(p_{7,4} + p_{7,6})}{p_{7,4}} \right) - p_{4,6}p_{4,7}}{p_{4,3}^2}.$$

Т.к. значения p_{ij} больше нуля, то в определителе матрицы Λ_W все слагаемые должны быть строго отрицательны. Из этого следует невырожденность матрицы Λ_W , поскольку определитель $\det(\Lambda_W)$ не равен нулю.

Описанный алгоритм получения частного решения может быть использован при нахождении общего решения систем 2.8, 2.9.

2.4 Оптимальное решение задачи оценки потока

Для нахождения оптимального решения задачи оценки потока на ненаблюдаемой части сети установим сенсор в один из узлов графа G . Пусть это будет узел 5, т.е. $M = \{5\}$. Для упрощения реализации алгоритма изменим нумерацию узлов графа G , при этом обозреваемым узлам придадим максимальные номера. Изменение номера узла 5 на 9, а узла 9 на 5 приводит к получению нового орграфа $G' = (V', E')$, где множество вершин нового графа V' совпадает с множеством вершин V графа G , а дуги нового графа образуют множество $E' = \{(1, 2), (2, 1), (1, 9), (9, 1), (2, 8), (8, 2), (3, 4), (4, 3), (3, 8), (8, 3), (4, 7), (7, 4), (8, 9), (9, 8), (7, 6), (6, 7), (2, 9), (9, 2), (6, 4), (4, 6), (5, 6), (6, 5), (5, 9), (9, 5), (5, 2), (2, 5), (3, 9), (9, 3)\}$ (см. рисунок 2.8). В результате таких изменений множество обозреваемых узлов $M' = \{9\}$ содержит только узел 9.

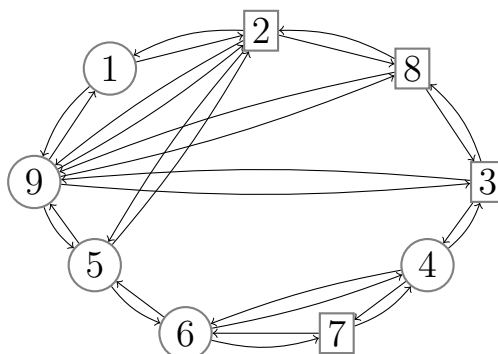


Рисунок 2.8 – Орграф $G' = (V', E')$

Учитывая известные значения потоков в узлах множества M' , мы можем удалить из орграфа G' все узлы и дуги, исходящие и входящие в узлы множества M' (см. рисунок 2.9). Теперь согласно [6] мы можем построить разрез $CC(M') = \{(1, 9), (9, 1), (8, 9), (9, 8), (2, 9), (9, 2), (5, 9), (9, 5), (3, 9), (9, 3)\}$. Найдем множество $V(CC(M')) = \{1, 2, 3, 5, 8, 9\}$, построим $M^+ = V(CC(M')) \setminus M = \{1, 2, 3, 5, 8\}$, сформируем множества $M^* = M \cup M^+ = \{1, 2, 3, 5, 8, 9\}$ и $\widetilde{V}' = V \setminus M^* = \{4, 6, 7\}$.

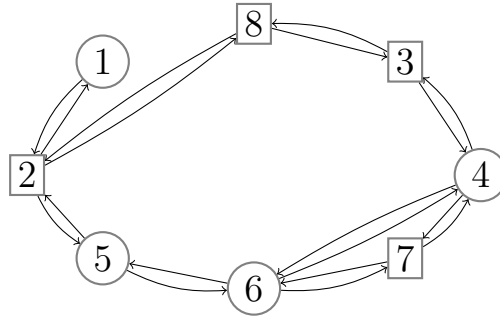


Рисунок 2.9 – Орграф после удаления обозреваемых узлов и дуг, входящих и исходящих из узла 9

Для получения нового орграфа $\overline{G} = (\overline{V}, \overline{E})$ удалим дуги, исходящие из узлов $i \in M^+$. Тогда множество вершин полученного графа \overline{V} равно $\{1, 2, 3, 4, 5, 6, 7, 8\}$, а дуги составлены из узлов $\overline{E} = \{(4, 3), (4, 7), (7, 4), (7, 6), (6, 7), (6, 4), (4, 6), (6, 5)\}$ (см. рисунок 2.10) [6].

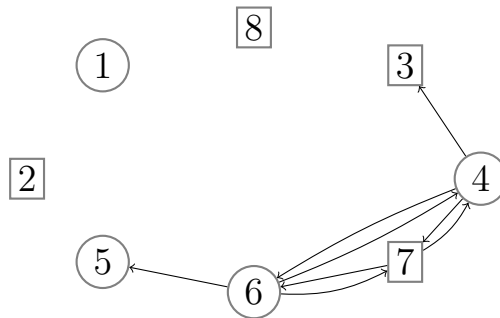


Рисунок 2.10 – Орграф \overline{G}

Для построения опоры T графа $\overline{G} = (\overline{V}, \overline{E})$ дополним его узлом ξ и дугами $(\xi, 2), (\xi, 3), (\xi, 7), (\xi, 8)$. Получаем новый орграф $\widehat{G} = (\widehat{V}, \widehat{E})$, с множеством узлов $\widehat{V} = \overline{V} \cup \{\xi\}$ и множеством дуг $\widehat{E} = \overline{E} \cup \{(\xi, 2), (\xi, 3), (\xi, 7), (\xi, 8)\}$.

Полученный орграф представим в виде рисунка (на рисунке 2.11 опора графа отмечена жирными дугами).

Как видно из рисунка, узел 1 относится к отдельной компоненте связности. Из этого следует, что при соблюдении корректности выходных данных будут выполнены условия:

$$\frac{f_{2,9}p_{2,1}}{p_{2,9}} + f_{9,1} = f_{1,9} \left(\frac{p_{1,2}}{p_{1,9}} + 1 \right).$$

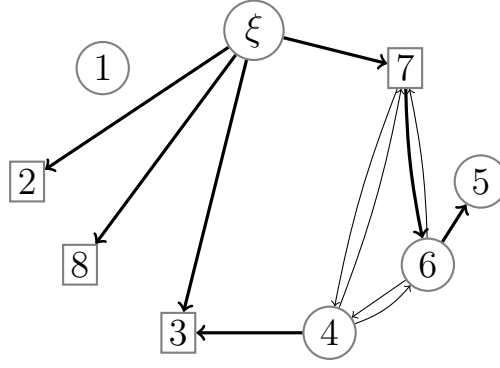


Рисунок 2.11 – Орграф \widehat{G} с помеченной опорой для системы (10)

Нам осталось решить задачу оценки потока для оставшейся компоненты связности. С этой целью построим корневые и вспомогательные структуры, выбрав в качестве корня узел ξ . Данное построение представим в таблице 2.6.

Таблица 2.6 – Корневые структуры для хранения корневого дерева T с корнем в узле ξ

i	2	3	4	5	6	7	8	ξ
$pred[i]$	ξ	ξ	3	6	7	ξ	ξ	0
$dir[i]$	1	1	-1	1	1	1	1	0
$depth[i]$	1	1	2	3	2	1	1	0

Для орграфа \widehat{G} вычислим значения $a_i, i \in \bar{V}$ и запишем систему уравнений баланса:

$$\begin{aligned}
 x_{\xi,2} &= \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,8} + p_{2,9})}{p_{2,9}} - \frac{f_{1,9}p_{1,2}}{p_{1,9}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}} - \frac{f_{8,9}p_{8,2}}{p_{8,9}} - f_{9,2}, \\
 x_{\xi,8} &= \frac{f_{8,9}(p_{8,2} + p_{8,3} + p_{8,9})}{p_{8,9}} - \frac{f_{2,9}p_{2,8}}{p_{2,9}} - \frac{f_{3,9}p_{3,8}}{p_{3,9}} - f_{9,8}, \\
 x_{\xi,3} + x_{4,3} &= \frac{f_{3,9}(p_{3,4} + p_{3,8} + p_{3,9})}{p_{3,9}} - \frac{f_{8,9}p_{8,3}}{p_{8,9}} - f_{9,3}, \\
 -x_{4,3} - x_{4,6} - x_{4,7} + x_{6,4} + x_{7,4} &= -\frac{f_{3,9}p_{3,4}}{p_{3,9}}, \\
 x_{\xi,7} + x_{4,7} + x_{6,7} - x_{7,4} - x_{7,6} &= 0, \\
 x_{4,6} - x_{6,4} - x_{6,5} - x_{6,7} + x_{7,6} &= -\frac{f_{5,9}p_{5,6}}{p_{5,9}}, \\
 x_{6,5} &= \frac{f_{5,9}(p_{5,2} + p_{5,6} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - f_{9,5}, \\
 -x_{\xi,2} - x_{\xi,3} - x_{\xi,7} - x_{\xi,8} &= -f_{1,9} - f_{2,9} - f_{3,9} - f_{5,9} - f_{8,9} + f_{9,1} + f_{9,2} + \\
 &\quad + f_{9,3} + f_{9,5} + f_{9,8}.
 \end{aligned} \tag{2.10}$$

Для построения частного решения системы 2.10 допустим, что все неопорные потоки равны 0.

$$\begin{aligned}
\tilde{x}_{4,6} &\rightarrow 0, \quad \tilde{x}_{4,7} \rightarrow 0, \quad \tilde{x}_{6,4} \rightarrow 0, \quad \tilde{x}_{6,7} \rightarrow 0, \quad \tilde{x}_{7,4} \rightarrow 0, \quad \tilde{x}_{4,3} \rightarrow \frac{f_{3,9}p_{3,4}}{p_{3,9}}, \\
\tilde{x}_{6,5} &\rightarrow \frac{f_{5,9}(p_{5,2} + p_{5,6} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - f_{9,5}, \\
\tilde{x}_{7,6} &\rightarrow \frac{f_{5,9}(p_{5,2} + p_{5,6} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \frac{f_{5,9}p_{5,6}}{p_{5,9}} - f_{9,5}, \\
\tilde{x}_{9,2} &\rightarrow \frac{f_{2,9}(p_{2,1} + p_{2,5} + p_{2,8} + p_{2,9})}{p_{2,9}} - \frac{f_{1,9}p_{1,2}}{p_{1,9}} - \frac{f_{5,9}p_{5,2}}{p_{5,9}} - \frac{f_{8,9}p_{8,2}}{p_{8,9}} - f_{9,2}, \\
\tilde{x}_{9,3} &\rightarrow \frac{f_{3,9}(p_{3,4} + p_{3,8} + p_{3,9})}{p_{3,9}} - \frac{f_{3,9}p_{3,4}}{p_{3,9}} - \frac{f_{8,9}p_{8,3}}{p_{8,9}} - f_{9,3}, \\
\tilde{x}_{9,7} &\rightarrow \frac{f_{5,9}(p_{5,2} + p_{5,6} + p_{5,9})}{p_{5,9}} - \frac{f_{2,9}p_{2,5}}{p_{2,9}} - \frac{f_{5,9}p_{5,6}}{p_{5,9}} - f_{9,5}, \\
\tilde{x}_{9,8} &\rightarrow \frac{f_{8,9}(p_{8,2} + p_{8,3} + p_{8,9})}{p_{8,9}} - \frac{f_{2,9}p_{2,8}}{p_{2,9}} - \frac{f_{3,9}p_{3,8}}{p_{3,9}} - f_{9,8}.
\end{aligned}$$

Характеристические векторы $\delta(\tau, \rho) = (\delta_{ij}^{\tau\rho}, (i, j) \in E), (\tau, \rho) \in E \setminus E_0$ приведены в таблице 2.7.

Таблица 2.7 – Характеристические векторы

	$\delta_{4,3}$	$\delta_{4,7}$	$\delta_{7,4}$	$\delta_{7,6}$	$\delta_{6,7}$	$\delta_{6,4}$	$\delta_{4,6}$	$\delta_{6,5}$	$\delta_{\xi,2}$	$\delta_{\xi,3}$	$\delta_{\xi,7}$	$\delta_{\xi,8}$
(4,7)	-1	1	0	0	0	0	0	0	0	1	-1	0
(7,4)	1	0	1	0	0	0	0	0	0	-1	1	0
(6,7)	0	0	0	1	1	0	0	0	0	0	0	0
(6,4)	1	0	0	1	0	1	0	0	0	-1	1	0
(4,6)	-1	0	0	-1	0	0	1	0	0	1	-1	0

Систему дополнительных уравнений можно представить следующим образом:

$$\begin{aligned}
x_{4,3} - \frac{p_{4,7}x_{4,7}}{p_{4,3}} = 0, \quad x_{4,3} - \frac{p_{4,6}x_{4,6}}{p_{4,3}} = 0, \\
x_{6,7} - \frac{p_{6,4}x_{6,4}}{p_{6,7}} = 0, \quad x_{6,7} - \frac{p_{6,5}x_{6,5}}{p_{6,7}} = 0, \\
x_{7,4} - \frac{p_{7,6}x_{7,6}}{p_{7,4}} = 0.
\end{aligned} \tag{2.11}$$

Построим на основе системы дополнительных уравнений 2.11 матрицу Λ .

$$\Lambda = \begin{pmatrix} -\frac{p_{4,7}}{p_{4,3}} - 1 & 1 & 0 & 1 & -1 \\ -1 & 1 & 0 & 1 & -\frac{p_{4,6}}{p_{4,3}} - 1 \\ 0 & 0 & 1 & -\frac{p_{6,4}}{p_{6,7}} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -\frac{p_{7,6}}{p_{7,4}} & -\frac{p_{7,6}}{p_{7,4}} & \frac{p_{7,6}}{p_{7,4}} \end{pmatrix}.$$

Равные значения ранга матрицы Λ и числа неопорных дуг являются доказательством единственности решения системы. Рассмотрим множества $\widehat{E}_W = \widehat{E}_{-T}$, тогда $\widehat{E}_{-T} \setminus \widehat{E}_W = \emptyset$. Следовательно, матрицы Λ_W и Λ совпадают.

Произведем вычисления определителя матрицы Λ_W

$$\det(\Lambda_W) = -\frac{p_{6,4}(p_{4,6}p_{4,7}p_{7,4} + p_{4,3}(p_{4,6}p_{7,4} + p_{4,7}(p_{7,4} + p_{7,6})))}{p_{4,3}^2 p_{6,7} p_{7,4}}.$$

Учитывая положительное значение p_{ij} , все слагаемые определителя являются строго положительными, из чего следует, что $\det(\Lambda_W)$ не равен 0, поэтому матрица Λ_W является невырожденной. Описанными выше правилами мы можем воспользоваться для решения систем 2.10, 2.11. Таким образом показано, что оптимальное решение задачи расположения сенсоров для графа G , изображенного на рисунке 2.1, достигается при одном обзореваемом узле $M = \{5\}$.

ГЛАВА 3

ИССЛЕДОВАНИЕ ПРОЦЕССОВ ОЦЕНКИ НЕОДНОРОДНОГО ПОТОКА В МУЛЬТИСЕТЯХ

3.1 Моделирование процессов оценки мультипотока

С целью моделирования оценки мультипотока изучим конечный связный ориентированный мультиграф (мультисеть) $G = (V, E)$ с множеством узлов V и множеством мультидуг E , определенных на декартовом произведении $V \times V$, при этом $|V| < \infty, |E| < \infty$.

Для построения мультиграфа введем условные обозначения. Так, множество, включающее r типов потока в мультисети, обозначим через $K = \{1, 2, \dots, r\}$. Для обозначения связной сети, которая соответствует типу потока $k \in K$, будем использовать $G^k = (V^k, E^k)$, где $V^k \in V$. Обозначим через E^k множество дуг для потока типа k , где $k \in K$.

Предположим, что для любого значения k из множества K граф $G^k = (V^k, E^k)$ является двунаправленным. Из этого следует, что если существует дуга $(i, j)^k \in E^k$, то существует и дуга $(j, i)^k \in E^k$.

Функцией сетевого потока x представим трафик в мультисети $G = (V, E)$. Рассмотрим следующую систему линейных алгебраических уравнений:

$$\sum_{j \in V_i^+(E^k)} x_{ij}^k - \sum_{j \in V_i^-(E^k)} x_{ji}^k = \begin{cases} x_i^k & i \in V_k^*, \\ 0 & i \in V^k \setminus V_k^*, k \in K, \end{cases} \quad (3.1)$$

где V_k^* – множество узлов с неизвестным внешним потоком x_i^k в узле $i \in V_k^*, V_k^* \subseteq V^k$.

При условии, что множество V_k^* не пусто, ранг матрицы системы линейных уравнений 3.1 для графа связности $G^k = (V^k, E^k)$ будет равняться $|V^k|$ для всех k из K . [10, 11] Если же предположить, что множество V_k^* пусто, то ранг матрицы системы 3.1 равен $|V^k - 1|, k \in K$ [6].

Для получения информации о неизвестных дуговых потоках x_{ij}^k и внешних потоках x_i^k узлов $i \in V_k^*, k \in K$ установим сенсоры в узлах мультиграфа G . Такие узлы назовем обозреваемыми и обозначим их множество через $M = \bigcup_{k \in K} M_k$, где M_k – множество обозреваемых узлов для потока K . Допустим возможность пустоты множества M_k .

Дуговые потоки, проходящие через узел i множества V условно обозначим $K(i)$. Если предположить, что узел i является обозреваемым, то будут известны значения потоков на всех исходящих и входящих дугах для узла i множества M_k . Обозреваемый узел i позволяет определить значения внешних потоков в узлах $i \in M \cap V_k^* \forall i \in M, \forall k \in K(i)$:

$$x_{ij}^k = f_{ij}^k, j \in V_i^+(E^k), x_{ji}^k = f_{ji}^k, j \in V_i^-(E^k), x_i^k = f_i^k, \quad (3.2)$$

$$i \in M \cap V_k^* \neq \emptyset \quad \forall k \in K(i) \quad \forall i \in M,$$

где $f_{ij}^k, f_{ji}^k, f_i^k$ – константы.

Для точного расчета объема потока по каждой исходящей дуге будем учитывать информацию о коэффициентах разбиения потока p_{ij}^k , где $0 < p_{ij}^k \leq 1, (i, j) \in E^k$. Эта информация дает возможность выразить общий исходящий из узла i поток $F(i) = \sum_{j \in V_i^+(E^k)} x_{ij}^k$ как функцию от объема потока по каждой исходящей дуге.

Чтобы рассчитать взаимосвязь дуговых потоков, используем известные коэффициенты разбиения p_{ij}^k и составим систему дополнительных уравнений. Допустим, что для узла $i \in V$ выполняется соотношение $|V_i^+(E^k)| \geq 2$, значит, для любой дуги, которая выходит из узла i , дуговой поток может быть выражен через дуговой поток x_{i,ν_i}^k следующим образом:

$$x_{ij}^k = \frac{p_{ij}^k}{p_{i,\nu_i}^k} x_{i,\nu_i}^k, \quad j \in V_i^+(E^k) \setminus \{\nu_i\}, \quad |V_i^+(E^k)| \geq 2, \quad k \in K(i), \quad (3.3)$$

где $(i, \nu_i)^k$ – каноническая дуга, исходящая из узла i . Обозначим $\beta_{ij}^k = \frac{p_{ij}^k}{p_{i,\nu_i}^k}$, если $|V_i^+(E^k)| \geq 2$, и $\beta_{ij}^k = p_{(i, \nu_i)^k}$, если $|V_i^+(E^k)| = 1$.

Специальные программируемые устройства, установленные в узлах мультиграфа G , предоставляют сведения, которые позволяют определить численные значения x_{ij}^k дуговых потоков с помощью коэффициентов разбиения p_{ij}^k, p_{i,ν_i}^k . Для этого в уравнение системы 3.1 подставим численные значения дуговых потоков, полученные из системы 3.3. Известные значения дуговых и внешних потоков $x_i^k = f_i^k, k \in K(i), i \in M$ позволяют удалить из мультиграфа $G = (V, E)$ соответствующие дуги и узлы. По этому же принципу изменим систему уравнений 3.1 (удалим уравнения, которые содержат известные дуговые и внешние потоки), обозначим число уравнений с неизвестными дуговыми потоками через q . Полученная система имеет следующий вид:

$$\begin{aligned} \sum_{j \in V_i^+(\bar{E}^k)} x_{ij}^k - \sum_{j \in V_i^-(\bar{E}^k)} x_{ij}^k &= \begin{cases} a_i^k + x_i^k & i \in V_k^*, \\ a_i^k & i \in V^k \setminus V_k^*, k \in K, \end{cases} \\ \sum_{(i,j) \in \bar{E}} \lambda_{ij}^{kp} x_{ij}^k &= 0, \quad p \in P = \{1, \dots, q\}, \quad P \neq \emptyset \end{aligned} \quad (3.4)$$

где \bar{V}_k^* – множество узлов с неизвестным внешним потоком x_i^k в мультисети \bar{G} .

В результате названных изменений мультиграфа G получаем новый мультиграф $\bar{G} = (\bar{V}, \bar{E})$.

Сформулируем задачу размещения сенсоров в узлах мультиграфа: для оптимизации системы контроля за потоками установить минимальное

количество обозреваемых узлов M , при котором система (3.4) допускает единственное решение, и обозначить узлы мультисети G для размещения сенсоров.

3.2 Примеры локализации сенсоров в узлах мультиграфа

Пример 1. Рассмотрим мультиграф G (см. рисунок 3.1). Пусть множество обозреваемых узлов $M = \{5\}$, множество типов потоков для обозреваемого узла равно $K(5) = \{2, 3\}$.

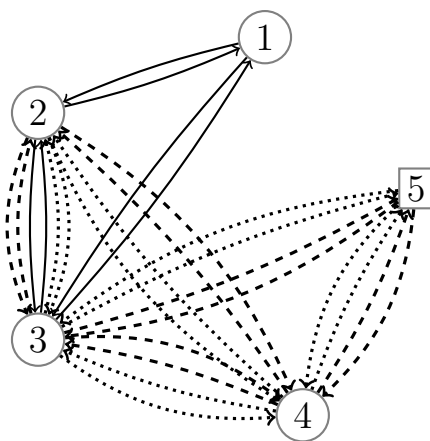


Рисунок 3.1 – Мультиграф $G = (V, E)$. Узел $M = \{5\}$ является обозреваемым

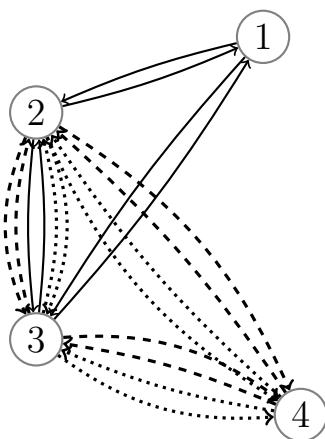


Рисунок 3.2 – Мультиграф $G' = (V', E')$ после преобразования (3.2)

Поток типа $k = 1$ не входит в множество $K(5)$. Система (3.4) для $k = 1$ является недоопределенной.

Пример 2. Рассмотрим мультиграф G на рисунке 3.3. Пусть для него множество обозреваемых узлов равно $M = \{3\}$.

При установке сенсора в обозреваемый узел 3, для которого множество типов потоков $K(3) = \{1, 2, 3\}$, дальнейшем использовании информации (3.2) и удалении соответствующих дуг получим новый мультиграф, представленный на рисунке 3.4

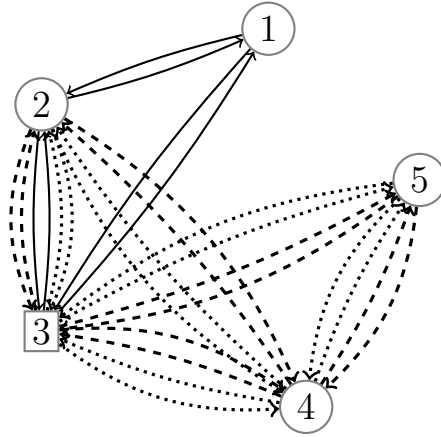


Рисунок 3.3 – Мультиграф $G = (V, E)$. Узел $M = \{3\}$ является обозреваемым

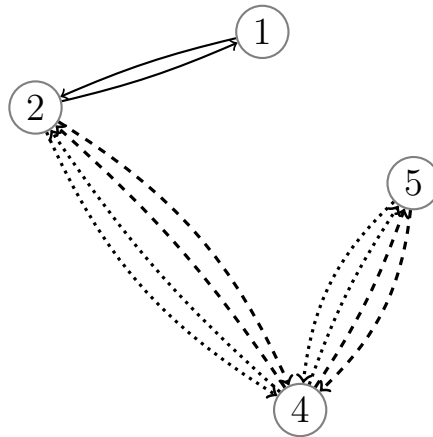


Рисунок 3.4 – Мультиграф $G' = (V', E')$ после преобразования (3.2)

Для узла $M = \{3\}$ система (3.4) для графа \bar{G} с неизвестными дуговыми и внешними потоками имеет единственное решение.

В примере 1 система является недоопределенной для множества обозреваемых узлов $M = \{5\}$. Это означает, что для $M = \{5\}$ недостаточно собранной информации для единственности решения системы (3.4). В случае $M = \{3\}$ можно найти единственное решение системы (3.4).

ГЛАВА 4

ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧИ

4.1 Генерация данных и решения задачи оценки потоков на ненаблюдаемой части сети

В программной среде Shiny на основе языка программирования R было разработано приложение для генерации, отображения графа и решения задачи оценки потоков. Для задания графа G необходимо установить количество вершин $|V|$, количество ребер $|E|$ и $|V^*|$ – количество узлов с переменным внешним потоком. Код генерации графа приведен в приложении А.

После нажатия на кнопку генерации на странице приложения можно увидеть иллюстрацию полученного графа (рис. 4.1). На рисунке узлы с переменным внешним потоком помечены красным цветом.

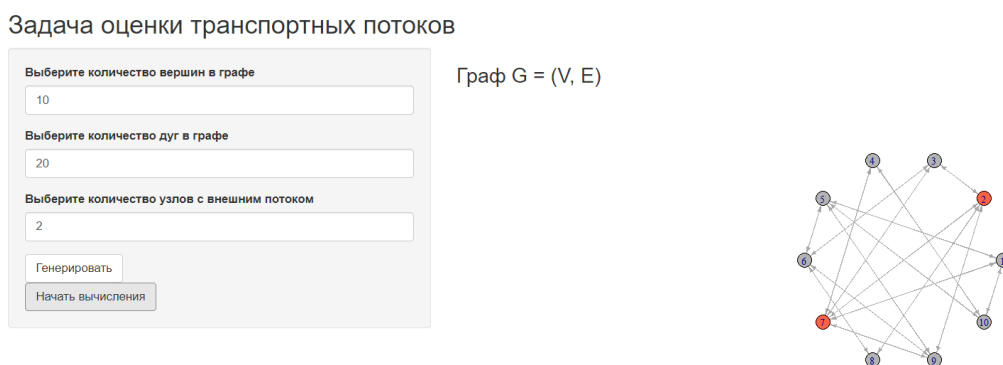


Рисунок 4.1 – Скриншот программы с иллюстрацией полученного графа G

После запуска вычислений на экране можно увидеть значения множеств M , M^* , M^+ , ..., иллюстрацию графа G' - ненаблюдаемую часть графа G (рис. 4.2).

Следующий этап вычислений - подсчет корневых структур для хранения в памяти компьютера леса корневых деревьев. (рис. 4.3)

Код программы находится в приложениях Б, В.

4.2 Сравнение матричного метода и метода декомпозиции для решения задачи оценки потоков в сетях

Матричный метод находит единственное оптимальное решение задачи оценки потоков в сетях путем построения усеченной модифицированной матрицы инцидентности исходного графа, данная матрица дополняется столбцами с учетом узлов с внешним потоком, а также строками с учетом

Множества обозреваемых узлов M , $M^*=V(CC(M))$

M : 1
 M^* : 5, 7, 10
 M^* : 1, 5, 7, 10

Граф G' - ненаблюдаемая часть графа G

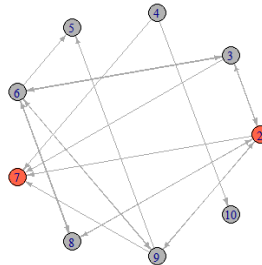


Рисунок 4.2 – Скриншот программы с иллюстрацией ненаблюдаемой части графа G

Корневые структуры для хранения в памяти компьютера леса корневых деревьев

	1	2	3	4	5	6	7	8	9	10
pred[i]	0	0	2	7	6	3	0	2	2	4
dir[i]	0	0	1	-1	1	1	0	1	1	1
depth[i]	0	0	1	1	3	2	0	1	1	2

Лес корневых деревьев с корнями в узлах 2, 7

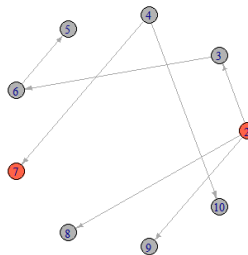


Рисунок 4.3 – Скриншот программы с иллюстрацией таблицы корневых структур и леса корневых деревьев

данных от наблюдаемых узлов о дуговых потоках на входящих в них дугах графа. Производится решение системы с обновленной матрицей значений потоков.

В таблице 4.1 представлены численные результаты построения оптимального решения с использованием матричного метода и методов декомпозиции.

Численные результаты решения задачи оценки потоков на ненаблюдаемой части двунаправленной сети показали практическую применимость методов декомпозиции.

В [7] на реальных данных получены результаты построения субоптимальных решений исследуемой задачи оценки множества M обозреваемых узлов двунаправленной сети G , а также выполнен анализ численной устойчивости решений разреженных систем линейных алгебраических уравнений с матрицами неполного/полного рангов.

$$x_{(2,9)} = -f_{(5,1)} * p_{(5,9)} / p_{(5,1)} - f_{(7,1)} * p_{(7,9)} / p_{(7,1)}$$

$$x_{(6,5)} = -f_{(1,5)} + f_{(5,1)} + f_{(5,1)} * p_{(5,6)} / p_{(5,1)} + f_{(5,1)} * p_{(5,9)} / p_{(5,1)} + f_{(5,1)} * p_{(5,10)} / p_{(5,1)} - f_{(10,1)} * p_{(10,5)} / p_{(10,1)}$$

$$x_{(3,6)} = -f_{(5,1)} * p_{(5,6)} / p_{(5,1)}$$

$$x_{(2,3)} = -f_{(7,1)} * p_{(7,3)} / p_{(7,1)}$$

$$x_{(2)} = -f_{(7,1)} * p_{(7,2)} / p_{(7,1)}$$

Характеристические векторы

x	delta_{(2, 7)}	delta_{(3, 2)}	delta_{(3, 7)}	delta_{(6, 3)}	delta_{(6, 8)}	delta_{(6, 9)}	delta_{(8, 2)}	delta_{(8, 6)}	delta_{(9, 2)}	c
x_{(2,3)}	0	1	1	0	1	1	0	-1	0	
x_{(2,8)}	0	0	0	0	-1	0	1	1	0	
x_{(2,9)}	0	0	0	0	0	-1	0	0	1	
x_{(3,2)}	0	1	0	0	0	0	0	0	0	
x_{(3,6)}	0	0	0	1	1	1	0	-1	0	
x_{(3,7)}	0	0	1	0	0	0	0	0	0	
x_{(4,7)}	0	0	0	0	0	0	0	0	0	
x_{(4,10)}	0	0	0	0	0	0	0	0	0	
x_{(6,3)}	0	0	0	1	0	0	0	0	0	
x_{(6,5)}	0	0	0	0	0	0	0	0	0	

1-10 of 19 rows

Previous 1 2 Next

Рисунок 4.4 – Скриншот программы с иллюстрацией частного решения и таблицы характеристических векторов

Таблица 4.1 – Численные результаты

$ V $	$ E $	$ V^* $	t_1, c	t_2, c
50	100	8	4.88453	4.28
75	300	16	117.496	13.34
100	300	20	121.02	22.25
200	800	48	2215.2	215.32
1000	4000	64	5391.43	1552.32

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была рассмотрена задача расположения специальных программируемых устройств (сенсоров) и задача оценки транспортных потоков.

- Определена теоретическая основа построения опоры сети разреженной линейной системы. Проанализированы разреженные недоопределенные системы в задачах оптимального расположения специальных программируемых устройств.
- Смоделирован процесс идентификации обзриваемых узлов графа.
- Смоделирован процесс оценки мультипотока и рассмотрен пример локализации специальных программируемых устройств в узлах мультиграфа.
- Определены теоретическая и технологическая составляющие, обеспечивающие комплексное решение задач оптимального размещения сенсоров для оценки однородного потока в двунаправленной сети.
- Приведен пример построения оптимального решения задачи оценки потока на ненаблюдаемой части сети с минимальной мощностью множества обзриваемых узлов.
- На языке программирования R реализован алгоритм декомпозиции с применением современных технологий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Пилипчук Л. А. Оптимальные пути: алгоритмические, структурные и технологические решения / Л. А. Пилипчук, А. С. Пилипчук, Е. Н. Полячок // Весн. Гр. дзярж. ун-та імя Я. Купалы. Сер. 2, Матэматыка. Фізіка. Інфарматыка, вылічальная тэхніка і кіраванне. – 2020. №3. – Р. 144–151.
2. Пилипчук Л. А. Линейные неоднородные задачи потокового программирования: учеб.-метод. пособие / Л. А. Пилипчук. – Минск: БГУ, 2009. – 222 с.
3. Пилипчук Л. А. Исследование процессов оценки неоднородного потока в мультисетях / Л. А. Пилипчук, М. П. Романчук // Информационные системы и технологии = Information Systems and Technologies : материалы междунар. науч. конгресса по информатике. В 3 ч. Ч. 1, Респ. Беларусь, Минск, 27–28 окт. 2022 г. / Белорус. гос. ун-т ; редкол.: С. В. Абламейко (гл. ред.) [и др.]. – Минск : БГУ, 2022. – С. 243–248.
4. Пилипчук Л. А. О проблеме оценки однородного потока в двунаправленной сети / Л. А. Пилипчук, М. П. Романчук // Информационные технологии и системы 2021 (ИТС 2021)=Information Technologies and Systems 2021 (ITS 2021): материалы междунар. науч. конф. (Республика Беларусь, Минск, 24 ноября 2021) –Минск: БГУИР, 2021. – С. 158.
5. Bianco L. A network based model for traffic sensor location with implication in O/D matrix estimates. / Bianco L., Confessore G., Reverberi P. // Transportation Science. – 2001. №35. – Р. 50–60.
6. Pilipchuk L. A. Sparse Linear Systems and Their Applications / L. A. Pilipchuk. – Minsk: BSU, 2013. – 236 p.
7. Пилипчук, Л.А. Идентификация сенсорной конфигурации и управление потоками / Л. А. Пилипчук, А. С. Пилипчук, Е. Н. Полячок, А. И. Фаразей // Журнал Белорусского государственного университета. Математика. Информатика = Journal of the Belarusian State University. Mathematics and Informatics. – 2018. – № 2. – С. 67-76.
8. Bianco L. Combinatorial aspects of the sensor location problem / L. Bianco, G. Confessore, M. Gentili // Annals of Operation Research. – 2006. №144. – Р. 201–234.
9. Йенсен П. Потокное программирование. Пер. с англ. / П. Йенсен, Д. Барнес. – Москва: Радис и связь, 1984. – 392 с.
10. Пилипчук Л. А. О методах декомпозиции разреженных недоопределенных систем с матрицами полного и неполного ранга / Л. А. Пилипчук // Известия Гомельского государственного университета имени Ф. Скорины. – 2016. №6 (99). – С. 87–91.

11. Pilipchuk L. A. The general solutions of sparse systems with rectangular matrices in the problem of sensors optimal location in the nodes of a generalized graph / L. A. Pilipchuk, O. V. German, A. S. Pilipchuk // Вестник БГУ. Серия 1, Физика. Математика. Информатика. – 2015. №2. – P. 91–96.
12. Пилипчук Л. А. Реализация методов декомпозиции в задаче оценки потока на ненаблюдаемой части двунаправленной сети / Л. А. Пилипчук, М. П. Романчук // Труды 10-го международного научного семинара АМАДЕ-2021, 13–17 сентября 2021 г., Минск, Беларусь, БГУ. – Минск : ИВЦ Минфина, 2022. – С. 65-77.
13. Pilipchuk L. A. Algorithms of Solving Large Sparse Underdetermined Linear Systems with Embedded Network Structure / L. A. Pilipchuk, Y. V. Malakhouskaya, D. R. Kincaid, M. Lai // East-West J. of Mathematics. – 2002. №2. – P. 191–201.
14. Gentile M. Locating active sensors on traffic networks / M. Gentile, P. Mirchandani // Annals of Operation Research. – 2006. №1. – P. 201–234.
15. Пилипчук Л. А. Разреженные недоопределенные системы линейных алгебраических уравнений / Л. А. Пилипчук. – Минск: БГУ, 2012. – 260 с.
16. Bianco L. Locating sensors to observe network arc flows: exact and heuristic approaches / L. Bianco, C. Cerrone, R. Cerulli, M. Gentili // Computers and Operation Research. – 2014. №46, – P. 12–22.
17. Габасов Р. Методы линейного программирования: в 3 частях. Ч. 3: Специальные задачи / Р. Габасов, Ф. М. Кириллова. – Минск – 1980. – 368 с.
18. Ahuja R. K. Networks flows: Theory, Algorithms and Applications / R. K. Ahuja, T. L. Magnanti, J. B. Orlin. – New Jersey. –1993. – 864 p.
19. Оре О. Теория графов / О. Оре. – Москва: Наука. –1980. – 336 с.

Код для генерации графа

```

library(magrittr)
library(rlist)
library(data.table)

generate_graph <- function(n, a) {
  # list of all edges
  edges <- seq(1, n)
  u <- list()
  for (i in 1:n) {
    if (length(edges) == 0) {
      break
    }
    nodes <- unlist(u) %>% unique()
    # random choice of element from edges
    if (length(edges) == 1) {
      to <- edges
    } else {
      to <- sample(edges, 1)
    }

    from <- ifelse(length(nodes) > 0, sample(nodes, 1), sample(edges, 1))
    u <- list.append(u, c(from, to))

    if (length(nodes) == 0) {
      # delete 'from' from edges
      edges <- edges[!edges == from]
    }
    # delete 'to' from edges
    edges <- edges[!edges == to]
  }

  for (i in 1:(a - n)) {
    u <- list.append(u, sample(seq(10), 2))
  }

  full_u <- u
  for (i in 1:length(u)) {
    full_u <- list.append(full_u, rev(u[[i]]))
  }
  return(full_u %>% unique())
}

generate <- function(v_num, u_num, dynamic_num) {
  graph <- generate_graph(v_num, u_num)
  list_2d <- sapply(graph, cbind)
  graph_table <- data.frame(t(list_2d))
  colnames(graph_table) <- c("x", "y")
}

```

```
graph_table <- graph_table[order(graph_table$x, graph_table$y),] %>%  
setDT()  
  
nodes <- unlist(graph) %>% unique() %>% sort()  
dynamic <- sample(nodes, dynamic_num)  
sensors <- c(1)  
  
return(list(v=nodes, u=graph_table, dyn=dynamic, s=sensors))  
}
```

Код необходимых для вычислений функций

```

library(magrittr)
library(rlist)
library(matlib)
library(data.table)

generate_sets <- function(E, I, S, M) {
  IM <- setdiff(I, M)

  IX <- list()
  for (i in IM) {
    temp <- setdiff(unique(c(E[x == i]$x, E[x == i]$y)), M)
    if (length(temp) == 1 & temp[1] == i) {
      IX <- list.append(IX, i)
    }
  }

  M_star <- c(E[x %in% M]$x, E[x %in% M]$y) %>%
unique() %>% setdiff(IX) %>% sort()

  SM <- setdiff(S, M) %>% sort()
  M_plus <- setdiff(M_star, M) %>% sort()
  return(list(IM=IM, IX=IX, M_star=M_star, M_plus=M_plus, SM=SM))
}

get_unobserved_part <- function(E, M, M_plus) {
  unobserved <- E[!(x %in% M | y %in% M)]
  unobserved <- unobserved[!x %in% M_plus]
  return(unobserved)
}

construct_helper <- function(E, ch, par, dp, mark, pred, depth, dir, thread) {
  child <- c(E[x == ch & mark[y] != 1]$y, E[y == ch & mark[x] != 1]$x) %>%
unique()

  mark[ch] <- 1
  thread <- c(ch, thread)
  pred[ch] <- par
  depth[ch] <- dp

  flag <- ifelse(nrow(E[x == par & y == ch]) == 0, FALSE, TRUE)
  dir[ch] <- ifelse(flag, 1, ifelse(par != 0, -1, 0))
  dp <- dp + 1

  if (length(child) != 0) {
    for (k in child) {
      res <- construct_helper(E, k, ch, dp, mark, pred, depth, dir, thread)
      thread <- res$thread
      mark <- res$mark
    }
  }
}

```

```

    pred <- res$pred
    depth <- res$depth
    dir <- res$dir
  }
}
return(list(thread=thread, mark=mark, pred=pred, depth=depth, dir=dir))
}

generate_structures <- function(E, M, SM, V_dynamic, nV) {
  mark <- rep(0, nV)
  mark <- replace(mark, SM, 1)

  pred <- rep(0, nV)
  depth <- rep(0, nV)
  dir <- rep(0, nV)
  thread <- M
  sign <- rep(0, nV)
  sign <- replace(sign, V_dynamic, 1)

  for (j in SM) {
    res <- construct_helper(E, j, 0, 0, mark, pred, depth, dir, thread)
    thread <- res$thread
    mark <- res$mark
    pred <- res$pred
    depth <- res$depth
    dir <- res$dir
  }
  return(list(pred=pred, thread=thread, depth=depth, dir=dir, sign=sign))
}

construct_tree_and_characteristic <- function(M, E, pred, dir) {
  tr <- data.table(x=c(0), y=c(0))

  for (i in 1:length(pred)) {
    if (pred[i] != 0) {
      if (dir[i] == 1) {
        new_edge <- list(pred[i], i)
      }else{
        new_edge <- list(i, pred[i])
      }
      tr <- rbind(tr, new_edge)
    }else{
      if(!i %in% M) {
        tr <- rbind(tr, list(i, 0))
      }
    }
  }
  tr <- tr[-1,]
  not_tr <- E[, .(x, y)][!tr, on=c("x", "y")]
  return(list(tree=tr, cvv=not_tr))
}

```

```

characteristic_vectors <- function(delta, m, n, pred, dir, depth, sign, l) {
  deleted <- delta[x == m & y == n & value == 0]
  delta <- delta[!(x == m & y == n & value == 0)]
  deleted$value <- 1
  delta <- rbind(delta, deleted)

  q <- 0
  v <- n
  u <- m
  ff <- -1
  v1 <- n
  u1 <- m
  if (depth[m] < depth[n]) {
    q <- depth[n] - depth[m]
  }
  if (depth[m] > depth[n]) {
    q <- depth[m] - depth[n]
    ff <- 1
    v <- m
    u <- n
    v1 <- m
    u1 <- n
  }
  if (q != 0) {
    for (i in 1:q) {
      if (v > 0) {
        if (dir[v] == 1) {
          delta <- swap(delta, pred[v], v, dir[v], ff, l)
        }
        if (dir[v] == -1) {
          delta <- swap(delta, v, pred[v], dir[v], ff, l)
        }
        v <- pred[v]
      }
    }
  }

  while (v != u) {
    if (v > 0) {
      if (dir[v] == 1) {
        delta <- swap(delta, pred[v], v, dir[v], ff, l)
      }
      if (dir[v] == -1) {
        delta <- swap(delta, v, pred[v], dir[v], ff, l)
      }
    }
    if (u > 0) {
      if (dir[u] == 1) {
        delta <- swap(delta, pred[u], u, dir[u], -ff, l)
      }
      if (dir[u] == -1) {
        delta <- swap(delta, u, pred[u], dir[u], -ff, l)
      }
    }
  }
}

```

```

    }
  }
  if (pred[v1] != 0) {
    if (dir[pred[v1]] != 0) {
      v1 <- v
    }
  }
  v <- pred[v]
  if (pred[u1] != 0) {
    if (dir[pred[u1]] != 0) {
      u1 <- u
    }
  }
  u <- pred[u]

  if (identical(v, numeric(0)) | identical(u, numeric(0))) {
    return(delta)
  }
}

if (u == 0 & v == 0) {
  delta <- end_node(delta, u1, pred, dir, sign, -ff)
  delta <- end_node(delta, v1, pred, dir, sign, ff)
}

return(delta)
}

swap <- function(delta, i, j, d, f, l) {
  deleted <- delta[x == i & y == j & value == 0]
  delta <- delta[!(x == i & y == j & value == 0)]
  deleted$value <- d * f
  delta <- rbind(delta, deleted)

  return(delta)
}

end_node <- function(delta, u, pred, dir, sign, f) {
  if (dir[u] == 1) {
    deleted <- delta[x == pred[u] & y == 0 & value == 0]
    delta <- delta[!(x == pred[u] & y == 0 & value == 0)]
    deleted$value <- sign[pred[u]] * delta[x == pred[u] & y == u]$value
    delta <- rbind(delta, deleted)
  }
  if (dir[u] == -1) {
    deleted <- delta[x == pred[u] & y == 0 & value == 0]
    delta <- delta[!(x == pred[u] & y == 0 & value == 0)]
    deleted$value <- -1 * sign[pred[u]] * delta[x == u & y == pred[u]]$value
    delta <- rbind(delta, deleted)
  }
  if (dir[u] == 0) {
    deleted <- delta[x == u & y == 0 & value == 0]

```

```

    delta <- delta[!(x == pred[u] & y == 0 & value == 0)]
    deleted$value <- f * sign[u]
    delta <- rbind(delta, deleted)
  }
  return(delta)
}

generate_system_first <- function(E, IM_IX, V, V_dynamic) {
  eq <- matrix(ncol = nrow(E))
  for (i in IM_IX) {
    temp <- rep(0, nrow(E))
    for (j in 1:nrow(E)) {
      if (E[j, ]$y == i) {
        temp[j] <- -1
      }
      if (E[j, ]$x == i) {
        temp[j] <- 1
      }
    }
    eq <- rbind(eq, temp)
  }
  colnames(eq) <- E$formula
  eq <- eq[-1,]

  b <- data.table(coeff=rep(0, length(V)), index=seq(1, length(V)), formula="")
  b$coeff <- replace(b$coeff, V_dynamic, 1)
  b[, formula := ifelse(coeff != 0, paste0("x_", index), formula)]
  return(list(system_matrix=eq, system_vector=b))
}

generate_system_second <- function(E, M, vars) {
  eq <- E[x %in% M | y %in% M]
  eq$formula <- paste0("f_{", eq$x, ", ", eq$y, "}")

  for (i in eq$index) {
    vars[i, ]$simplified <- paste0("f_{", E[i]$x, ", ", E[i]$y, "}")
  }
  return(list(system_matrix=eq, variables=vars))
}

generate_system_third <- function(E, IM_IX, M, vars) {
  vector_of_indexes <- rep(0, nrow(E))
  for (i in IM_IX) {
    out <- E[x == i]
    if (nrow(out) > 1) {
      j <- out[y %in% M]
      if (nrow(j) == 0) {
        v <- out[1, ]
      } else {
        v <- j[1, ]
      }
    }
    to_change <- setdiff(out, v)
  }
}

```

```

    if (nrow(j) != 0) {
      to_change <- setdiff(to_change, j)
    }
    vector_of_indexes <- replace(vector_of_indexes, to_change$index, v$index)
  }
}

eq2 <- matrix(vector_of_indexes, ncol = nrow(E))
colnames(eq2) <- E$formula
for (i in 1:length(eq2)) {
  if (eq2[i] != 0 & eq2[i] != i) {
    vars[i,]$equation2 <- paste0("p_{", E[i]$x, ",", E[i]$y, "} * ",
    vars[eq2[i ],]$simplified,
    " / p_{", E[eq2[i]]$x, ",", E[eq2[i]]$y, "}")
  }
}
return(list(system_matrix=eq2, variables=vars))
}

system_coefficient_first <- function(E, eq0, b, IM_IX, vars) {
  coeff <- rep("", length(IM_IX))
  for (i in 1:length(IM_IX)) {
    eq_formula = ""
    for (j in 1:nrow(E)) {
      if (eq0[i, j] == 1 & eq_formula == "") {
        eq_formula <- vars[j,]$simplified
        if (vars[j,]$simplified != vars[j,]$original) {
          coeff[i] <- vars[j,]$simplified
        }
      }else{
        if (eq0[i, j] == 1 & eq_formula != "") {
          eq_formula <- paste(eq_formula, "+", vars[j,]$simplified)
          if (vars[j,]$simplified != vars[j,]$original) {
            coeff[i] <- paste(coeff[i], "+", vars[j,]$simplified)
          }
        }
        if (eq0[i, j] == -1) {
          eq_formula <- paste(eq_formula, "-", vars[j,]$simplified)
          if (vars[j,]$simplified != vars[j,]$original) {
            coeff[i] <- paste(coeff[i], "-", vars[j,]$simplified)
          }
        }
      }
    }
  }
  if (b[IM_IX[i]] != 0) {
    eq_formula <- paste0(eq_formula, "= x_", IM_IX[i])
  } else{
    eq_formula <- paste(eq_formula, "= 0")
  }
}
return(coeff)
}

```

```

system_coefficient_second <- function(E, system_vars, vars, eq0, eq2, b, IM_IX) {
  coeff <- data.table(t(rep(0, length(system_vars))))
  colnames(coeff) <- system_vars
  for (i in 1:length(IM_IX)) {
    temp <- data.table(t(rep(0, length(system_vars))))
    colnames(temp) <- system_vars
    for (j in 1:nrow(E)) {
      if (eq0[i, j] != 0) {
        if (vars[j]$simplified == vars[j]$original) {
          temp[1, ][vars[j]$original] <- eq0[i, j]
        }
      }
    }

    if (b[IM_IX[i]]$coeff != 0) {
      temp[1, ][b[IM_IX[i]]$formula] <- -1
    }
    coeff <- coeff %>% rbind(temp[1,])
  }
  for (j in 1:nrow(E)) {
    if (vars[j]$equation2 == 0) {
      next
    }

    temp <- data.table(t(rep(0, length(system_vars))))
    colnames(temp) <- system_vars
    if (eq2[j] != 0 & eq2[j] != j) {
      temp[vars[j]$original] <- 1
      temp[vars[eq2[j]]$original] <- paste0("-", substring(vars[j]$equation2, 1, unlist(
        (gregexpr("'", vars[j]$equation2))[1]), " /", strsplit(vars[j]$equation2, "/")
        [[1]][2])
    }
    coeff <- coeff %>% rbind(temp[1,])
  }
  return(coeff)
}

get_partial_solution <- function(CV, pred, thread, dir, sign, coeff1) {
  partsol = CV$formula
  for (q in 1:length(thread)) {
    if (sign[thread[q]] == 0) {
      if (dir[thread[q]] == 1) {
        partsol <- append(partsol, paste0("x_{", pred[thread[q]], ", ", thread[q], "} = ",
          coeff1[thread[q]]))
      }
      if (dir[thread[q]] == -1) {
        partsol <- append(partsol, paste0("x_{", thread[q], ", ", pred[thread[q]], "} = -",
          coeff1[thread[q]]))
      }
    }
  }
  else{
    if (dir[thread[q]] == 1) {

```

```

    partsol <- append(partsol, paste0("x_{", pred[thread[q]], ",", thread[q], "} = ",
        coeff1[thread[q]]))
  }
  if (dir[thread[q]] == -1) {
    partsol <- append(partsol, paste0("x_{", thread[q], ",", pred[thread[q]], "} = -",
        coeff1[thread[q]]))
  }
  if (dir[thread[q]] == 0) {
    partsol <- append(partsol, paste0("x_{", thread[q], "} = ", coeff1[thread[q]]))
  }
}
}
return(partsol)
}

```

Код приложения Shiny

```

library(shiny)
library(igraph)
library(DT)
library(reactable)
library(dplyr)
source("decomp.R")
source("generation.R")

# Define UI for application
ui <- fluidPage(
  titlePanel(""),

  sidebarLayout(
    sidebarPanel(
      numericInput(inputId = "v_num",
        label = "",
        value = 10,
        min = 2,
        max = 100),
      numericInput(inputId = "u_num",
        label = "",
        value = 20,
        min = 2,
        max = 200),
      numericInput(inputId = "dynamic_num",
        label = "",
        value = 2,
        min = 1,
        max = 50),
      actionButton(inputId = "run_generate",
        label = "",
        tags$br(),
        actionButton(inputId = "run_count",
          label = ""
        ))
    ),

    # Show results
    mainPanel(
      htmlOutput("graph_text"),
      plotOutput("graph_plot"),
      htmlOutput("sets_text"),
      plotOutput("graph_updated_plot"),
      htmlOutput("structures_text"),
      reactableOutput("structures_table"),
      htmlOutput("tree_text"),
      plotOutput("tree_plot"),
      htmlOutput("partial_solution_text"),

```

```

htmlOutput("character_vectors_text"),
reactableOutput("character_vectors_table")
)
)
)

# Define server logic
server <- function(input, output) {
  V <- 0
  V_dynamic <- 0
  E <- 0
  M <- 0
  colrs <- c("gray70", "tomato")

  observeEvent(input$run_generate, {
    data <- generate(input$v_num, input$u_num, input$dynamic_num)
    nodes <- data.table(id=data$v, type=rep(1, input$v_num))
    nodes[id %in% data$dyn]$type <- 2

    V <<- data$v
    V_dynamic <<- data$dyn
    M <<- data$s
    E <<- data$u
    links <- data.table(from=E$x, to=E$y, weight=rep(1, nrow(E)))

    net <- graph_from_data_frame(d=links, vertices=nodes, directed=T)
    V(net)$color <- colrs[V(net)$type]

    output$graph_text <- renderUI({
      HTML("<h3> G = (V, E)</h3>")
    })
    output$graph_plot <- renderPlot({
      plot(net, edge.arrow.size=.5, layout=layout_in_circle(net))
    })
  })

  observeEvent(input$run_count, {
    nV <- length(V)

    E$formula <- paste0("x_{", E$x, ", ", E$y, "}")
    E$index <- seq(1, nrow(E))

    sets <- generate_sets(E, V, V_dynamic, M)
    IM <- sets$IM
    IX <- sets$IX
    M_star <- sets$M_star
    M_plus <- sets$M_plus
    SM <- sets$SM
    IM_IX <- setdiff(IM, IX)

    output$sets_text <- renderUI({
      str1 <- paste("<b>M: </b>", paste(M, collapse=', '))
    })
  })
}

```

```

str2 <- paste("<b>M+: </b>", paste(M_plus, collapse=', '))
str3 <- paste("<b>M*: </b>", paste(M_star, collapse=', '))
HTML(paste("<h3>
      V(CC(M))</h3>", str1, str2, str3, "<h3>  G' –
                                             M, M*=
                                             G</h3>", sep = '<br/>')
    )
})

unobserved_E <- get_unobserved_part(E, M, M_plus)
nodes <- data.table(id=IM, type=rep(1, length(IM)))
nodes[id %in% V_dynamic]$type <- 2
links <- data.table(from=unobserved_E$x, to=unobserved_E$y, weight=rep
  (1, nrow(unobserved_E)))
net <- graph_from_data_frame(d=links, vertices=nodes, directed=T)
V(net)$color <- colrs[V(net)$type]

output$graph_updated_plot <- renderPlot({
  plot(net, edge.arrow.size=.5, layout=layout_in_circle(net))
})

#
structures <- generate_structures(unobserved_E, M, SM, V_dynamic, nV)
pred <- structures$pred
thread <- structures$thread
depth <- structures$depth
dir <- structures$dir
sign <- structures$sign

structures_table <- data.frame(t(pred))
structures_table[nrow(structures_table) + 1,] <- dir
structures_table[nrow(structures_table) + 1,] <- depth
colnames(structures_table) <- V
rownames(structures_table) <- c("pred[i]", "dir[i]", "depth[i]")
output$structures_text <- renderUI({
  HTML("<h3>
      </h3>")
})
output$structures_table <- renderReactable({
  reactable(structures_table, bordered = TRUE, resizable = TRUE,
    defaultColDef=colDef(minWidth=40))
})

#
tree_and_characteristic <- construct_tree_and_characteristic(M, unobserved_
  E, pred, dir)
t <- tree_and_characteristic$tree
CV <- tree_and_characteristic$cvv

t <- t[y != 0]
links <- data.table(from=t$x, to=t$y, weight=rep(1, nrow(t)))
net_tree <- graph_from_data_frame(d=links, vertices=nodes, directed=T)

```

```

V(net_tree)$color <- colrs[V(net_tree)$type]

output$tree_text <- renderUI({
  HTML("<h3>
      ", paste(SM, collapse=', '), "</h3>")
})

output$tree_plot <- renderPlot({
  plot(net_tree, edge.arrow.size=.5, layout=layout_in_circle(net_tree))
})

# generate first part of the system
syst_1_part <- generate_system_first(data.frame(E), IM_IX, V, V_
  dynamic)
eq0 <- syst_1_part$system_matrix
b <- syst_1_part$system_vector

# variables
vars <- data.frame(E$index, E$formula, E$formula)
colnames(vars) <- c("index", "original", "simplified")

# generate second part of the system
syst_2_part <- generate_system_second(E, M, vars)
vars <- syst_2_part$variables
eq1 <- syst_2_part$system_matrix

# generate third part of the system
vars$equation2 <- rep(0, nrow(vars))
syst_3_part <- generate_system_third(E, IM_IX, M, vars)
eq2 <- syst_3_part$system_matrix
vars <- syst_3_part$variables

# change some values to 0 in vars
list_to_change <- E[x %in% eq1$x & !(index %in% eq1$index)]
for (i in 1:nrow(list_to_change)) {
  row <- list_to_change[i]
  vars[row$index, ]$simplified <- paste0("f_{", row$x, ",", eq1[x == row$x]$
    y,
    "} * p_{", row$x, ",", row$y, "} / ",
    "p_{", row$x, ",", eq1[x == row$x]$y, "}")
  vars[row$index, ]$equation2 <- 0
}

coeff1 <- system_coefficient_first(E, eq0, b, IM_IX, vars)

vars <- vars %>% setDT()
system_vars <- c(b[formula != ""]$formula, vars[simplified == original]$
  original)
system_vars_indexes <- data.frame(x=c(b[formula != ""]$index, E[index
  %in% vars[simplified == original]$index]$x),

```

```

y=c(rep(0, nrow(b[formula != ""])), E[index %in% vars[simplified == original]$
index]$y))

for (i in 1:length(coeff1)) {
  if (coeff1[i] == "") {
    coeff1[i] <- 0
  }
}
coeff1 <- c(TRUE, coeff1, rep(0, nrow(vars[equation2 != 0])))
coeff2 <- system__coefficient__second(E, system__vars, vars, eq0, eq2, b, IM__IX
)

l <- length(M) + length(coeff1) - length(pred)

lambda <- data.frame(x=c(0), y=c(0), z=c(0), value=c(0))
for (i in 1:l) {
  for (j in 1:length(coeff2[i + nV])) {
    lambda <- rbind(lambda, data.frame(x=system__vars__indexes[j, ]$x,
y=system__vars__indexes[j, ]$y,
z=i,
value=coeff2[i + nV, ][[j ]]) )
  }
}

CV[, x_formula:=paste0("x_{", x, ", ", y, "}")]
CV[, formula:=paste0(x_formula, " = 0")]
vars[original %in% CV$x_formula]$simplified <- 0

part_solution <- get__partial__solution(CV, pred, thread, dir, sign, coeff1)
output$partial__solution__text <- renderUI({
  HTML(paste("<h3>
part_solution, collapse='<br/>', sep = '<br/>'))
  "</h3>", paste(
}))

delta0 <- data.frame(x=system__vars__indexes$x,
y=system__vars__indexes$y,
formula=system__vars,
value=rep(0, length(system__vars)))

ch_vectors <- data.table()
first <- TRUE
for (q in 1:nrow(CV)) {
  vect = CV[q,]
  if (vect$y != 0) {
    if (first == TRUE) {
      temp <- characteristic_vectors(delta0 %>% setDT(), vect$x, vect$y,
pred, dir, depth, sign, l)[y != 0]
      ch_vectors$formula <- temp$formula
      ch_vectors$delta <- temp$value
      first <- FALSE
    } else {
      temp <- characteristic_vectors(delta0 %>% setDT(), vect$x, vect$y,

```

```

      pred, dir, depth, sign, l)[y != 0] %>% .[, .(value, formula)]
    ch_vectors <- left_join(ch_vectors, temp, by="formula")
  }
}
}
CV[, cols:=paste0("delta_{", x, ", ", y, "}")]
colnames(ch_vectors) <- c("x", CV[y != 0]$cols)

output$character_vectors_text <- renderUI({
  HTML("<h3>                                </h3>")
})
output$character_vectors_table <- renderReactable({
  reactable(ch_vectors, bordered = TRUE, resizable = TRUE)
})
})
}

shinyApp(ui = ui, server = server)

```
