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

Гуринович Вадим Витальевич

ТЕХНОЛОГИИ И АЛГОРИТМЫ ДЕКОМПОЗИЦИИ РАЗРЕЖЕННЫХ ЛИНЕЙНЫХ СИСТЕМ В ЗАДАЧАХ ПОТОКОВОГО ПРОГРАММИРОВАНИЯ

Дипломная работа

Научный руководитель:
кандидат физ.-мат. наук, доцент
Л. А. Пилипчук

Допущена к защите
«___» _____ 20__ г.
Заведующий кафедрой компьютерных технологий и систем
доктор педагогических наук, профессор В. В. Казаченок

Минск, 2025

ОГЛАВЛЕНИЕ

РЕФЕРАТ

Дипломная работа: 63с., 16 рис., 6 табл., 16 источников, 2 приложения.

Ключевые слова: ПОТОКОВОЕ ПРОГРАММИРОВАНИЕ, НЕДООПРЕДЕЛЕННАЯ СИСТЕМА, ГРАФ, МУЛЬТИГРАФ, МАТЕМАТИЧЕСКАЯ МОДЕЛЬ, ДЕКОМПОЗИЦИЯ, АЛГОРИТМ, МЕТОД.

Текст реферата

Объект исследования - алгоритмы и методы декомпозиции мультипотока в сетях и их практическое применение.

Цель исследования - получить эффективные методы и алгоритмы декомпозиции мультипотока в сетях, показать, что полученные математические модели могут использоваться на практике в различных сферах.

Методы исследования - изучение тематической литературы, применение утверждений линейной алгебры, применение утверждений теории алгоритмов и структур данных.

Полученные результаты - разработанные методы и алгоритмы декомпозиции мультипотока, программный модуль построения общего решения задачи потокового программирования, программный модуль демонстрирующий пример декомпозиции мультипотока.

Область возможного применения - логистика, городское планирование, медицина, управление водными ресурсами, энергетика.

РЭФЕРАТ

Дыпломная работа: 63с., 16 мал., 6 табл., 16 крыніц, 2 прыкладання.

Ключавыя словы: ПАТОКАВАЕ ПРАГРАМІРАВАННЕ, НЕДАВЫЗНАЧАНАЯ СІСТЭМА, ГРАФ, МУЛЬТЫГРАФ, МАТЭМАТЫЧНАЯ МОДЭЛЬ, ДЭКАНПАЗІЦЫЯ, АЛГАРЫТМ, МЕТАД.

Тэкст рэферата

Аб'ект даследавання - алгарытмы і метады дэканапазіцыі мультыпотоку ў сетках і іх практычнае прымяненне.

Мэта даследавання - атрымаць эфектыўныя метады і алгарытмы дэканапазіцыі мультыпотоку ў сетках, паказаць, што атрыманыя матэматычныя мадэлі могуць выкарыстоўвацца на практыцы ў розных сферах.

Метады даследавання - вывучэнне тэматычнай літаратуры, прымяненне тверджанняў лінейнай алгебры, прымяненне тверджанняў тэорыі алгарытмаў і структур дадзеных.

Атрыманыя вынікі - распрацаваныя метады і алгарытмы дэканапазіцыі мультыпотоку, праграмны модуль пабудовы агульнага рашэння задачы патокаваг а праграмавання, праграмны модуль дэманструючы прыклад дэканапазіцыі мультыпотоку.

Сфера магчымага прымянення - лагістыка, гарадское планаванне, медыцына, кіраванне воднымі рэсурсамі, энергетыка.

SUMMARY

Diploma thesis: 63 pp., 16 pics., 6 tables, 16 sources, 2 supplements.

Keywords: FLOW PROGRAMMING, UNDERDETERMINED SYSTEM, GRAPH, MULTIGRAPH, MATHEMATICAL MODEL, DECOMPOSITION, ALGORITHM, METHOD.

Summary text

The object of the research - multiflow decomposition algorithms and methods for networks and their practical application.

The purpose of the research - to obtain effective multiflow decomposition methods and algorithms for networks, to show that the obtained mathematical models can be used in practice in various fields.

Methods of research - study of thematic literature, application of statements of linear algebra, application of statements of the theory of algorithms and data structures.

Obtained results - developed effective methods and algorithms for multiflow decomposition, a software module for constructing a general solution to a flow programming problem, a software module demonstrating an example of multiflow decomposition.

Recommendations on the usage - logistics, urban planning, medicine, water resources management, energy.

ВВЕДЕНИЕ

В современном мире крайне актуальными являются задачи, связанные с управлением потоками. Потоковыми моделями можно представить самые разные процессы – от потоков автотранспорта на городских улицах до кровеносной системы человека и различных бизнес-процессов.

В связи с этим немаловажной задачей является сбор информации о потоковых системах при помощи специальных устройств-наблюдателей, называемых сенсорами. Установка таких устройств, однако, может быть крайне дорогостоящей, к тому же, как было показано в книге «Разреженные недоопределенные системы линейных алгебраических уравнений» за авторством Пилипчук Людмилы Андреевны [9], если пытаться искать оптимальное расположение сенсоров используя подход полного перебора, то установление оптимальности каждого возможного их расположения есть полином, зависящий от кол-ва узлов графа. Если перемножить это на время, требуемое для перебора всех возможных вариантов расположения, то получаем крайне трудоемкий алгоритм, использование которого абсолютно неэффективно в случае работы с сетью минимально больших размеров.

Моделирование процессов оценки потоков в ненаблюдаемой части может быть представлено как недоопределенная система линейных алгебраических уравнений, где переменные соответствуют неизвестным потокам и интенсивностям узлов, а уравнения — условиям баланса. Для обеспечения полной наблюдаемости сети необходимо установить сенсоры в определенных узлах, чтобы собрать полную информацию о функции потока. Обработка данных от сенсоров в этих узлах приводит к созданию разреженной системы линейных алгебраических уравнений, единственное решение которой гарантирует полную наблюдаемость сети.

Таким образом целью данной дипломной работы является изучение существующих методов решения поставленной задачи и определение наиболее эффективных из них. Также в данной работе будут рассмотрены возможные сценарии применения представленных методов и будет показано, что выстраиваемые этими методами математические модели можно считать соответствующими условиям данных сценариев с высокой точностью.

ГЛАВА 1
ОБЩИЙ ВИД НЕДОПРЕДЕЛЕННОЙ РАЗРЕЖЕННОЙ СИСТЕМЫ

1.1 Линейная недоопределенная система

Введем в рассмотрение ориентированный связный мультиграф $S = (I, U)$ не содержащий петель, где I – множество узлов, U – множество дуг, определенных на $I \times I, |I| < \infty$, $|U| < \infty$, при этом каждой дуге $(i,j)^k \in U$ ставится в соответствие некоторое число $k \in K, K = \{1, \ldots, |K|\}$, называемое типом потока ($|K| < \infty$).

Для удобства обозначений введем множество $K(i) = \{k \in K : i \in I^k\}$ типов потока, транспортируемых через узел $i \in I$, а также множество $K(i,j) = \{k \in K : (i,j)^k \in U^k\}$ типов потока, протекающих по дугам между вершинами i и j из множества вершин I. Совокупность всех дуг, соединяющих узлы i и j, с различными типами потока будем обозначать просто как $(i,j) = \{(i,j)^k \in U : k \in K(i,j)\}$ и в случаях, где это удобно, будем называть данную совокупность дугой (i,j).

Обозначим связную сеть, соответствующую типу потока $k \in K$ через $S^k = (I^k, U^k), I^k \subseteq I, U^k = \{(i,j)^k : (i,j) \in \widetilde{U}^k\}, \widetilde{U}^k \subseteq U$ – множество дуг сети S^k для потока типа k. Очевидно, совокупность всех таких подсетей образует исходную сеть S.

$$\bigcap_{k=1}^{|K|} S^k = S \tag{1.1}$$

Для окончательного формирования исходной задачи остается только ввести подмножество U_0 множества U и непустые подмножества $K_0(i,j) \subseteq K(i,j)$, $(i,j) \in U_0$, представляющие собой исходные данные для формирования условий баланса различных типов потока на определенных дугах.

На основании этих данных получим недоопределенную систему вида:

$$\sum_{j \in I_i^+(U^k)} x_{ij}^k - \sum_{j \in I_i^-(U^k)} x_{ji}^k = a_i^k, \quad i \in I^k,\ k \in K, \tag{1.2}$$

$$\sum_{(i,j)^k \in U} \lambda_{ij}^{kp} x_{ij}^k = \alpha_p,\ p = \overline{1,q}, \tag{1.3}$$

$$\sum_{k \in K_0(i,j)} x_{ij}^k = z_{ij}, \quad (i,j) \in U_0, \tag{1.4}$$

где $I_i^+(U^k) = \{j \in I^k : (i,j)^k \in U^k\}$ – множество исходящих дуг , $I_i^-(U^k) = \{j \in I^k : (j,i)^k \in U^k\}$ – множество входящих дуг; $a_i^k, \lambda_{ij}^{kp}, \alpha_p, z_{ij} \in \mathbf{R}$ – параметры системы (коэффициенты); $x = (x_{ij}^k, (i,j)^k \in U^k, k \in K)$ – неизвестные значения потока.

Уравнения (1.2) являются условиями баланса потока на узлах, уравнения (1.3) являются условиями баланса потока в системе и уравнения (1.4) являются условиями баланса потока различных типов на дугах определенного ранее множества U_0.

Исходя из такого представления матрица системы (1.2) – (1.4) имеет следующую блочную структуру:

$$A = \begin{bmatrix} M \\ Q \\ T \end{bmatrix}. \tag{1.5}$$

В матрице M записана левая часть уравнений (1.2), причем в виде блочно-диагональной структуры, т.е. матрица представляется в виде диагональных блоков $M_k, k = 1, \ldots, |K|$ каждый из которых имеет размер $|I^k| \times |U^k|$ и представляет собой матрицу инцедентности системы S^k. Исходя из данной структуры очевидно, что данная часть матрицы в высокой степени разреженна.

В матрице Q записана левая часть уравнений (1.3), тут уже каждое уравнение связывает некоторым соотношением дуги всей системы и в общем случае данная часть матрицы разреженной являться не будет. Состоит данная матрица из элементов $\lambda_{ij}^{kp}, (i,j) \in U, k \in K(i,j), p = \overline{1,q}$ и имеет размер $q \times |U|$.

Последняя матрица T представляет собой левую часть уравнений (1.3) и состоит из нулей и единиц, являясь, по сути, выборочной матрицей инцедентности сети. Её размеры – $|U_0| \times |U|$.

Ранг матрицы системы (1.1) равен $\sum\limits_{k \in K} |I^k| - |K|$.

Поскольку матрица M системы (1.2) имеет блочно-диалгональный вид, состоит из диагональных блоков $M_k,\ k = 1, \ldots, |K|$ и $rank\, M_k = |I^k| - 1$ (согласно источнику [4]), то $rank\, M = \sum\limits_{k=1}^{|K|} rank\, M_k = \sum\limits_{k \in K} (|I^k| - 1) = \sum\limits_{k \in K} |I^k| - |K|$.

Теперь рассмотрим ранги систем (1.3) и (1.4). Будем полагать, что система строк объединения матриц Q и T линейно независима (в противном случае задача сводится к решению системы меньшей размерности уже с независимой системой строк), тогда $rank\, Q = q, rank\, T = |U_0|$.

Очевидно, что для того, чтобы система имела решение, необходимо, чтобы кол-во уравнений не превышало число неизвестных (при условии, что система строк матрицы линейно независима).

Без ограничения общности будем полагать, что это так, то есть:

$$\sum_{k\in K}|I^k| + q + |U_0| < |U|. \tag{1.6}$$

1.2 Критерий опорности

Одним из эффективнейших алгоритмов, значительно ускоряющих решение линейных систем алгебраических уровнений является построение остовного дерева для нахождения базиса пространства решений, такой метод был описан в [3] и является актуальным по сегодняшний день.

Применительно к поставленной задачи воспользуемся терминологией введенной в книге [8]:

Опорой сети $S = (I, U)$ *системы* (1.1) *называют множество дуг* $U_T = \{U_T^k \subseteq U^k, k \in K\}$, *такое, что система*

$$\sum_{j\in I_i^+(\hat{U}^k)} x_{ij}^k - \sum_{j\in I_i^-(\hat{U}^k)} x_{ji}^k = 0, i \in I^k,\ k \in K \tag{1.7}$$

имеет только тривиальное решение для $\hat{U^k} = U_T^k, k \in K \setminus k_0$, *но нетривиальное решение для* $\hat{U^{k_0}} = U_T^{k_0} \bigcup (i,j)^{k_0},\ (i,j)^{k_0} \notin U_T^{k_0},\ k_0 \in K$.

То есть каждая подсистема системы на опорном множестве дуг имеет тривиальное решение, однако при добавлении к данному множеству какой-либо другой дуги подсистема, частью которой является добавленная дуга, принимает нетривиальное решение.

Воспользуемся терминологией из всё той же книги и введем понятие опоры сети.

Множество $U_T = \{U_T^k, k \in K\}$ *является опорой сети* $S = (I, U)$ *для системы* (1.2) – (1.4) *если для каждого* $k \in K$ *множество дуг* U_T^k *является покрывающим (остовным) деревом сети* $S^k = (I^k, U^k)$.

Использование данного алгоритма в нашем случае абсолютно уместно, так как матрица системы (1.2) имеет блочно-диагональную структуру, следовательно, мы можем разбить нашу систему на $|K|$ независимых подсистем, каждая из которых соответствует отдельному блоку для фиксированного $k \in K$ и имеет следующий вид:

$$\sum_{j\in I_i^+(U^k)} x_{ij}^k - \sum_{j\in I_i^-(U^k)} x_{ji}^k = a_i^k, i \in I^k. \tag{1.8}$$

Что соотвествует вышевведенному определению опоры, нужно лишь заменить правую часть на однородную.

1.3 Характеристические векторы

Введем конструктивное определение характеристического вектора. Воспользуемся тем фактом, что, при добавлении любой дуги графа в остовное дерево мы получим единственный цикл [7].

Рассмотрим такую ситуацию, пусть $(u,v)^k \in U^k \setminus U_T^k$ – некоторая, не содержащаяся в опоре дуга. Тогда её добавление в опору приведет к созданию ровно одного цикла. Обозначим этот цикл L_{uv}^k и будем называть его циклом, порожденным дугой $(u,v)^k \in U^k \setminus U_T^k$.

Определим направление обхода цикла L_{uv}^k в соответствии с направлением дуги $(u,v)^k$, которая является прямой.

Говорят, что дуга $(i,j)^k \in L_{uv}^k$, где $k \in K$ фиксировано, называется прямой дугой цикла L_{uv}^k, если направление дуги $(i,j)^k$ совпадает с направлением дуги $(u,v)^k$ в цикле L_{uv}^k. Аналогично если дуга $(i,j)^k \in L^k$, где $k \in K$ фиксировано, имеет направление, противоположное направлению дуги $(u,v)^k$ в цикле L_{uv}^k, то такую дугу называют обратной дугой цикла L_{uv}^k.

Введем в рассмотрение L_{uv}^{k+} – множество прямых дуг цикла L_{uv}^k и множество L_{uv}^{k-} обратных дуг цикла L_{uv}^k, а также функцию направления дуги:

$$\operatorname{sign}(i,j)^k = \begin{cases} 1,\ (i,j)^k \in L_{uv}^{k+}, \\ 0,\ (i,j)^k \notin L_{uv}^k, \\ -1, (i,j)^k \in L_{uv}^{k-}. \end{cases} \tag{1.9}$$

Теперь, определив все необходимые термины, мы можем перейти к формулировке, что такое характеристический вектор.

Вектор $\delta^k(u,v) = (\delta_{ij}^k(u,v), (i,j)^k \in U^k)$, удовлетворяющий условиям:

• *Вектор $\delta^k(u,v)$ определен на всем множестве дуг $(u,v)^k \in U^k \setminus U_T^k$.*

• $\delta_{ij}^k(u,v) = \operatorname{sign}(i,j)^k$.

Называется характеристическим вектором дуги $(u,v)^k$ относительно опоры U_T^k.

Так как в дальнейшем метод подразумевает использование лишь одной опоры для каждой из k подсистем, то сократим это до *характеристического вектора дуги* $(u,v)^k$.

Для дальнейшего применения характеристических векторов небходимо привести следующие их свойства:

• Характеристический вектор $\delta^k(u,v)$, дуги $(u,v)^k \in U^k \setminus U_T^k$, является решением однородной системы (1.7);

• Множество $\{\delta^k(\tau,\rho), (\tau,\rho)^k \in U^k \setminus U_T^k\}$ характеристических векторов образует базис пространства решений однородной системы (1.7).

Доказательство данных утверждений можно найти в книге [9].

На основании этих утверждений можно сформулировать и доказать следующую теорему:

Система (1.7) имеет следующее общее решение:

$$x_{ij}^k = \sum_{(u,v)^k \in U^k \setminus U_T^k} x_{uv}^k \text{sign}(i,j)^{L_{uv}^k} + \\ + \left(\tilde{x}_{ij}^k - \sum_{(u,v)^k \in U^k \setminus U_T^k} \tilde{x}_{uv}^k \text{sign}(i,j)^{L_{uv}^k} \right), \quad (1.10)$$
$$(i,j)^k \in U_T^k, \quad (u,v)^k \in U^k \setminus U_T^k,$$

где $\tilde{x}^k = (\tilde{x}_{ij}^k, (i,j)^k \in U^k)$ – некоторое найденное частное решение системы (1.8).

Мы знаем, что на основании описанного выше второго свойства характеристических векторов их множество образует базис пространства решений однородной системы.

Исходя из этой информации, компоненты решения $x^k, k = \overline{1, |K|}$, можно представить в виде следующей линейной комбинации характеристических векторов просуммированных с некоторым частным решением неоднородной системы:

$$x^k = \sum_{(u,v)^k \in U^k \setminus U_T^k} \alpha_{uv}^k \delta^k(u,v) + \tilde{x}^k, \quad (1.11)$$

где $\alpha_{uv}^k \in \mathbf{R}$ – некоторые коэффициенты, которые необходимо найти.

Покоординатно распишем данное выражение, в итоге получим:

$$x_{ij}^k = \sum_{(u,v)^k \in U^k \setminus U_T^k} \alpha_{uv}^k \delta_{ij}^k(u,v) + \tilde{x}_{ij}^k, \quad (i,j)^k \in U_T^k; \quad (1.12)$$

$$x_{uv}^k = \alpha_{uv}^k + \tilde{x}_{uv}^k, \quad (u,v)^k \in U^k \setminus U_T^k. \quad (1.13)$$

Представим (1.13) как $\alpha_{uv}^k = x_{uv}^k - \tilde{x}_{uv}^k, \quad (u,v)^k \in U^k \setminus U_T^k$, и подставим в (1.12). В итоге получим:

$$x_{ij}^k = \sum_{(u,v)^k \in U^k \setminus U_T^k} (x_{uv}^k - \tilde{x}_{uv}^k) \delta_{ij}^k(u,v) + \tilde{x}_{ij}^k, \quad (i,j)^k \in U_T^k. \quad (1.14)$$

Что после раскрытия скобок даст нам исходное выражение (1.10), что и требовалось доказать.

Для нахождения частного решения при практическом применении алгоритма предлагается заменить $\tilde{x}_{uv}^k = 0$, $(u,v)^k \in U^k \setminus U_T^k$ и находить частное решение из системы:

$$\sum_{j \in I_i^+(U_T^k)} \tilde{x}_{ij}^k - \sum_{j \in I_i^-(U_T^k)} \tilde{x}_{ji}^k = a_i^k, \quad i \in I^k.$$

При использовании данной рекомендации формула упрощается до:

$$x_{ij}^k = \sum_{(u,v)^k \in U^k \setminus U_T^k} x_{uv}^k \mathrm{sign}(i,j)^{L_{uv}^k} + \tilde{x}_{ij}^k \,, (i,j)^k \in U_T^k. \tag{1.15}$$

Будем предполагать, что рекоммендации по нахождению частного решения соблюдены и использовать (1.15) в качестве основной формулы.

ГЛАВА 2
ДЕКОМПОЗИЦИЯ РАЗРЕЖЕННОЙ СИСТЕМЫ

В этой главе будут рассмотрены алгоритмы декомпозиции системы линейных алгебраических уравнений полного ранга с дополнительными условиями форматов (1.3), (1.4). Более подробное описание каждого из переходов приведено в книге [17]. Обозначения, принятые в первой главе, актуальны и для данной.

2.1 Основные переходы

Воспользуемся основной формулой (1.15) для каждой из получаемых опор подсетей $S_k, k = \overline{1, |K|}$ и заменим переменные на соответствующие выражения в ограничениях второго типа (1.3):

$$\sum_{(i,j)^k \in U} \lambda_{ij}^{kp} x_{ij}^k = \sum_{k \in K} \sum_{(i,j)^k \in U_T^k} \lambda_{ij}^{kp} \left(\sum_{(u,v)^k \in U^k \setminus U_T^k} x_{uv}^k \mathrm{sign}(i,j)^{L_{uv}^k} + \tilde{x}_{ij}^k \right) + \\ + \sum_{k \in K} \sum_{(u,v)^k \in U^k \setminus U_T^k} \lambda_{uv}^{kp} x_{uv}^k = \alpha_p, \quad p = \overline{1, q}. \tag{2.1}$$

При помощи серии эквивалентных преобразований представим данное выражение как

$$\sum_{k \in K} \sum_{(u,v)^k \in U^k \setminus U_T^k} x_{uv}^k \left(\lambda_{uv}^{kp} + \sum_{(i,j)^k \in U_T^k} \lambda_{ij}^{kp} \mathrm{sign}(i,j)^{L_{\tau\rho}^k} \right) = \\ = \alpha_p - \sum_{k \in K} \sum_{(i,j)^k \in U_T^k} \lambda_{ij}^{kp} \tilde{x}_{ij}^k \,, \; p = \overline{1, q} \,. \tag{2.2}$$

Введем вспомогательную определяющую функцию (в литературе на нее обычно ссылаются как "функция Determining"[17]).

$$D_p(L_{uv}^k) = \sum_{(i,j)^k \in L_{uv}^k} \lambda_{ij}^{kp} \mathrm{sign}(i,j)^{L_{uv}^k} \tag{2.3}$$

Также введем вспомогательный вектор:

$$A^p = \alpha_p - \sum_{k \in K} \sum_{(i,j)^k \in U_T^k} \lambda_{ij}^{kp} \, \tilde{x}_{ij}^k \,, \; p = \overline{1, q} \,. \tag{2.4}$$

С учетом нововведенных обозначений уравнения (2.2) преобразуются в следующее:

$$\sum_{k\in K}\sum_{(u,v)^k\in U^k} D_p(L_{uv}^k)x_{uv}^k = A^p, \quad p=\overline{1,q}\,. \tag{2.5}$$

Сделаем разделение переменных по множествам $U^k\backslash U_T^k$, а также U_T^k, в результате уравнение (2.5) преобразуется в:

$$\sum_{k\in K}\sum_{(u,v)^k\in U^k\backslash U_T^k} D_p(L_{uv}^k)x_{uv}^k = A^p - \sum_{k\in K}\sum_{(u,v)^k\in U_T^k} D_p(L_{uv}^k)x_{uv}^k, p=\overline{1,q}\,. \tag{2.6}$$

Итого декомпозиция данной подсистемы завершена, далее необходимо показать, что аналогичные суждения актуальны и для систем (1.4) типа, после чего собрать решение воедино.

2.2 Возврат к исходной системе и группировка

При рассмотрении и анализе систем (1.3) и (1.4) можно обратить внимание на то, что множество всех линейных комбинаций вида (1.4) является подмножеством всех линейных комбинаций вида (1.3).

Можно выполнить над системой (1.4) аналогичную серию эквивалентных преобразований с использованием характеристических векторов и образующих циклы подмножеств дуг U_T^k, $k=\overline{1,|K|}$. В итоге получим:

$$\begin{gathered}\sum_{k\in K}\sum_{(u,v)^k\in U_T^k}\delta_{ij}(L_{uv}^k)x_{uv}^k = \\ = z_{ij} - \sum_{k\in K_0(i,j)\in U_T^k}\tilde{x}_{ij}^k - \sum_{k\in K_0(i,j)}\sum_{(u,v)^k\in U^k\backslash U_T^k}\delta_{ij}(L_{uv}^k)x_{uv}^k, \quad (i,j)\in U_0.\end{gathered} \tag{2.7}$$

Решаемая в данном сегменте алгоритма подзадача состоит в том, чтобы найти эффективный способ хранения уравнений типа (1.3) и (1.4) в рамках некоторой структуры данных (на практике будем двумерный массив с некоторой заданной нумерацией), для чего необходимо придумать компактное представление, позволяющее объединить данные типы уравнений общим видом.

Для решения данной задачи в [17] предлагается использовать экстраполирование функции "Determining"на множество U_0 следующим образом:

$$D_{ij}(L_{uv}^k) = \sum_{(t,s)^k\in L_{uv}^k} \mathrm{sign(t,s)^{L_{uv}^k}sign(i,j)^{L_{uv}^k}}, (i,j)\in U_0. \tag{2.8}$$

Данное представления позволяет объединить полученные представления декомпозиционной системы общей формулой, что позволяет объединить эти представления в одну матрицу, реализованную, как было указано выше, на двумерном массива.

Введем некоторую нумерацию каждой дуги множества $U_0 \cap U^k$ и будем хранить вектор из $|U_0| + q$ значений полученных коэффициентов согласно данной нумерации, в общем отобразим указанное множество во множество соответсвующих номеров. Причина использования двумерного массива заключается в том, что обращение к ячейкам памяти в данной структуре данных выполняется за константное время. Следует обратить внимание, что в данных множествах дуги могут повторяться и при отображении каждой из них в некоторый номер следует хранить хэш-таблицу уже отображенных дуг с их номерами для избежания увеличения размерности получаемой матрицы, а также для дальнейшего восстановления дуг по номеру элемента таблицы.

Итоговая форма уравнений с использованием функции "Determining":

$$\sum_{k \in K_0(i,j) \in U_T^k} \tilde{x}_{ij}^k = \sum_{k \in K} \sum_{(u,v)^k \in U^k \setminus U_T^k} D_{ij}(L_{uv}^k) x_{uv}^k + z_{ij}, (i,j) \in U_0. \tag{2.9}$$

Обозначим введенную нумерацию как $h_k(i,j)$ и, с учетом всего вышенаписанного, запишем получившуюся матрицу:

$$Rx = \beta. \tag{2.10}$$

Вектор свободных членов β равняется:

$$\beta = \begin{pmatrix} \beta_p, & p = \overline{1,q}, \\ \beta_{q+h(i,j)}, & (i,j) \in U_0 \end{pmatrix}, \tag{2.11}$$

где вектор $\beta_{q+h(i,j)} = \sum\limits_{k \in K} \sum\limits_{(u,v)^k \in U^k \setminus U_T^k} D_{ij}(L_{uv}^k) x_{uv}^k + z_{ij}$

и вектор $\beta_p = A^p - \sum\limits_{k \in K} \sum\limits_{(u,v)^k \in U_T^k} D_p(L_{uv}^k) x_{uv}^k,\ p = \overline{1,q}$

В левой же части $R = \binom{R_1}{R_2}$, $R_1 = (D_p(L_{uv}^k),\ p = \overline{1,q})$ – подсистема, размерность которой $q \times |U|$, $R_2 = (D_{ij}(L_{uv}^k)),\ h_k(i,j)^k = \overline{q+1, q+|U_0|})$ – подсистема, размерность которой $|U_0| \times |U|$, $x = (x_{uv}^k, (u,v)^k \in U^k,\ k \in K)$ – список переменных, выстроенный на основе введенной нумерации $h_k(i,j)^k, (i,j)^k \in U$.

Далее для решения полученной системы воспользуемся обратным преобразованием:

$$x = R^{-1}\beta\,. \tag{2.12}$$

Если все рекомендации по построению остовных деревьев, группировке переменных, использованию надлежащей функции "Determining"и прочие [6] были выполнены в надлежащем порядке, то детерминант полученной системы будет отличен от нуля, иначе говоря – она будет невырождена и иметь единственное решение.

Пусть найдено обратное преобразование $R^{-1} = (R_{t,s}, t = \overline{1, q + |U_0|}, s = \overline{1, |U|})$, тогда получаем решение:

$$x^k_{uv} = \sum_{p=1}^{q} R_{t,p}\beta_p + \sum_{(i,j)\in U_0} R_{t,q+h^k(i,j)}\beta_{q+h^k(i,j)},$$

$$t = h^k(u,v), (u,v)^k \in U^k \backslash U^k_T,\ k \in K.$$

Оставшиеся компоненты $\tilde{x}^k = (\tilde{x}^k_{ij}, (i,j)^k \in U^k_T)$ были найдены ранее как частное решение системы (1.8).

Таким образом, определены все неизвестные $x^k = (x^k_{ij},\ (i,j)^k \in U^k)$, $k \in K$ системы (1.2) – (1.4).

2.3 Преимущества и специфика

Кратко изложим наиболее важные аспекты метода. Хотя строгие оценки сложности алгоритмов здесь не рассматриваются, следует обратить внимание на то, что описанный подход осуществлен на надлежащих структурах данных, что ведет к эффективному алгоритму: часть вычислений выполнена на небольших подмножествах дуг, например на изолированных циклах (1.9), (1.10) или покрывающих (остовных) деревьях (2.1), (2.7). Использование сетевой структуры ограничений позволяет выполнять декомпозицию опоры, и, следовательно, декомпозицию системы, и, наконец, выполняется обращение матрицы R (2.12) размера, намного меньшего, чем размер исходной системы (1.2) – (1.4). Кроме этого, независимые результаты, полученные для каждого типа потока k из множества K, например теорема (1.10), свойства характеристических векторов $\delta^k(u,v)$, формулы (1.15) дают возможность строить решения больших недоопределенных систем линейных уравнений в параллельной среде.

Однако преимущества указанного подхода к решению недоопределенных линейных систем оценивается в контексте больших неоднородных задач потокового программирования, где системы вида (1.2) – (1.4) являются частью системы основных ограничений задачи.

ГЛАВА 3
ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ И АЛГОРИТМОВ ДЕКОМПОЗИЦИИ РАЗРЕЖЕННЫХ ЛИНЕЙНЫХ СИСТЕМ: ПРИМЕНЕНИЕ В РАЗЛИЧНЫХ СФЕРАХ

В современном мире разреженные линейные системы находят широкое применение в самых различных сферах деятельности, от медицины до городского планирования. Эффективное решение задач, связанных с анализом и обработкой данных, требует применения продвинутых технологий и алгоритмов, способных справляться с большими объемами информации и обеспечивать высокую точность результатов. Декомпозиция разреженных линейных систем представляет собой один из ключевых подходов, позволяющий значительно упростить вычислительные процессы и оптимизировать использование ресурсов.

В медицине, например, алгоритмы декомпозиции помогают в анализе больших массивов данных, получаемых от медицинских устройств и сенсоров, что способствует более точной диагностике и персонализированному лечению. В логистике эти технологии позволяют оптимизировать маршруты доставки и управление запасами, снижая затраты и повышая эффективность операций. Городское планирование также выигрывает от применения алгоритмов декомпозиции, так как они позволяют моделировать и прогнозировать потоки людей и транспортных средств, что способствует созданию более комфортной городской среды.

Управление водными ресурсами является еще одной важной областью, где технологии декомпозиции находят свое применение. Они позволяют эффективно анализировать данные о потоках воды, обеспечивая устойчивое управление водными ресурсами и предотвращая негативные последствия, такие как наводнения или засухи. Кроме того, алгоритмы декомпозиции могут быть использованы в таких сферах, как экология, энергетика и агропромышленный комплекс, где требуется анализ сложных систем и процессов.

3.1 Применение технологий и алгоритмов декомпозиции в логистике

Логистика представляет собой комплекс мероприятий, направленных на эффективное управление потоками товаров, услуг и информации от точки производства до конечного потребителя. В условиях глобализации и увеличения объемов торговли логистические процессы становятся все более сложными и многоуровневыми. Основные задачи логистики включают оптимизацию транс-

портных маршрутов, управление запасами, распределение ресурсов и планирование производственных процессов. Для решения этих задач необходимо принимать во внимание множество факторов, таких как спрос, время доставки и доступность ресурсов, для анализа которых необходимо знать значения товарных потоков в логистической сети. При этом в целях экономии хотелось бы минимизировать количество сенсоров, использующихся в процессе сбора информации.

Приведем пример простейшей логистической сети с несколькими источниками и стоками, а также транспортными узлами.

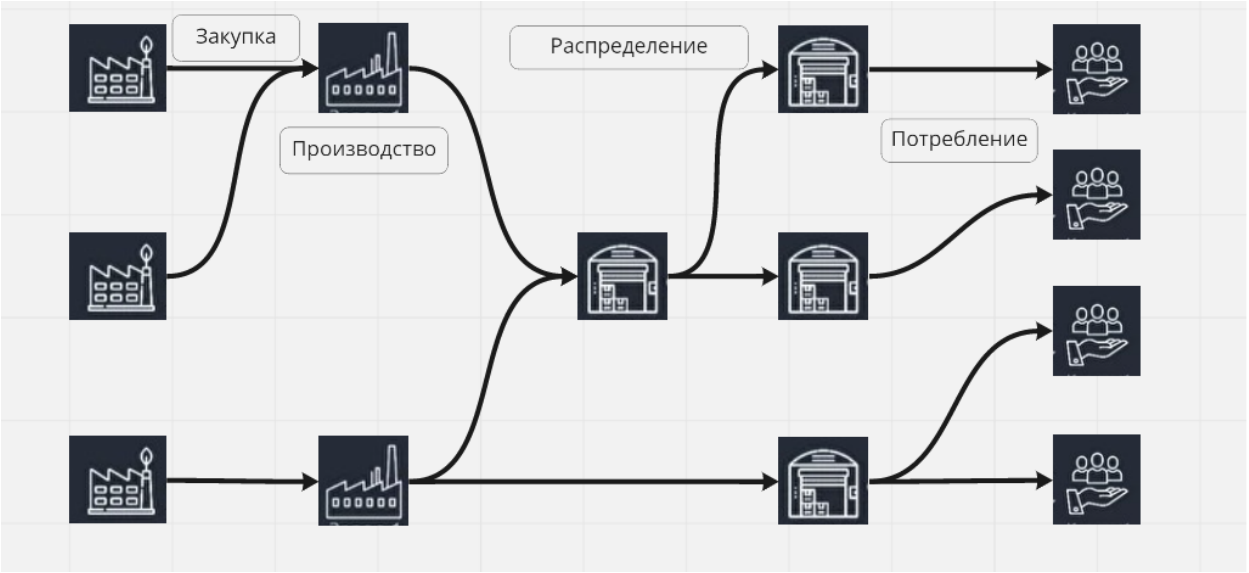

Рисунок 3.1 – Транспортная сеть

Сеть, представленную на рисунке 3.1, нетрудно представить в виде орграфа, изображенного на рисунке 3.2:

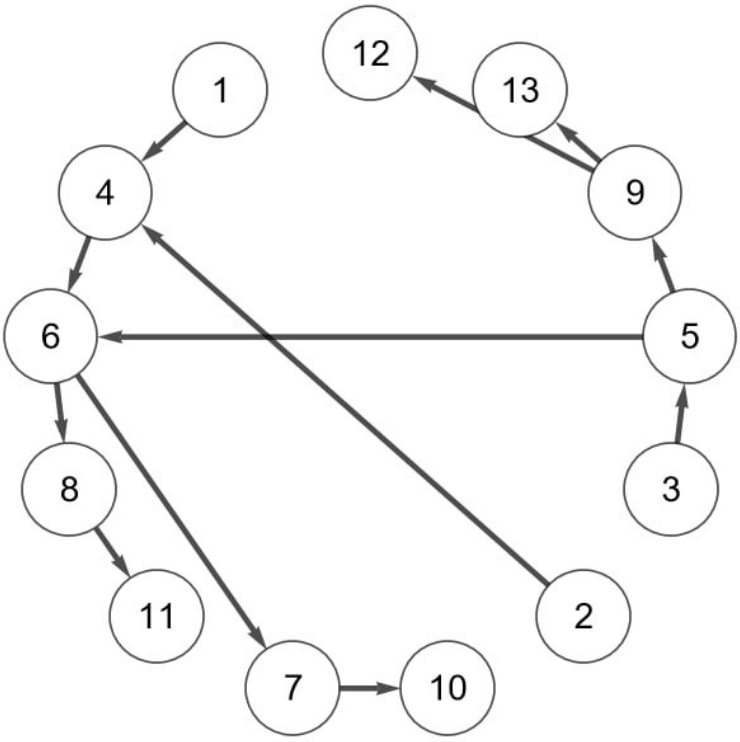

Рисунок 3.2 – Модель транспортной сети на графе

Или для случая мультисети (k = 3 различных товаров) представить в виде мультиграфа, изображенного на рисунке 3.3:

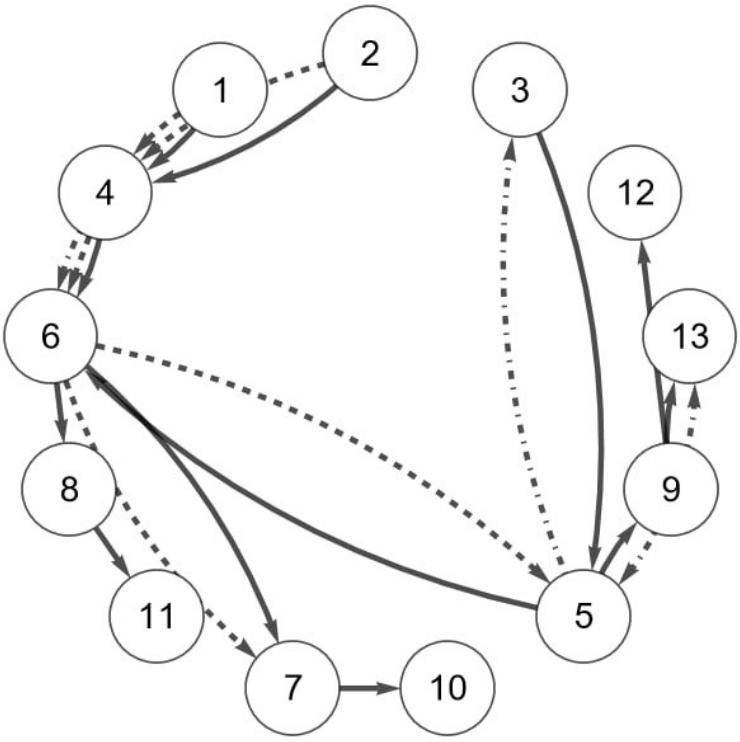

Рисунок 3.3 – Модель транспортной сети на мультиграфе

Таким образом показано, что задача оптимального распложения видео-

камер (или других следящих устройств) в логистической сети представима в виде задачи нахождения оптимального расположения сенсоров в графе (мультиграфе в случае мультипотока).

Следовательно, подход, основанный на мультиграфах, не только упрощает процесс проектирования системы видеонаблюдения, но и обеспечивает более высокую степень адаптивности и гибкости в управлении логистическими процессами.

3.2 Применение технологий и алгоритмов декомпозиции в медицине

Современная медицина все больше полагается на математические модели и алгоритмы для анализа биологических данных. Одной из актуальных задач является отслеживание количества питательных веществ в кровеносной системе, что имеет важное значение для диагностики и лечения различных заболеваний. В данной главе рассматривается возможность применения алгоритмов декомпозиции к моделям кровеносной системы человека, для чего показывается, что кровеносную систему человека можно представить в виде орграфа.

Приведем пример простейшей модели кровеносной системы человека, представленной на рисунке 3.4, и упрощенно представим ее в виде орграфа.

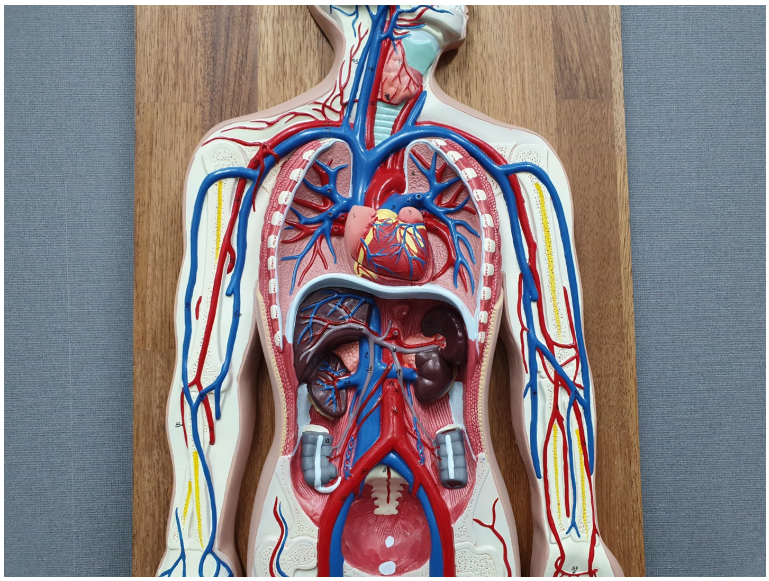

Рисунок 3.4 – Модель кровеносной системы

Можно представить данную модель в виде орграфа, взяв в качестве узлов внутренние органы такие как сердце, легкие, почки, печень, желудок и прочие, а также места слияния крупных вен и артерий, таких как аорта и венозные синусы. В качестве значений потока можно рассматривать давление, оказываемое потоком крови на стенки сосудов.

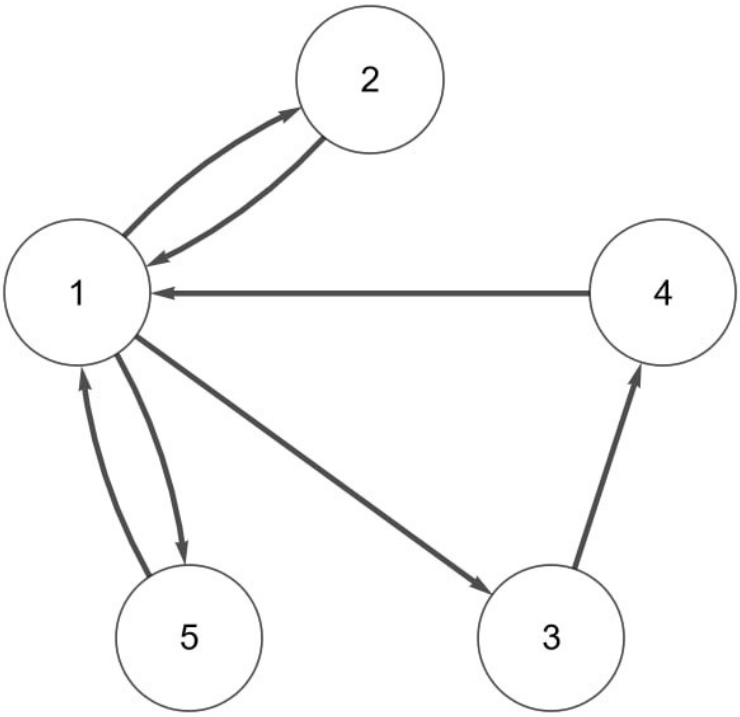

Рисунок 3.5 – Модель кровеносной системы на графе

На рисунке 3.5 узел с номером 1 отражает сердце, узел с номером 2 отражает легкие, узел с номером 3 отражает почки, узел с номером 4 отражает печень, узел с номером 5 отражает желудок, а дуги отражают совокупность кровеносных сосудов и систем, находящихся между двумя органами. Очевидно, данную модель можно дополнить большим числом узлов и дуг, более подробно представить в ней венозную систему и систему капилляров.

Также хорошей практикой, увеличивающей соответствие данной модели реальным кровеносным системам, будет разбиение каждого органа на два и более узла, связанных между собой двунаправленными дугами. Так, например, сердце можно разбить на правый желудочек, левый желудочек, правое предсердие и левое предсердие.

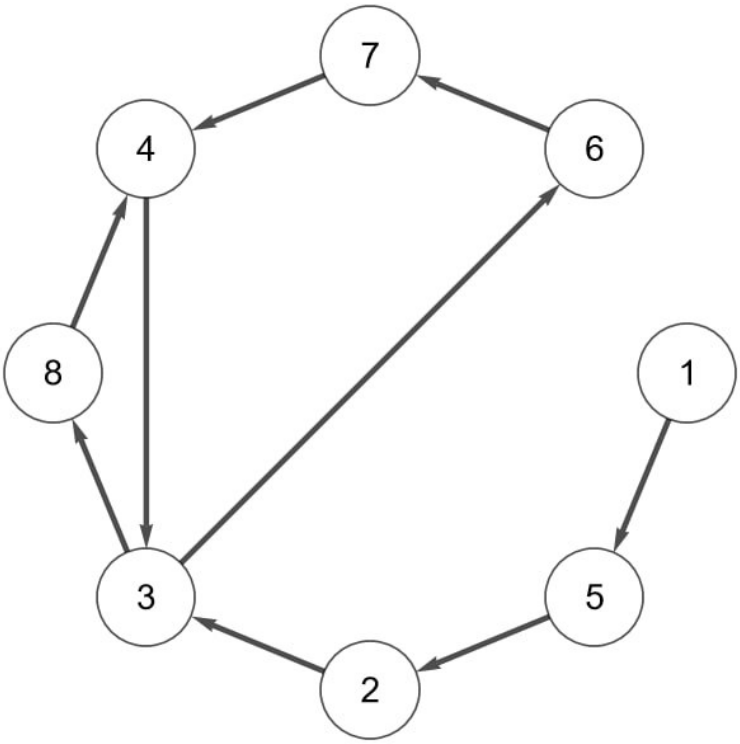

Рисунок 3.6 – Усовершенствованная модель кровеносной системы на графе

На рисунке 3.6 узел с номером 1 отражает правый желудочек сердца, узел с номером 2 отражает левое предсердие, узел с номером 3 отражает левый желудочек сердца, узел с номером 4 отражает правое предсердие, узел с номером 5 отражает легкие, узел с номером 6 отражает почки, узел с номером 7 отражает печень, а узел с номером 8 отражает желудок.

Такая модель уже более полно отражает данный участок кровеносной системы. Помимо упомянутого выше добавления большего числа узлов, отражающих, например, места слияния крупных вен и артерий, можно также рассматривать в качестве значений потока не просто объем циркулирующей крови, а содержание в крови тех или иных веществ. Несмотря на то, что на сегодняшний день медицина не разработала способ измерять кол-во тех или иных веществ в глубоких участках кровеносной системы, это не значит, что подобные методы не будут разработаны в будущем. Тем не менее данные методики можно применять к каким-то участкам кровеносной системы в которых допустима установка катетеров, либо же рассматривать в качестве двух типов потока венозную и артериальную кровь.

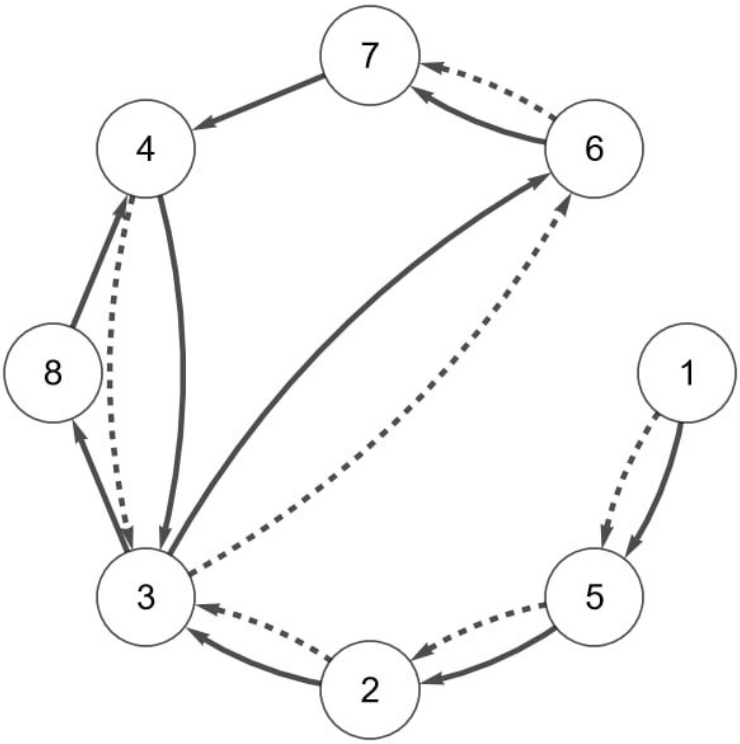

Рисунок 3.7 – Модель кровеносной системы на мультиграфе

На рисунке 3.7 номера узлов соответствуют тем же органам, что и на предыдущей модели, непрерывными стрелочками отображаются подсистемы, по которым протекает артериальная кровь, а пунктирными – по которым протекает венозная кровь.

Таким образом показано, что алгоритмы и технологии декомпозиции разреженных линейных систем могут использоваться в медицине, так как модели кровеносной системы представимы в виде математической модели на орграфе или же мультиграфе.

Следовательно, подход, основанный на мультиграфах, уменьшает количество потенциально затраченных медицинским учереждением ресурсов на установку отслеживающих поток крови устройств, а также минимизирует сопутствующие при вмешательстве в кровеносную систему риски.

3.3 Применение технологий и алгоритмов декомпозиции в городском планировании

Задача управления и наблюдения за потоками является немаловажной частью современного городского планирования. Размеры современных транспортных сетей, а также объемы городского трафика за последние 100 лет в среднем увеличились в десятки раз [12], в силу этого факта полагаться исключительно на интуицию и опыт планировщика становится совершенно неприемлемо. Для

решения данной проблемы также можно использовать алгоритмы и технологии декомпозиции потока в сетях.

Приведем пример некоторой городской транспортной сети.

Рисунок 3.8 – Городская транспортная сеть

Изображенная на рисунке 3.8 сеть состоит из нескольких перекрестков, связанных участками дорог различных типов, а именно грунтовых и асфальтированных, а также пешеходных тратуаров и иных важных для рассмотрения пешеходных маршрутов.

Можно представить данную сеть в виде двунапревленного орграфа, то есть такого орграфа, в котором для каждой исходящей дуги существует симметричная ей входящая.

По аналогии с приведенными ранее примерами будем строить математические модели данной сети в порядке усложнения их структуры. Начнем с простейшего примера такой сети, не учитывающего неоднородность дорожной структуры, а также не принимающего во внимание пешеходный трафик:

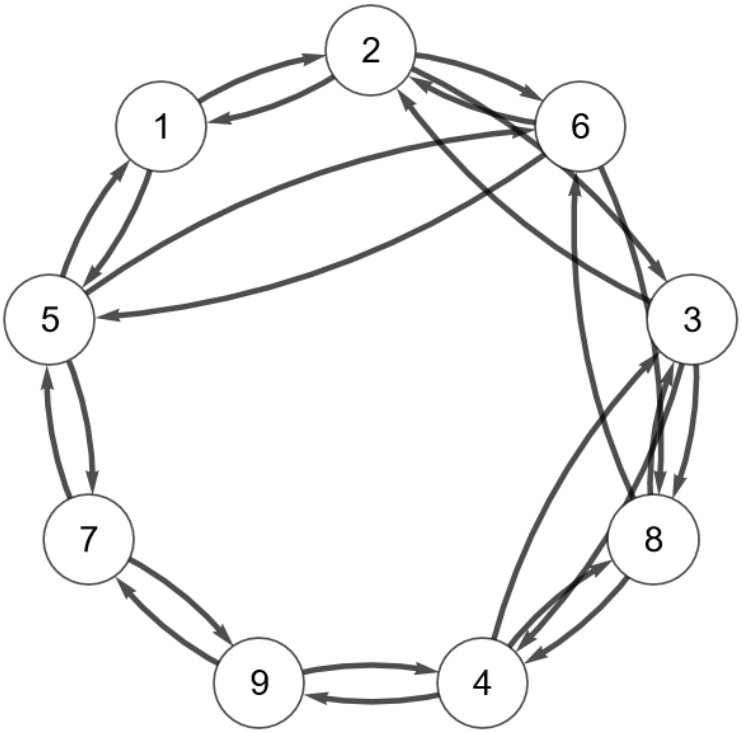

Рисунок 3.9 – Модель городской транспортной сети

На представленной на рисунке 3.9 модели вершины графа отображают перекрестки дорог, дуги – сами дороги, при этом все существующие между вершинами дуги имеют симметричные, так как поток дорожного трафика, очевидно, всегда направлен в обе стороны за исключением дорог с односторонним движением, которых в данном примере нет.

В связи с этим в данном случае может показаться разумным использовать обычный граф вместо орграфа, однако такой подход не нагляден и может привести к путанице, так как для корректности полученного ответа важно разделять поток на прямой дуге и обратной, что не может быть продемонстрированно на обычном графе.

Покажем также на рисунке 3.10 пример построения такой модели с двумя и более типами потока, взяв в качестве потока второго типа пешеходный трафик.

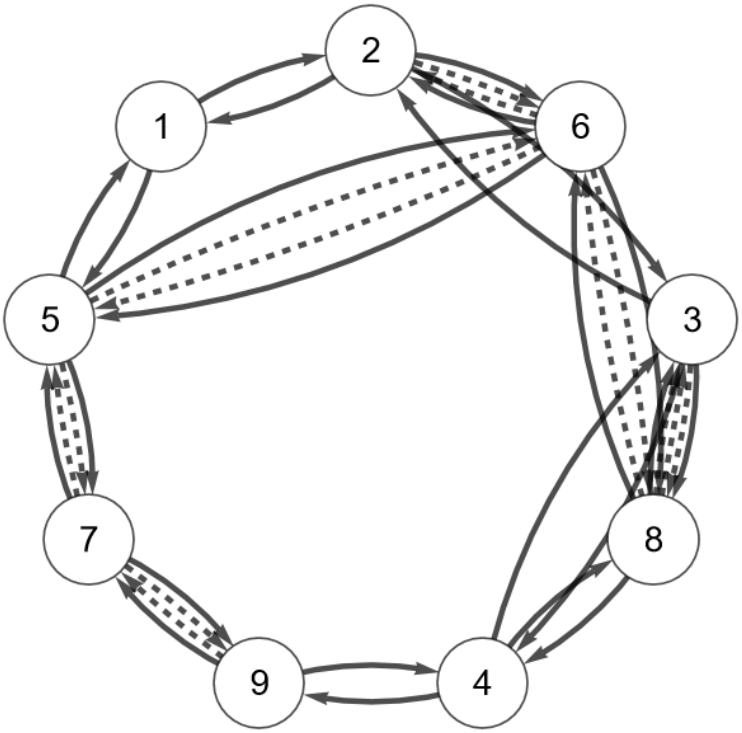

Рисунок 3.10 – Усовершенствованная модель городской транспортной сети

Таким образом показано, что дорожные сети можно представить в виде математических моделей на графах, которые соответствуют всем необходимым критериям для использования на них представленных алгоритмов и методов.

ГЛАВА 4
ПОСТРОЕНИЕ ОБЩЕГО РЕШЕНИЯ РАЗРЕЖЕННОЙ СИСТЕМЫ С МАТРИЦЕЙ ИНЦИДЕНТНОСТИ ГРАФА И РЕАЛИЗАЦИЯ В WOLFRAM MATHEMATICA

В ходе выполнения дипломной работы была реализована программа построения обшего решения поставленной задачи. Рассмотрим этапы выполнения программы.

4.1 Построение остовного дерева и визуализация орграфа

Чтения графа из файла с данными осуществлялось комбинацией внутренних функций Wolfram Mathematica, приведем код:

```
SetDirectory[DirectoryName[ToFileName[FileName/.NotebookInformation[
EvaluationNotebook[]]]]]
data = Import[input − − − .txt]
```

Оптимальным способом нахождения базиса пространства решений является построение остовного и корневого деревьев. Приведем код построения и визуализации остовного дерева:

```
edgestyle1={};
For[i=1,i<=Length[arcsn],i++,{x1,y1}=List@@arcsn[[i]];
For[j=1,j<=Length[tree],j++,{x2,y2}=List@@tree[[j]];
If[(x1==x2&&y1==y2)||(x1==y2&&x2==y1), AppendTo[edgestyle1, x1->y1]]]]
Print[Spanning tree]Graph[arcsn, GraphLayout → CircularEmbedding,
VertexLabels → Placed[Name, Center], VertexSize → 0.3, VertexLabelStyle → 15,
EdgeShapeFunction->{{Arrow, ArrowSize → 0.06}},
VertexStyle → LightGray, EdgeStyle->{(#)->
Directive[Black, Thick]&/@edgestyle1/.List->Sequence}]
```

Результат выполнения данной части программы представлен на рисунке 4.1 (выделенные жирным дуги принадлежат остовному дереву).

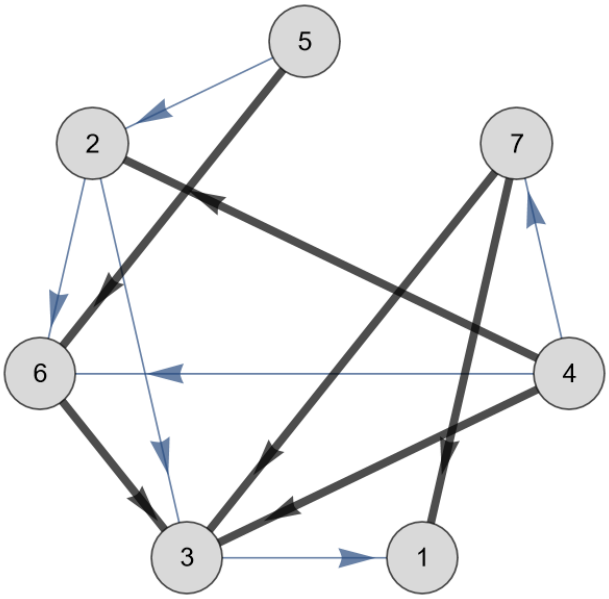

Рисунок 4.1 – Ориентированный граф. Дуги остовного дерева выделены жирными линиями

4.2 Построение корневого дерева и его хранение в памяти компьютера

Для определения базисных дуг необходимо выполнить несколько ключевых шагов, начиная с построения корневого дерева. Это дерево служит основой для дальнейшего анализа и обработки данных. Важно не только создать это дерево, но и обеспечить его правильное хранение в памяти компьютера. Эффективное представление структуры корневого дерева в памяти является критически важным, так как от этого зависит скорость доступа к данным и общая производительность алгоритмов, использующих это дерево. Правильная организация хранения позволяет оптимизировать операции поиска, вставки и удаления, что в конечном итоге способствует более эффективной работе с базисными дугами.

Приведем код, реализующий данную часть программы:

```
tRoot = 2;
BuildRootTree[g_]:=Module[{s={},root=tRoot},
tPred=ConstantArray[0,vertexCount];listUt={};
tDepth=ConstantArray[0,vertexCount];
tDir=ConstantArray[0,vertexCount];
tDinast={};
DepthFirstScan[UndirectedGraph[g],root,{FrontierEdge->
Function[e,{AppendTo[s,e[[1]]->e[[2]]],
tPred[[e[[2]]]]=e[[1]],
```

```
tDepth[[e[[2]]]]=1+tDepth[[e[[1]]]],
For[i=1,i<=Length[arcsn],i++,
{y,z}=List@@arcsn[[i]];(*Print[{y,z}, | ,e[[1]], | ,e[[2]]];*)
If[e[[1]]==y && e[[2]]==z,(*Print[1 |,{y,z}, | ,e[[1]], | ,e[[2]]];*)
AppendTo[listUt,y->z];tDir[[e[[2]]]]=1;];If[e[[2]]==y && e[[1]]==z,
(*Print[2 |,{y,z}, | ,e[[1]], | ,e[[2]]];*)
AppendTo[listUt,y->z];tDir[[e[[2]]]]=-1;]}],
PrevisitVertex->Function[u,AppendTo[tDinast,u]]}];
Return[s];
];
```

В таблице 4.1 приведены результаты выполнения этой части программы, а именно хранение корневого дерева со всеми необходимыми для построения базиса пространства решений параметрами в памяти компьютера:

Таблица 4.1 – Хранение корневого дерева в памяти компьютера

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

Приведем также код, визуализирующий таблицу с необходимыми структурами данных:

```
Ucycle=List[];
listUn=Complement[arcsn,listUt];
arcUnCount=Length[listUn];
listUt=Sort[listUt];
delta=Flatten[ComputeCharacteristicVectors[listUn,listUt],1];
Print[ ]
TableForm[Table[delta[[i,j]][[2]],{i,1,arcUnCount},{j,1,edgeCount}],
TableHeadings->{listUn,Table[delta[[1,i]][[1]],{i,1,edgeCount}]},
TableAlignments->Center]
```

А также результат построения и визуализации самого корневого дерева на рисунке 4.2:

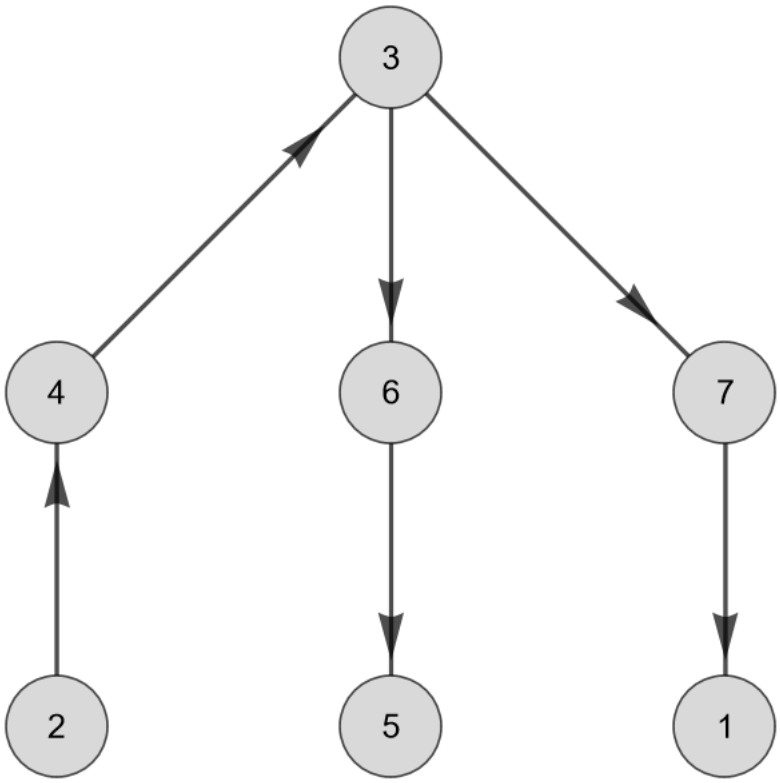

Рисунок 4.2 – Корневое дерево с корнем в узле 2

4.3 Построение базиса пространства решений и частного решения

На основе полученных из построения корневого дерева данных мы можем построить базис пространства решений нашей системы:

Таблица 4.2 – Базис пространства решений

$x_{i\to j}$	$x_{2\to 3}$	$x_{2\to 6}$	$x_{3\to 1}$	$x_{4\to 6}$	$x_{4\to 7}$	$x_{5\to 2}$	$x_{4\to 2}$	$x_{4\to 3}$	$x_{5\to 6}$	$x_{6\to 3}$	$x_{7\to 1}$	$x_{7\to 3}$
2→ 3	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**-1**	**0**	**0**	**0**	**0**
2→ 6	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**-1**	**0**	**1**	**0**	**0**
3→ 1	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**-1**	**1**
4→ 6	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**-1**	**0**	**1**	**0**	**0**
4→ 7	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**-1**	**0**	**0**	**0**	**1**
5→ 2	**0**	**0**	**0**	**0**	**0**	**1**	**-1**	**1**	**-1**	**-1**	**0**	**0**

Приведем код построения базиса пространства решений с помощью вспомогательной определяющей функции "Determining"(2.3):

Реализация функции "Determining":

```
Determining[k_,l_,pred_,dir_,depth_]:=Module[{deltaC={}},
i=k;j=l;
While[i!=j,
If[depth[[i]]>depth[[j]], {
If[dir[[i]]==1,
AppendTo[deltaC,Subscript[x,pred[[i]]->i]->1];
AppendTo[Ucycle,pred[[i]]->i],
AppendTo[deltaC,Subscript[x,i->pred[[i]]]->-1];
AppendTo[Ucycle,i->pred[[i]]]];
i=pred[[i]];},
If[depth[[j]]>depth[[i]],{
If[dir[[j]]==1,
AppendTo[deltaC,Subscript[x,pred[[j]]->j]->-1];
AppendTo[Ucycle,pred[[j]]->j],
AppendTo[deltaC,Subscript[x,j->pred[[j]]]->1];
AppendTo[Ucycle,j->pred[[j]]]];
j=pred[[j]];},{
If[dir[[i]]==1,
AppendTo[deltaC,Subscript[x,pred[[i]]->i]->1];
AppendTo[Ucycle,pred[[i]]->i],
AppendTo[deltaC,Subscript[x,i->pred[[i]]]->-1];
AppendTo[Ucycle,i->pred[[i]]];];
If[dir[[j]]==1,
AppendTo[deltaC,Subscript[x,pred[[j]]->j]->-1];
AppendTo[Ucycle,pred[[j]]->j],
AppendTo[deltaC,Subscript[x,j->pred[[j]]]->1];
AppendTo[Ucycle,j->pred[[j]]];];
i=pred[[i]];
j=pred[[j]];}]];];
Return[deltaC];];
```

Построение базиса пространства решений:

```
Ucycle=List[];
ComputeCharacteristicVectors[lUn_,lUt_]:=Module[{cv={}},
dN0=Subscript[x,#1]->0&/@lUn;
For[n=1,n<=Length[lUn],n++,
arc=lUn[[n]];
dN=dN0; dN[[n]]=Subscript[x,arc]->1;
dTc=Determining[arc[[1]],arc[[2]],tPred,tDir,tDepth];
```

```
listUt0=Complement[lUt,Ucycle];Ucycle=List[];
dT0=Subscript[x,#1]->0&/@listUt0;
dGeneral=Join[dN,Sort[Join[dTc,dT0]]];
AppendTo[cv,{dGeneral}];];
Return[cv];
];
```

Для нахождения общего решения системы будем использовать тот факт, что общее решение неоднородного уравнения являет собой сумму общего решения соответствующих ему однородного и частного решений.

Общее решение однородной части получаем из таблицы 4.2.

```
xgeneral=Table[Subscript[x,listUtsort[[j]]]->
(xpart[[arcUnCount+j]][[2]]+Sum[deltaT[[j,i]][[2]]Subscript[x,listUn[[i]]],
{i,1,edgeCount-vertexCount+1}]),{j,vertexCount-1}];
```

Подставляем нули в небазисные переменные и из исходной системы получаем некоторое частное решение:

$$\begin{array}{cccc} \tilde{x}_{2,3}=0 & \tilde{x}_{2,6}=0 & \tilde{x}_{3,1}=0 & \tilde{x}_{4,6}=0 \\ \tilde{x}_{4,7}=0 & \tilde{x}_{5,2}=0 & \tilde{x}_{4,2}=-5 & \tilde{x}_{4,3}=-3 \\ \tilde{x}_{5,6}=-2 & \tilde{x}_{6,3}=4 & \tilde{x}_{7,1}=5 & \tilde{x}_{7,3}=2 \end{array} \tag{4.1}$$

Приведем модуль программы для нахождения частного решения:

```
ComputeParticularSolution[lUn_,b_,pred_,dinast_,dir_]:=Module[{},
xpN0=Subscript[x̃, #1] ->0&/@lUn;
xpUt = List[];
xp = ConstantArray[0, vertexCount];
For[n = vertexCount, n > 1, n--,i = dinast[[n]];
xp[[i]]+= − dir[[i]]b[[i]];
xp[[pred[[i]]]]+=dir[[i]]dir[[pred[[i]]]]xp[[i]];
];
```

4.4 Построение общего решения системы с матрицей инцедентности графа

Остается только взять переменные, соответствующие базисным элементам матрицы, представленной на таблице 4.2, и выразить их через небазисные. После чего сложить с полученным частным решением. С этой тривиальной задачей вполне справится фукнция Wolfram Mathematica – **Solve**.

После выполнения данных операций получаем:

$$\begin{gathered} x_{4,2} = x_{2,3} + x_{2,6} - x_{5,2} - 5 \\ x_{4,3} = -x_{2,3} - x_{2,6} - x_{4,6} - x_{4,7} + x_{5,2} - 3 \\ x_{5,6} = -x_{5,2} - 2 \\ x_{6,3} = x_{2,6} + x_{4,6} - x_{5,2} + 1 \\ x_{7,1} = -x_{3,1} + 5 \\ x_{7,3} = x_{3,1} + x_{4,7} + 2 \end{gathered} \tag{4.2}$$

4.5 Численные эксперименты

Сравним данный метод решения, с решением, получаемым средствами Wolfram Mathematica, по скорости выполнения:

Таблица 4.3 – Численные эксперименты

Время работы алгоритма (сек)	Размерность графа $S = (I,U)$		
	$\lvert I\rvert = 100$, $\lvert U\rvert = 300$	$\lvert I\rvert = 150$, $\lvert U\rvert = 400$	$\lvert I\rvert = 200$, $\lvert U\rvert = 500$
t1	0.377	0.83	1.848
t2	36.74	68.99	142.12
t2 / t1	97.46	83.12	76.9

По результатами исследования, приведенным в таблице 4.3, видно, что представленный метод решения выигрывает у методов WolframMathematica более чем на 2 порядка.

ГЛАВА 5
ПРИМЕР ДЕКОМПОЗИЦИИ МУЛЬТИПОТОКА

5.1 Входные данные задачи

Рассмотрим пример декомпозиции мультипотока в задаче неоднородного линейного потокового программирования вида (1.2) с ограничениями второго (1.3) и третьего (1.4) типа. Будем рассматривать систему $S = (I, U)$, где I – множество вершин орграфа, U – множество его дуг трех различных типов потока, то есть $(i, j)^k \in U, k = \overline{1,3}(|K| = 3)$.

Запишем конкретный состав множеств I и U:

$$I = \{1, 2, 3, 4, 5\},$$

$$U = \{\{1,2\}^1, \{4,2\}^1\,\{5,1\}^1, \{5,2\}^1, \{5,4\}^1,$$
$$\{1,2\}^2, \{2,3\}^3, \{4,1\}^2, \{4,3\}^2, \{5,4\}^2\},$$
$$\{1,2\}^3, \{4,1\}^3, \{4,2\}^3, \{4,3\}^3, \{5,1\}^3, \{5,4\}^3\}.$$

Итого система имеет следующий вид:

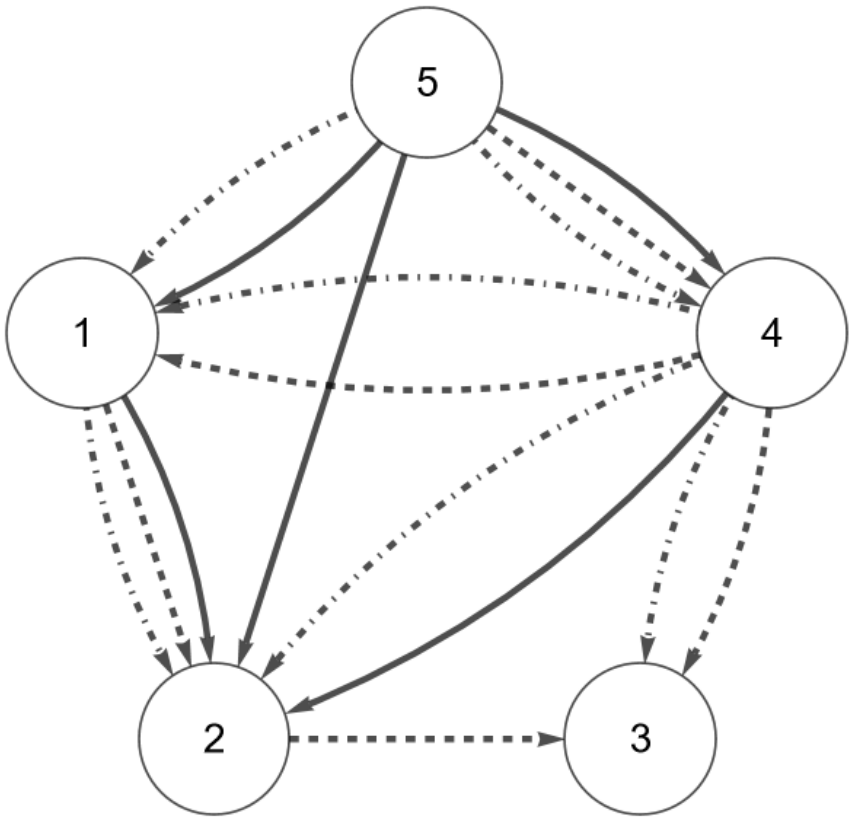

Рисунок 5.1 – Исходная мультисеть

Приведем код визуализации графа, представленного на рис. 5.1:

```
EdgeTaggedGraph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 5->2, 5->1, Style[1->2, Dashed], Style
[2->3, Dashed], Style[4->3, Dashed], Style[5->4, Dashed],
Style[4->1, Dashed], Style[1->2, DotDashed], Style[4->3, DotDashed],
Style[4->2, DotDashed], Style[4->1, DotDashed],
Style[5->4, DotDashed], Style[5->1, DotDashed]}, VertexLabels->
Placed["Name", Center], VertexStyle->White, VertexSize->0.35,
VertexLabelStyle->Directive[Black, 17], GraphLayout->
"CircularEmbedding", GraphLayout->"CircularEmbedding",
EdgeStyle->Directive[Black, Thickness[0.0075]], DirectedEdges->True]
```

Разобьем данную систему на подсистемы $S^k = (I^k, U^k)$ с различными типами потока $k = \overline{1,3}$ таким образом, что выполняется (1.1):

$$I^1 = \{1,2,4,5\}, I^2 = I^3 = \{1,2,3,4,5\}.$$

$$U^1 = \{\{1,2\}^1, \{4,2\}^1\, \{5,1\}^1, \{5,2\}^1, \{5,4\}^1\}$$
$$U^2 = \{\{1,2\}^2, \{2,3\}^3, \{4,1\}^2, \{4,3\}^2, \{5,4\}^2\}\}$$
$$U^3 = \{\{1,2\}^3, \{4,1\}^3, \{4,2\}^3, \{4,3\}^3, \{5,1\}^3, \{5,4\}^3\}$$

Составим для данного примера систему ограничений. Распишем условия баланса вида (1.2):

$$\begin{gathered}
\begin{gathered} x_{12}^1 + x_{51}^1 = 3, \\ -x_{12}^1 - x_{52}^1 - x_{42}^1 = -15, \\ x_{42}^1 - x_{54}^1 = -8, \\ x_{51}^1 + x_{52}^1 + x_{54}^1 = 20, \end{gathered} \qquad \begin{gathered} x_{12}^2 - x_{41}^2 = 0, \\ -x_{12}^2 + x_{23}^2 = -6, \\ -x_{23}^2 - x_{43}^2 = -5, \\ x_{41}^2 + x_{43}^2 - x_{54}^2 = 7, \\ x_{54}^2 = 4, \end{gathered} \\
\begin{gathered} x_{12}^3 - x_{41}^3 - x_{51}^3 = -10, \\ -x_{12}^3 - x_{42}^3 = -10, \\ -x_{43}^3 = -7, \\ x_{41}^3 + x_{42}^3 + x_{43}^3 - x_{54}^3 = 9, \\ x_{51}^3 + x_{54}^3 = 18. \end{gathered}
\end{gathered} \tag{5.1}$$

Далее дополним полученную систему (5.1) следующими уравнениями вида (1.3):

$$
\begin{gathered}
7x_{12}^1 + 4x_{12}^2 + 2x_{12}^3 + 8x_{23}^2 + 9x_{41}^2 + 9x_{41}^3 + 9x_{42}^1 + 4x_{42}^3 + 2x_{43}^2 + \\
+5x_{43}^3 + 9x_{54}^1 + 3x_{54}^2 + 7x_{54}^3 = 556, \\
6x_{12}^1 + 7x_{12}^2 + 6x_{12}^3 + 3x_{23}^2 + 7x_{41}^2 + 5x_{42}^1 + 2x_{42}^3 + 2x_{43}^3 + 10x_{51}^1 + \\
+7x_{51}^3 + 8x_{52}^1 + 9x_{54}^1 + 5x_{54}^2 + 7x_{54}^3 = 501, \\
2x_{12}^1 + x_{12}^2 + 5x_{12}^3 + 5x_{23}^2 + 2x_{41}^1 + 6x_{41}^3 + 3x_{42}^1 + 3x_{43}^2 + 2x_{43}^3 + 8x_{51}^1 + \\
+x_{51}^3 + 8x_{52}^1 + 2x_{54}^1 + 4x_{54}^2 + 8x_{54}^3 = 307.
\end{gathered}
\tag{5.2}
$$

И, наконец, добавим одно ограничение типа (1.4):

$$
x_{54}^1 + x_{54}^3 = 13. \tag{5.3}
$$

Таким образом постановка исходной задачи завершена.

5.2 Применение методов и алгоритмов декомпозиции

Представим каждую из подсистем в виде орграфа и приведем код их визуализации:

```
Graph[{1,2,3,4,5},{1->2,4->2,5->4,5->1,5->2},VertexLabels->
Placed[“Name”,Center],VertexStyle->White,VertexSize->0.35,GraphLayout->
“CircularEmbedding”,EdgeStyle->
Directive[Black, Thickness[0.0075]], VertexLabelStyle->Directive[Black, 17]]
Graph[{1, 2, 3, 4, 5}, {1->2, 2->3, 4->3, 5->4, 4->1},
VertexLabels->Placed[“Name”, Center], VertexStyle->
White, VertexSize->0.35, GraphLayout->“CircularEmbedding”,
EdgeStyle->Directive[Dashed, Black, Thickness[0.0075]], VertexLabelStyle->
Directive[Black, 17]]
Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 4->1, 4->3, 5->1},
VertexLabels->Placed[“Name”, Center], VertexStyle->White, VertexSize->0.35,
GraphLayout->“CircularEmbedding”,
EdgeStyle->{Directive[DotDashed, Black, Thickness[0.008]]},
VertexLabelStyle->Directive[Black, 17]]
```

Результат выполнения данного сегмента (изображение графов каждой из подсистем):

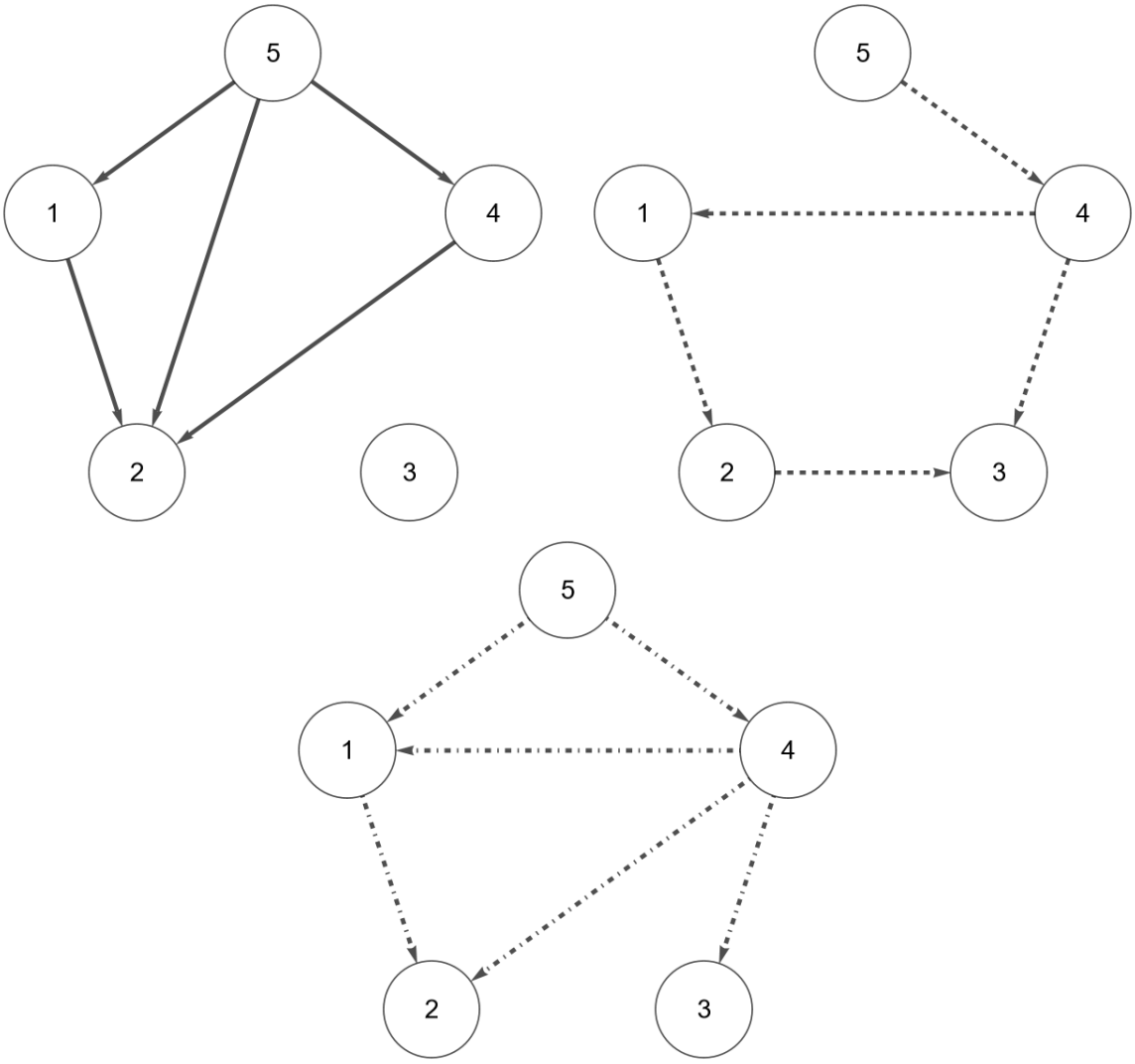

Рисунок 5.2 – Коллекция подграфов мультиграфа, представленного на рисунке 5.1

Для каждого из изображенных на рисунке 5.2 графов выделим по остовному дереву для нахождения опоры сети и дальнейшего определения базиса пространства решений. Для нахождения остовного дерева в каждой из подсетей можно воспользоваться алгоритмами, описанными в [7], в данном случае будем считать, что остовные деревья уже были найдены.

Приведем код визуализации каждого из остовных деревьев для подсетей $S^k, k = \overline{1,3}$:

t1 = TreePlot[{1->2, 4->2, 5->4}, Top, 1, VertexStyle->White, VertexSize->0.35,
VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Black, Thickness[0.0075]],
VertexLabelStyle->Directive[Black, 17]];
t2 = Graph[{3}, {}, VertexSize->0.035, VertexStyle->White, VertexLabels
->Placed["Name", Center], EdgeStyle->Directive[Black, Thickness[0.0075]],
VertexLabelStyle->Directive[Black, 17]];

```
GraphicsRow[{t1, t2}]
TreePlot[{1->2,2->3,4->3,5->4},Top,1,VertexSize->0.35, VertexStyle->White,
VertexLabels->Placed["Name",Center],
DirectedEdges->True,EdgeStyle->Directive[Dashed,Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]]
TreePlot[{1->2,4->2,4->3,5->4},Top,1,VertexSize->0.35, VertexStyle->White,
VertexLabels->Placed["Name",Center],
DirectedEdges->True,EdgeStyle->Directive[DotDashed,Black, Thickness[0.02]],
VertexLabelStyle->Directive[Black, 17]]
```

Собственно результат визуализации остовных деревьев:

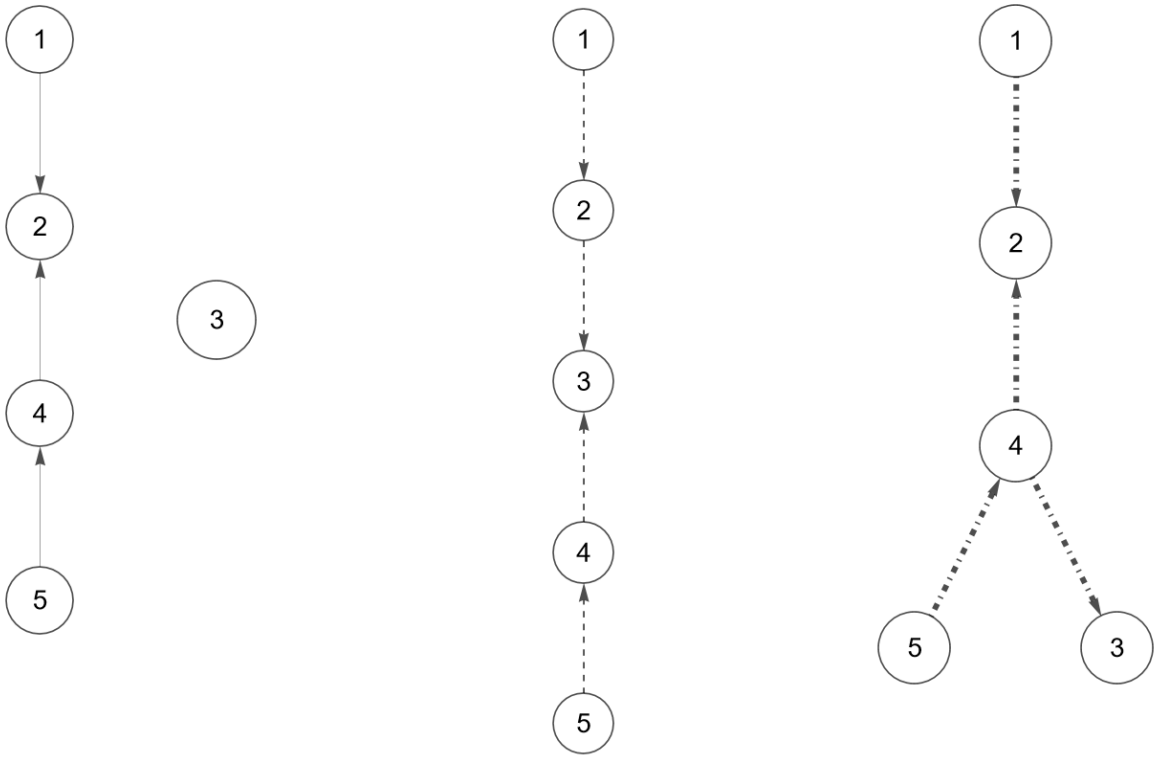

Рисунок 5.3 – Полученные остовные деревья

На основании представленных на рисунке 5.3 остовных деревьев строится вспомогательные структуры данных – корневые деревья, собственно получаемые корневые деревья:

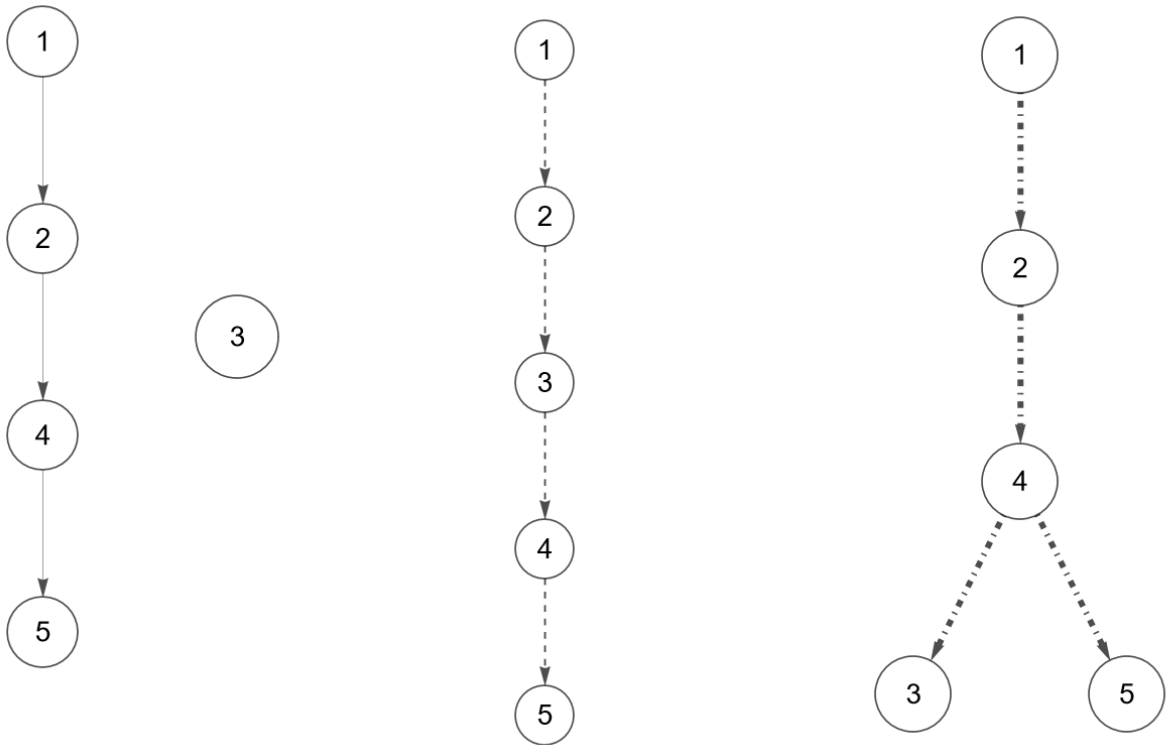

Рисунок 5.4 – Полученные корневые деревья с корнями {1, 1, 1} соответственно

Хранить информацию об изображенных на рисунке 5.4 деревьях будем в массивах в порядке обхода корневых деревьев алгоритмом поиска в глубину начиная с корня. Для нахождения характеристических векторов (решений систем (1.7)) в соответствии с (1.9) и (1.10) нам необходимо знать инвертированный массив порядка династического обхода с названием thread, массив предков каждого из элементов дерева с названием pred и массив исходных направлений дуг dir.

Таблица 5.1 – Вспомогательные структуры данных первой подсистемы

i	1	2	3	4	5
thread[i]	5	4	2	1	-
pred[i]	0	1	0	2	4
dir[i]	0	1	0	-1	-1

Таблица 5.2 – Вспомогательные структуры данных второй подсистемы

i	1	2	3	4	5
thread[i]	5	4	3	2	1
pred[i]	0	1	2	3	4
dir[i]	0	1	1	-1	-1

Таблица 5.3 – Вспомогательные структуры данных третьей подсистемы

i	1	2	3	4	5
thread[i]	5	3	4	2	1
pred[i]	0	1	4	2	4
dir[i]	0	1	1	-1	-1

Будем решать уравнения по очереди в порядке, задаваемым массивом thread, такой подход гарантирует, что в каждом из уравнений последовательно будет ровно одна неизвестная компонента. Значения используемых структур данных представлены в таблицах 5.1–5.3.

Приведем код, реализующий данную часть программы для $k = 2$:

```
(*thread*)
t = {5, 4, 3, 2, 1};
(*pred*)
p = {0, 1, 2, 3, 4};
(*dir*)
d = {0, 1, 1, −1, −1};
system2a = system2;
system2a[[4]] = system2a[[4]]/. {x2_{4,1} → 1} ;
system2a[[1]] = system2a[[1]]/. {x2_{4,1} → 1} ;
δ2_{4,1} = {x2_{4,1} → 1} ;
k2 = 4;

For[i = 1, i<=k2, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system2a[[t[[i]]]], x2_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system2a[[t[[i]]]], x2_{t[[i]],p[[t[[i]]]]}] [[1]]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system2a[[p[[t[[i]]]]]] = system2a[[p[[t[[i]]]]]/.δ];
δ2_{4,1} = Join [δ2_{4,1}, δ] ;
}];
```

Аналогичные программные модули реализованы и для других двух подсистем.

Для финального этапа выполнения программы необходимо предварительно также найти некоторое частное решение. Немаловажная часть эффективности полученного алгоритма заключается в том, что найти частное решения можно на всё тех же построенных структурах данных.

Приведем код построения частного решения:

```
(*thread*)
t = {5, 4, 3, 2, 1};
(*pred*)
p = {0, 1, 4, 2, 4};
(*dir*)
d = {0, 1, 1, −1, −1};
system3a = system3;
system3a[[4]] = system3a[[4]]/. {x3_{4,1} → 0} ;
system3a[[1]] = system3a[[1]]/. {x3_{4,1} → 0} ;
system3a[[5]] = system3a[[5]]/. {x3_{5,1} → 0} ;
system3a[[1]] = system3a[[1]]/. {x3_{5,1} → 0} ;
δ3 = {x3_{4,1} → 0, x3_{5,1}->0} ;

For[i = 1, i<=k3, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system3a[[t[[i]]]], x3_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system3a[[t[[i]]]], x3_{t[[i]],p[[t[[i]]]]}] [[1]]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system3a[[p[[t[[i]]]]]] = system3a[[p[[t[[i]]]]]/.δ];
δ3 = Join [δ3, δ] ;
}];
```

Собственно полученное частное решение:

$$\tilde{x}^3_{41} = 0, \tilde{x}^3_{51} = 0, \tilde{x}^3_{54} = 18, \tilde{x}^3_{43} = 7, \tilde{x}^3_{42} = 20, \tilde{x}^3_{12} = -10.$$

Можно подставить значения в третью подсистему системы (5.1) и убедиться в их верности, что программно в Wolfram Mathematica реализуется с помощью набора команд: **Print [Simplify [system3/.δ3]] ;**

Аналогичным образом получаем частные решения оставшихся двух подсистем (полную версию кода можно найти в приложении А):

$$\tilde{x}^1_{51} = 0, \tilde{x}^1_{52} = 0, \tilde{x}^1_{54} = 20, \tilde{x}^1_{42} = 12, \tilde{x}^1_{12} = 3.$$

$$\tilde{x}^2_{41} = 0, \tilde{x}^2_{54} = 4, \tilde{x}^2_{43} = 11, \tilde{x}^2_{23} = -6, \tilde{x}^2_{12} = 0.$$

После того, как были получены все необходимые сведения о системе и произведены соотвествующие вычисления, можно приступать к финальному этапу выполнения, а именно к формированию и решению системы (2.10).

5.3 Обратное преобразование и результаты

Для нахождения элементов матрицы R, а также правой части β воспользуемся формулами (2.11), приведем код подстановки полученных значений для нахождения вектора β по формулам:

$$A_1 = 556 - (7\mathbf{x1}_{1,2} + 4\mathbf{x2}_{1,2} + 2\mathbf{x3}_{1,2} + 8\mathbf{x2}_{2,3} + 9\mathbf{x2}_{4,1} + 9\mathbf{x3}_{4,1} + 9\mathbf{x1}_{4,2} + 4\mathbf{x3}_{4,2} + 2\mathbf{x2}_{4,3} + 5\mathbf{x3}_{4,3} + 9\mathbf{x1}_{5,1} + 9\mathbf{x3}_{5,1} + 5\mathbf{x1}_{5,2} + 9\mathbf{x1}_{5,4} + 3\mathbf{x2}_{5,4} + 7\mathbf{x3}_{5,4}/.\overline{\delta 1}/.\overline{\delta 2}/.\overline{\delta 3};$$

$$A_2 = 501 - (6\mathbf{x1}_{1,2} + 7\mathbf{x2}_{1,2} + 6\mathbf{x3}_{1,2} + 3\mathbf{x2}_{2,3} + 7\mathbf{x2}_{4,1} + 5\mathbf{x1}_{4,2} + 2\mathbf{x3}_{4,2} + 2\mathbf{x3}_{4,3} + 10\mathbf{x1}_{5,1} + 7\mathbf{x3}_{5,1} + 8\mathbf{x1}_{5,2} + 9\mathbf{x1}_{5,4} + 5\mathbf{x2}_{5,4} + 7\mathbf{x3}_{5,4}) /.\overline{\delta 1}/.\overline{\delta 2}/.\overline{\delta 3};$$

$$A_3 = 307 - (2\mathbf{x1}_{1,2} + \mathbf{x2}_{1,2} + 5\mathbf{x3}_{1,2} + 5\mathbf{x2}_{2,3} + 2\mathbf{x2}_{4,1} + 6\mathbf{x3}_{4,1} + 3\mathbf{x1}_{4,2} + 3\mathbf{x2}_{4,3} + 2\mathbf{x3}_{4,3} + 8\mathbf{x1}_{5,1} + \mathbf{x3}_{5,1} + 8\mathbf{x1}_{5,2} + 2\mathbf{x1}_{5,4} + 4\mathbf{x2}_{5,4} + 8\mathbf{x3}_{5,4})/.\overline{\delta 1}/.\overline{\delta 2}/.\overline{\delta 3};$$

$$A_4 = 13 - (\mathbf{x1}_{5,4} + \mathbf{x3}_{5,4}) /.\overline{\delta 1}/.\overline{\delta 2}/.\overline{\delta 3};$$

$$\beta_1 = A_1 - \left(\mathbf{R13}_{5,1}\mathbf{y3}_{5,1}\right);$$
$$\beta_2 = A_2 - \left(\mathbf{R23}_{5,1}\mathbf{y3}_{5,1}\right);$$
$$\beta_3 = A_3 - \left(\mathbf{R33}_{5,1}\mathbf{y3}_{5,1}\right);$$
$$\beta_4 = A_4 - \left(\mathbf{R43}_{5,1}\mathbf{y3}_{5,1}\right);$$

Приведем также код нахождения элементов матрицы R:

$$\mathbf{R11}_{5,1} = 7\mathbf{x1}_{1,2} + 9\mathbf{x1}_{4,2} + 9\mathbf{x1}_{5,1} + 9\mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,1};$$
$$\mathbf{R21}_{5,1} = 6\mathbf{x1}_{1,2} + 5\mathbf{x1}_{4,2} + 10\mathbf{x1}_{5,1} + 9\mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,1};$$
$$\mathbf{R31}_{5,1} = 2\mathbf{x1}_{1,2} + 3\mathbf{x1}_{4,2} + 8\mathbf{x1}_{5,1} + 2\mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,1};$$
$$\mathbf{R41}_{5,1} = 0\mathbf{x1}_{1,2} + 0\mathbf{x1}_{4,2} + 0\mathbf{x1}_{5,1} + \mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,1};$$

$$\mathbf{R11}_{5,2} = 7\mathbf{x1}_{1,2} + 9\mathbf{x1}_{4,2} + 5\mathbf{x1}_{5,2} + 9\mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,2};$$
$$\mathbf{R21}_{5,2} = 6\mathbf{x1}_{1,2} + 5\mathbf{x1}_{4,2} + 8\mathbf{x1}_{5,2} + 9\mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,2};$$
$$\mathbf{R31}_{5,2} = 2\mathbf{x1}_{1,2} + 3\mathbf{x1}_{4,2} + 8\mathbf{x1}_{5,2} + 2\mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,2};$$
$$\mathbf{R41}_{5,2} = 0\mathbf{x1}_{1,2} + 0\mathbf{x1}_{4,2} + 0\mathbf{x1}_{5,2} + \mathbf{x1}_{5,4}/.\delta\mathbf{1}_{5,2};$$

$$\mathbf{R12}_{4,1} = 4\mathbf{x2}_{1,2} + 8\mathbf{x2}_{2,3} + 9\mathbf{x2}_{4,1} + 2\mathbf{x2}_{4,3} + 3\mathbf{x2}_{5,4}/.\delta\mathbf{2}_{4,1};$$
$$\mathbf{R22}_{4,1} = 7\mathbf{x2}_{1,2} + 3\mathbf{x2}_{2,3} + 7\mathbf{x2}_{4,1} + 0\mathbf{x2}_{4,3} + 5\mathbf{x2}_{5,4}/.\delta\mathbf{2}_{4,1};$$
$$\mathbf{R32}_{4,1} = \mathbf{x2}_{1,2} + 5\mathbf{x2}_{2,3} + 2\mathbf{x2}_{4,1} + 3\mathbf{x2}_{4,3} + 4\mathbf{x2}_{5,4}/.\delta\mathbf{2}_{4,1};$$
$$\mathbf{R42}_{4,1} = 0\mathbf{x2}_{1,2} + 0\mathbf{x2}_{2,3} + 0\mathbf{x2}_{4,1} + 0\mathbf{x2}_{4,3} + 0\mathbf{x2}_{5,4}/.\delta\mathbf{2}_{4,1};$$

$$\mathbf{R13}_{4,1} = 2\mathbf{x3}_{1,2} + 9\mathbf{x3}_{4,1} + 4\mathbf{x3}_{4,2} + 5\mathbf{x3}_{4,3} + 7\mathbf{x3}_{5,4}/.\delta\mathbf{3}_{4,1};$$
$$\mathbf{R23}_{4,1} = 6\mathbf{x3}_{1,2} + 0\mathbf{x3}_{4,1} + 2\mathbf{x3}_{4,2} + 2\mathbf{x3}_{4,3} + 7\mathbf{x3}_{5,4}/.\delta\mathbf{3}_{4,1};$$

$$\mathbf{R33}_{4,1} = 5\mathbf{x3}_{1,2} + 6\mathbf{x3}_{4,1} + 0\mathbf{x3}_{4,2} + 2\mathbf{x3}_{4,3} + 8\mathbf{x3}_{5,4}/.\delta 3_{4,1};$$
$$\mathbf{R43}_{4,1} = 0\mathbf{x3}_{1,2} + 0\mathbf{x3}_{4,1} + 0\mathbf{x3}_{4,2} + 0\mathbf{x3}_{4,3} + \mathbf{x3}_{5,4}/.\delta 3_{4,1};$$

$$\mathbf{R13}_{5,1} = 2\mathbf{x3}_{1,2} + 4\mathbf{x3}_{4,2} + 5\mathbf{x3}_{4,3} + 9\mathbf{x3}_{5,1} + 7\mathbf{x3}_{5,4}/.\delta 3_{5,1};$$
$$\mathbf{R23}_{5,1} = 6\mathbf{x3}_{1,2} + 2\mathbf{x3}_{4,2} + 2\mathbf{x3}_{4,3} + 7\mathbf{x3}_{5,1} + 7\mathbf{x3}_{5,4}/.\delta 3_{5,1};$$
$$\mathbf{R33}_{5,1} = 5\mathbf{x3}_{1,2} + 0\mathbf{x3}_{4,2} + 2\mathbf{x3}_{4,3} + \mathbf{x3}_{5,1} + 8\mathbf{x3}_{5,4}/.\delta 3_{5,1};$$
$$\mathbf{R43}_{5,1} = 0\mathbf{x3}_{1,2} + 0\mathbf{x3}_{4,2} + 0\mathbf{x3}_{4,3} + 0\mathbf{x3}_{5,1} + \mathbf{x3}_{5,4}/.\delta 3_{5,1};$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{R11}_{5,1} & \mathbf{R11}_{5,2} & \mathbf{R12}_{4,1} & \mathbf{R13}_{4,1} \\ \mathbf{R21}_{5,1} & \mathbf{R21}_{5,2} & \mathbf{R22}_{4,1} & \mathbf{R23}_{4,1} \\ \mathbf{R31}_{5,1} & \mathbf{R31}_{5,2} & \mathbf{R32}_{4,1} & \mathbf{R33}_{4,1} \\ \mathbf{R41}_{5,1} & \mathbf{R41}_{5,2} & \mathbf{R42}_{4,1} & \mathbf{R43}_{4,1} \end{pmatrix};$$

Остается только выполнить обратное преобразование в соответствии с (2.12):

$$Y = \mathbf{Simplify}\left[\mathbf{Inverse}[\mathbf{R}].\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}\right];$$

Находить матрицу обратного преобразования будет встроенная функция Wolfram Mathematica "**Inverse**". Данная функция, согласно [18], реализует эффективный алгоритм обратного преобразования матрицы, а значит идеально подходит для наших целей.

Итого, реализованный в системе компьютерной алгебры Wolfram Mathematica программный модуль решает указанный пример, возвращая следующий ответ:

$$\left\{x1_{5,1} \to -\frac{2}{359}\left(-6406 + 323\,y3_{5,1}\right),\ x1_{5,2} \to \frac{1}{359}\left(-3837 + 287\,y3_{5,1}\right),\right.$$
$$x2_{4,1} \to \frac{1}{359}\left(73 + 30\,y3_{5,1}\right),\ x3_{4,1} \to \frac{1}{359}\left(-1612 + 267\,y3_{5,1}\right),\ x3_{5,1} \to y3_{5,1},$$
$$x1_{1,2} \to -\frac{323}{359}\left(-43 + 2\,y3_{5,1}\right),\ x1_{4,2} \to -13 + y3_{5,1},\ x1_{5,4} \to -5 + y3_{5,1},$$
$$x2_{1,2} \to \frac{1}{359}\left(73 + 30\,y3_{5,1}\right),\ x2_{2,3} \to \frac{1}{359}\left(-2081 + 30\,y3_{5,1}\right),$$
$$x2_{4,3} \to -\frac{6}{359}\left(-646 + 5\,y3_{5,1}\right),\ x2_{5,4} \to 4,\ x3_{1,2} \to \frac{2}{359}\left(-2601 + 313\,y3_{5,1}\right),$$
$$\left.x3_{4,2} \to -\frac{2}{359}\left(-4396 + 313\,y3_{5,1}\right),\ x3_{4,3} \to 7,\ x3_{5,4} \to 18 - y3_{5,1}\right\}$$

Рисунок 5.5 – Полученный ответ

Также как и ранее в случае с решениями подсистем, с помощью функции "**Symplify**"можно убедиться в правильности полученного ответа.

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломной работы были получены следующие результаты:

1. Были изучены основные понятия представленной к изучению темы. Такие как декомпозиция, сетевая часть системы, опора и многие другие сведения о разреженных недоопределенных системах линейных алгебраических уравнений.
2. Углублены знания системы компьютерной алгебры Wolfram Mathematica.
3. Были улучшены методы решения поставленных задач.
4. Поставленная задача была экстраполированна на случай с различными типами потока.
5. Было произведено численное сравнение полученных методов решения с методами, предлагаемыми Wolfram Mathematica.
6. На основе полученной в курсовом проекте базы была реализована быстродействennая программа нахождения общего решения.
7. На основе сформированной теоретической базы была реализована программа декомпозиции мультипотока.
8. Разработаны и реализованы алгоритмы, структуры данных и технологии нахождения оптимального решения.
9. Показано, что на практике можно представить объекты и проблемы реального мира в виде полученных математических моделей.
10. Отчет выполенен в системе помпьютерной верстки LaTex.

СПИСОК ИСТОЧНИКОВ

1. Альсевич В. В. Оптимизация линейных экономических моделей. Статические задачи. Учеб. пособие. / В. В. Альсевич, Р. Габасов, В. С. Глушенков — Минск, 2000. - 240 с.
2. Габасов Р. Конструктивные методы оптимизации. Ч.3. Сетевые задачи / Р. Габасов, Ф.М. Кириллова – Мн.: Университетское, 1986. – 222 с.
3. Габасов Р. Методы линейного программирования. Ч.1. Общие задачи. / Р. Габасов, Ф. М. Кириллова — Минск, 1977. - 176 с.
4. Габасов Р. Методы линейного программирования. Ч.2. Транспортные задачи. / Р. Габасов, Ф. М. Кириллова — Минск, 1978. - 243 с.
5. Габасов Р. Методы линейного программирования. Ч.3. Специальные задачи. / Р. Габасов, Ф. М. Кириллова, О. И. Костюкова — Минск, 1980. - 368 с.
6. Задачи линейной оптимизации с неточными данными / М. Фидлер, Й. Недома, Я. Рамик [и др.] ; пер. с англ. С.И. Кумкова ; под ред. С.П. Шарого Москва; Ижевск : Институт компьютерных исследований : Регуляр. и хаотич. динамика, 2008. – 286 с.
7. Котов, В.М. Алгоритмы и структуры данных: учеб. пособие / В.М. Котов, Е.П. Соболевская, А.А. Толстиков. – Минск: БГУ, 2011. – 267 с.
8. Пилипчук, Л. А. Линейные неоднородные задачи потокового программирования / Л. А. Пилипчук – Минск: БГУ, 2009. – 222 с.
9. Пилипчук, Л. А. Разреженные недоопределенные системы линейных алгебраических уравнений / Л. А. Пилипчук – Минск: БГУ, 2012. – 260 с.
10. Романовский, И. В. Субоптимальные решения / И. В. Романовский — Петрозаводск: Издательство Петрозаводского университета, 1998. - 96 с.
11. Романовский И.В. Перебор субоптимальных решений в задачах дискретной оптимизации. // Компьютерные инструменты в образовании. – ЛЭТИ С.-Петербург– 2012.– №6. – С. 25-34.
12. Сафронов Э.А. Транспортные системы городов и регионов : Учебное пособие / Э.А. Сафронов – Издательство АСВ, - Москва, 2013. - 288 с.
13. Ahuja R. K. Network Flows: Theory, Algorithms, and Applications / R. K. Ahuja, T. L. Magnanti, J. B. Orlin — New Jersey, 1993. - 863 p.
14. Bianco L. A network based model for traffic sensor location with implication in O/D matrix estimates. Transportation Science / Bianco L., Confessore G., Reverberi P. // Transportation Science Journal. – 2001. – Vol. 35, N 1. – P. 50-60.

15. Pilipchuk L. A. About multiflow decomposition methods in fractional linear programming problems / Pilipchuk L. A., Romanovsky Y. V., Hurinovich V. V. // ITS-2024, 2024 - P. 133-134.
16. Pilipchuk L.A. Algorithms for construction of optimal and suboptimal solutions in network optimization problems / L.A. Pilipchuk, A.S. Pilipchuk, Y.H. Pesheva // International Journal of Pure and Applied Mathematics (IJPAM). – 2009. – Vol. 54, N 2. – P. 193-205.
17. Pilipchuk L.A. Sparse Linear Systems and Their Applications / L. A. Pilipchuk – Minsk : BSU, 2013. - 235 p.
18. Wolfram: Computation Meets Knowledge [Electronic resourse]. – Mode of access: https://reference.wolfram.com/language/ref/Solve.html. – Date access: 11.03.2025.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Пример декомпозиции мультипотока

EdgeTaggedGraph[{1, 2, 3, 4, 5},

{1->2, 4->2, 5->4, 5->2, 5->1, Style[1->2, Dashed], Style

[2->3, Dashed], Style[4->3, Dashed], Style[5->4, Dashed],

Style[4->1, Dashed], Style[1->2, DotDashed], Style[4->3, DotDashed],

Style[4->2, DotDashed], Style[4->1, DotDashed],

Style[5->4, DotDashed], Style[5->1, DotDashed]}, VertexLabels->

Placed["Name", Center], VertexStyle->White, VertexSize->0.35,

VertexLabelStyle->Directive[Black, 17], GraphLayout->

"CircularEmbedding", GraphLayout->"CircularEmbedding",

EdgeStyle->Directive[Black, Thickness[0.0075]], DirectedEdges->True]

System = {

$x1_{1,2} - x1_{5,1} == 3$,

$-x1_{1,2} - x1_{5,2} - x1_{4,2} == -15$,

$x1_{4,2} - x1_{5,4} == -8$,

$x1_{5,1} + x1_{5,4} + x1_{5,2} == 20$,

$x2_{1,2} - x2_{4,1} == 0$,

$-x2_{1,2} + x2_{2,3} == -6$,

$-x2_{2,3} - x2_{4,3} == -5$,

$x2_{4,1} + x2_{4,3} - x2_{5,4} == 7$,

$x2_{5,4} == 4$,

$x3_{1,2} - x3_{4,1} - x3_{5,1} == -10$,

$-x3_{1,2} - x3_{4,2} == -10$,

$-x3_{4,3} == -7$,

$x3_{4,1} + x3_{4,3} + x3_{4,2} - x3_{5,4} == 9$,

$x3_{5,1} + x3_{5,4} == 18$,

$7x1_{1,2} + 4x2_{1,2} + 2x3_{1,2} + 8x2_{2,3} + 9x2_{4,1} + 9x3_{4,1} + 9x1_{4,2}$

```
+4x3_{4,2} + 2x2_{4,3} + 5x3_{4,3} + 9x1_{5,1} + 9x3_{5,1} + 5x1_{5,2} + 9x1_{5,4} + 3x2_{5,4} + 7x3_{5,4}==556,
6x1_{1,2} + 7x2_{1,2} + 6x3_{1,2} + 3x2_{2,3} + 7x2_{4,1} + 5x1_{4,2} + 2x3_{4,2} + 2x3_{4,3}
+10x1_{5,1} + 7x3_{5,1} + 8x1_{5,2} + 9x1_{5,4} + 5x2_{5,4} + 7x3_{5,4}==501,

2x1_{1,2} + x2_{1,2} + 5x3_{1,2} + 5x2_{2,3} + 2x2_{4,1} + 6x3_{4,1} + 3x1_{4,2} + 3x2_{4,3}
+2x3_{4,3} + 8x1_{5,1} + x3_{5,1} + 8x1_{5,2} + 2x1_{5,4} + 4x2_{5,4} + 8x3_{5,4}==307,
x1_{5,4} + x3_{5,4}==13
};

Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 5->1},
VertexLabels->Placed["Name", Center], VertexStyle->White, VertexSize->0.35, GraphLayout->"CircularEmbedding",
EdgeStyle->Directive[Black, Thickness[0.0075]], VertexLabelStyle->Directive[Black, 17]]

system1 = {x1_{1,2} - x1_{5,1}==0,
-x1_{1,2} - x1_{5,2} - x1_{4,2}==0,
0==0,
x1_{4,2} - x1_{5,4}==0,
x1_{5,1} + x1_{5,4} + x1_{5,2}==0
};
t1 = TreePlot[{1->2, 4->2, 5->4}, Top, 1, VertexStyle->White, VertexSize->0.35,
VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Black, Thickness[0.0075]], VertexLabelStyle
->Directive[Black, 17]];
t2 = Graph[{3}, {}, VertexSize->0.035, VertexStyle->White, VertexLabels
->Placed["Name", Center], EdgeStyle->Directive[Black, Thickness[0.0075]],
VertexLabelStyle->Directive[Black, 17]];
GraphicsRow[{t1, t2}]
(*thread*)
t = {5, 4, 2, 1};
t1 = TreePlot[{1->2, 2->4, 4->5}, Top, 1, VertexStyle->White, VertexSize->0.35,
VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Black, Thickness[0.0075]],
VertexLabelStyle->Directive[Black, 17]];
t2 = Graph[{3}, {}, VertexSize->0.035, VertexStyle->White, VertexLabels->
Placed["Name", Center], EdgeStyle->Directive[Black, Thickness[0.0075]],
```

```
VertexLabelStyle->Directive[Black, 17]];
GraphicsRow[{t1, t2}]
(*pred*)
p = {0, 1, 0, 2, 4};
(*dir*)
d = {0, 1, 0, −1, −1};
system1a = system1;
system1a[[5]] = system1a[[5]]/. {x1_{5,1} → 1};
system1a[[1]] = system1a[[1]]/. {x1_{5,1} → 1};
system1a[[5]] = system1a[[5]]/. {x1_{5,2} → 0};
system1a[[2]] = system1a[[2]]/. {x1_{5,2} → 0};
δ1_{5,1} = {x1_{5,1} → 1, x1_{5,2} → 0};
k1 = 3;

For[i = 1, i<=k1, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system1a[[t[[i]]]], x1_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system1a[[t[[i]]]], x1_{t[[i]],p[[t[[i]]]]}] [[1]]];
If[p[[p[[t[[i]]]]]] ≠ 0,
system1a[[p[[t[[i]]]]]] = system1a[[p[[t[[i]]]]]/.δ];
δ1_{5,1} = Join [δ1_{5,1}, δ];
}];
Print [δ1_{5,1}];
Print [Simplify [system1/.δ1_{5,1}]];

Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 5->2},
VertexLabels->Placed["Name", Center], VertexStyle->White, VertexSize->0.35, GraphLayout->
"CircularEmbedding",
EdgeStyle->Directive[Black, Thickness[0.0075]], VertexLabelStyle->Directive[Black, 17]]
system1b = system1;
system1b[[5]] = system1b[[5]]/. {x1_{5,1} → 0};
system1b[[1]] = system1b[[1]]/. {x1_{5,1} → 0};
system1b[[5]] = system1b[[5]]/. {x1_{5,2} → 1};
```

```
system1b[[2]] = system1b[[2]]/. {x1_{5,2} → 1};
δ1_{5,2} = {x1_{5,1} → 0, x1_{5,2} → 1};

For[i = 1, i<=k1, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system1b[[t[[i]]]], x1_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system1b[[t[[i]]]], x1_{t[[i]],p[[t[[i]]]]}] [[1]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system1b[[p[[t[[i]]]]]] = system1b[[p[[t[[i]]]]]]/.δ];
δ1_{5,2} = Join [δ1_{5,2}, δ] ;
}];
Print [δ1_{5,2}] ;
Print [Simplify [system1/.δ1_{5,2}]] ;
Graph[{1, 2, 3, 4, 5},
{1->2, 2->3, 4->3, 5->4, 4->1},
VertexLabels->Placed["Name", Center], VertexStyle->White, VertexSize->0.35, GraphLayout->
"CircularEmbedding",
EdgeStyle->Directive[Dashed, Black, Thickness[0.0075]], VertexLabelStyle->Directive[Black, 17]]
system2 = {
x2_{1,2} − x2_{4,1}==0,
−x2_{1,2} + x2_{2,3}==0,
−x2_{2,3} − x2_{4,3}==0,
x2_{4,1} + x2_{4,3} − x2_{5,4}==0,
x2_{5,4}==0
};
TreePlot[{1->2, 2->3, 4->3, 5->4}, Top, 1, VertexSize->0.35, VertexStyle->White,
VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Dashed, Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]]
(*thread*)
t = {5, 4, 3, 2, 1};
TreePlot[{1->2, 2->3, 3->4, 4->5}, Top, 1, VertexSize->0.35, VertexStyle->White,
VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Dashed, Black, Thickness[0.03]],
```

```
VertexLabelStyle->Directive[Black, 17]]
(*pred*)
p = {0, 1, 2, 3, 4};
(*dir*)
d = {0, 1, 1, −1, −1};
system2a = system2;
system2a[[4]] = system2a[[4]]/. {x2_{4,1} → 1} ;
system2a[[1]] = system2a[[1]]/. {x2_{4,1} → 1} ;
δ2_{4,1} = {x2_{4,1} → 1} ;
k2 = 4;

For[i = 1, i<=k2, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system2a[[t[[i]]]], x2_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system2a[[t[[i]]]], x2_{t[[i]],p[[t[[i]]]]}] [[1]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system2a[[p[[t[[i]]]]]] = system2a[[p[[t[[i]]]]]/.δ];
δ2_{4,1} = Join [δ2_{4,1}, δ] ;
}];
Print [δ2_{4,1}] ;
Print [Simplify [system2/.δ2_{4,1}]]
Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 4->1, 4->3, 5->1},
VertexLabels->Placed["Name", Center], VertexStyle->White, VertexSize->0.35,
GraphLayout->"CircularEmbedding",
EdgeStyle->{Directive[DotDashed, Black, Thickness[0.008]]},
VertexLabelStyle->Directive[Black, 17]]
system3 = {x3_{1,2} − x3_{4,1} − x3_{5,1}==0,
−x3_{1,2} − x3_{4,2}==0,
−x3_{4,3}==0,
x3_{4,1} + x3_{4,3} + x3_{4,2} − x3_{5,4}==0,
x3_{5,1} + x3_{5,4}==0
};
TreePlot[{1->2, 4->2, 4->3, 5->4}, Top, 1, VertexSize->0.35, VertexStyle->White,
```

```
VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[DotDashed, Black, Thickness[0.02]],
VertexLabelStyle->Directive[Black, 17]]
(*thread*)
t = {5, 3, 4, 2, 1};
TreePlot[{1->2, 2->4, 4->3, 4->5}, Top, 1, VertexSize->0.35,
VertexStyle->White, VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[DotDashed, Black,
Thickness[0.02]], VertexLabelStyle->Directive[Black, 17]]
(*pred*)
p = {0, 1, 4, 2, 4};
(*dir*)
d = {0, 1, 1, -1, -1};
system3a = system3;
system3a[[4]] = system3a[[4]]/. {x3_{4,1} → 1} ;
system3a[[1]] = system3a[[1]]/. {x3_{4,1} → 1} ;
system3a[[5]] = system3a[[5]]/. {x3_{5,1} → 0} ;
system3a[[1]] = system3a[[1]]/. {x3_{5,1} → 0} ;
δ3_{4,1} = {x3_{4,1} → 1, x3_{5,1}->0} ;
k3 = 4;

For[i = 1, i<=k3, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system3a[[t[[i]]]], x3_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system3a[[t[[i]]]], x3_{t[[i]],p[[t[[i]]]]}] [[1]]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system3a[[p[[t[[i]]]]]] = system3a[[p[[t[[i]]]]]/.δ];
δ3_{4,1} = Join [δ3_{4,1}, δ] ;
}];
Print [δ3_{4,1}] ;
Print [Simplify [system3/.δ3_{4,1}]] ;
Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 5->1, 4->3},
VertexLabels->Placed["Name", Center], VertexStyle->White, VertexSize->0.35,
```

```
VertexSize->Medium, GraphLayout->"CircularEmbedding",
EdgeStyle->{Directive[DotDashed, Black, Thickness[0.008]]},
VertexLabelStyle->Directive[Black, 17], DirectedEdges->True]


system3b = system3;
system3b[[4]] = system3b[[4]]/. {x3_{4,1} → 0} ;
system3b[[1]] = system3b[[1]]/. {x3_{4,1} → 0} ;
system3b[[5]] = system3b[[5]]/. {x3_{5,1} → 1} ;
system3b[[1]] = system3b[[1]]/. {x3_{5,1} → 1} ;
δ3_{5,1} = {x3_{4,1} → 0, x3_{5,1}->1} ;


For[i = 1, i<=k3, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system3b[[t[[i]]]], x3_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system3b[[t[[i]]]], x3_{t[[i]],p[[t[[i]]]]}] [[1]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system3b[[p[[t[[i]]]]]] = system3b[[p[[t[[i]]]]]]/.δ];
δ3_{5,1} = Join [δ3_{5,1}, δ] ;
}];
Print [δ3_{5,1}] ;
Print [Simplify [system3/.δ3_{5,1}]] ;



(*particular soulution*)
Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4},
VertexLabels->Placed["Name", Center], VertexStyle->White,
VertexSize->0.35, GraphLayout->"CircularEmbedding",
EdgeStyle->{Directive[Black, Thickness[0.008]]}, VertexLabelStyle->Directive[Black, 17]]
system1 = {x1_{1,2} − x1_{5,1}==3,
−x1_{1,2} − x1_{5,2} − x1_{4,2}== − 15,
0==0,
x1_{4,2} − x1_{5,4}== − 8,
```

```
x1_{5,1} + x1_{5,4} + x1_{5,2}==20
};
t1 = TreePlot[{1->2, 4->2, 5->4}, Top, 1, VertexSize->0.35,
VertexStyle->White, VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]];
t2 = Graph[{3}, {}, VertexSize->0.035, VertexStyle->White,
VertexLabels->Placed["Name", Center], EdgeStyle->Directive[Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]];
GraphicsRow[{t1, t2}]
(*thread*)
t = {5, 4, 2, 1};


t1 = TreePlot[{1->2, 2->4, 4->5}, Top, 1, VertexSize->0.35,
VertexStyle->White, VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]];
t2 = Graph[{3}, {}, VertexSize->0.035, VertexStyle->White,
VertexLabels->Placed["Name", Center], EdgeStyle->Directive[Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]];
GraphicsRow[{t1, t2}]
(*pred*)
p = {0, 1, 0, 2, 4};
(*dir*)
d = {0, 1, 0, -1, -1};

system1a = system1;
system1a[[5]] = system1a[[5]]/. {x1_{5,1} → 0};
system1a[[1]] = system1a[[1]]/. {x1_{5,1} → 0};
system1a[[5]] = system1a[[5]]/. {x1_{5,2} → 0};
system1a[[2]] = system1a[[2]]/. {x1_{5,2} → 0};
δ1 = {x1_{5,1} → 0, x1_{5,2} → 0};

For[i = 1, i<=k1, ++i,
{
```

```
If[d[[t[[i]]]]==1,
δ = Solve [system1a[[t[[i]]]], x1_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system1a[[t[[i]]]], x1_{t[[i]],p[[t[[i]]]]}] [[1]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system1a[[p[[t[[i]]]]]] = system1a[[p[[t[[i]]]]]]/.δ];
δ1 = Join [δ1, δ] ;
}];
Print [δ1] ;
Print [Simplify [system1/.δ1]] ;
Graph[{1, 2, 3, 4, 5},
{1->2, 2->3, 4->3, 5->4},
VertexLabels->Placed["Name", Center], VertexStyle->White,
VertexSize->0.35, GraphLayout->"CircularEmbedding",
EdgeStyle->{Directive[Dashed, Black, Thickness[0.008]]},
VertexLabelStyle->Directive[Black, 17]]
system2 = {
x2_{1,2} − x2_{4,1}==0,
−x2_{1,2} + x2_{2,3}== − 6,
−x2_{2,3} − x2_{4,3}== − 5,
x2_{4,1} + x2_{4,3} − x2_{5,4}==7,
x2_{5,4}==4
};
TreePlot[{1->2, 2->3, 4->3, 5->4}, Top, 1, VertexSize->0.35,
VertexStyle->White, VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Dashed, Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]]
(*thread*)
t = {5, 4, 3, 2, 1};
TreePlot[{1->2, 2->3, 3->4, 4->5}, Top, 1, VertexSize->0.35,
VertexStyle->White, VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[Dashed, Black, Thickness[0.03]],
VertexLabelStyle->Directive[Black, 17]]
(*pred*)
p = {0, 1, 2, 3, 4};
(*dir*)
```

```
d = {0, 1, 1, −1, −1};
system2a = system2;
system2a[[4]] = system2a[[4]]/. {x2_{4,1} → 0} ;
system2a[[1]] = system2a[[1]]/. {x2_{4,1} → 0} ;
δ2 = {x2_{4,1} → 0} ;
For[i = 1, i<=k2, ++i,
{
If[d[[t[[i]]]]==1,
δ = Solve [system2a[[t[[i]]]], x2_{p[[t[[i]]]],t[[i]]}] [[1]],
δ = Solve [system2a[[t[[i]]]], x2_{t[[i]],p[[t[[i]]]]}] [[1]]] ;
If[p[[p[[t[[i]]]]]] ≠ 0,
system2a[[p[[t[[i]]]]]] = system2a[[p[[t[[i]]]]]]/.δ];
δ2 = Join [δ2, δ] ;
}];
Print [δ2] ;
Print [Simplify [system2/.δ2]] ;
Graph[{1, 2, 3, 4, 5},
{1->2, 4->2, 5->4, 4->3},
VertexLabels->Placed["Name", Center], VertexStyle->White,
VertexSize->0.35, GraphLayout->"CircularEmbedding",
EdgeStyle->{Directive[DotDashed, Black, Thickness[0.008]]},
VertexLabelStyle->Directive[Black, 17]]
system3 = {x3_{1,2} − x3_{4,1} − x3_{5,1}== − 10,
−x3_{1,2} − x3_{4,2}== − 10,
−x3_{4,3}== − 7,
x3_{4,1} + x3_{4,3} + x3_{4,2} − x3_{5,4}==9,
x3_{5,1} + x3_{5,4}==18
};
TreePlot[{1->2, 4->2, 4->3, 5->4}, Top, 1, VertexSize->0.35,
VertexStyle->White, VertexLabels->Placed["Name", Center],
DirectedEdges->True, EdgeStyle->Directive[DotDashed, Black, Thickness[0.02]],
VertexLabelStyle->Directive[Black, 17]]
(*thread*)
t = {5, 3, 4, 2, 1};
TreePlot[{1->2, 2->4, 4->3, 4->5}, Top, 1, VertexSize->0.35,
```

VertexStyle->White, VertexLabels->Placed["Name", Center],

DirectedEdges->True, EdgeStyle->Directive[DotDashed, Black, Thickness[0.02]],

VertexLabelStyle->Directive[Black, 17]]

(*pred*)

$p = \{0, 1, 4, 2, 4\};$

(*dir*)

$d = \{0, 1, 1, -1, -1\};$

system3a = system3;

system3a[[4]] = system3a[[4]]/. $\{x3_{4,1} \to 0\}$;

system3a[[1]] = system3a[[1]]/. $\{x3_{4,1} \to 0\}$;

system3a[[5]] = system3a[[5]]/. $\{x3_{5,1} \to 0\}$;

system3a[[1]] = system3a[[1]]/. $\{x3_{5,1} \to 0\}$;

$\overline{\delta 3} = \{x3_{4,1} \to 0, x3_{5,1}\text{->}0\}$;

For[$i = 1, i$<=k3, ++i,

{

If[d[[t[[i]]]]==1,

δ = Solve $\left[\text{system3a}[[t[[i]]]], x3_{p[[t[[i]]]],t[[i]]}\right]$ [[1]],

δ = Solve $\left[\text{system3a}[[t[[i]]]], x3_{t[[i]],p[[t[[i]]]]}\right]$ [[1]]] ;

If[p[[p[[t[[i]]]]]] $\neq 0$,

system3a[[p[[t[[i]]]]]] = system3a[[p[[t[[i]]]]]/.δ];

$\overline{\delta 3}$ = Join $\left[\overline{\delta 3}, \delta\right]$;

}];

Print $\left[\overline{\delta 3}\right]$;

Print $\left[\text{Simplify}\left[\text{system3}/.\overline{\delta 3}\right]\right]$;

$R11_{5,1} = 7x1_{1,2} + 9x1_{4,2} + 9x1_{5,1} + 9x1_{5,4}/.\delta 1_{5,1};$

$R21_{5,1} = 6x1_{1,2} + 5x1_{4,2} + 10x1_{5,1} + 9x1_{5,4}/.\delta 1_{5,1};$

$R31_{5,1} = 2x1_{1,2} + 3x1_{4,2} + 8x1_{5,1} + 2x1_{5,4}/.\delta 1_{5,1};$

$R41_{5,1} = 0x1_{1,2} + 0x1_{4,2} + 0x1_{5,1} + x1_{5,4}/.\delta 1_{5,1};$

$R11_{5,2} = 7x1_{1,2} + 9x1_{4,2} + 5x1_{5,2} + 9x1_{5,4}/.\delta 1_{5,2};$

$R21_{5,2} = 6x1_{1,2} + 5x1_{4,2} + 8x1_{5,2} + 9x1_{5,4}/.\delta 1_{5,2};$

$R31_{5,2} = 2x1_{1,2} + 3x1_{4,2} + 8x1_{5,2} + 2x1_{5,4}/.\delta 1_{5,2};$

$R41_{5,2} = 0x1_{1,2} + 0x1_{4,2} + 0x1_{5,2} + x1_{5,4}/.\delta 1_{5,2};$

$R12_{4,1} = 4x2_{1,2} + 8x2_{2,3} + 9x2_{4,1} + 2x2_{4,3} + 3x2_{5,4}/.\delta 2_{4,1};$

$$R22_{4,1} = 7x2_{1,2} + 3x2_{2,3} + 7x2_{4,1} + 0x2_{4,3} + 5x2_{5,4} /. \delta 2_{4,1};$$

$$R32_{4,1} = x2_{1,2} + 5x2_{2,3} + 2x2_{4,1} + 3x2_{4,3} + 4x2_{5,4} /. \delta 2_{4,1};$$

$$R42_{4,1} = 0x2_{1,2} + 0x2_{2,3} + 0x2_{4,1} + 0x2_{4,3} + 0x2_{5,4} /. \delta 2_{4,1};$$

$$R13_{4,1} = 2x3_{1,2} + 9x3_{4,1} + 4x3_{4,2} + 5x3_{4,3} + 7x3_{5,4} /. \delta 3_{4,1};$$

$$R23_{4,1} = 6x3_{1,2} + 0x3_{4,1} + 2x3_{4,2} + 2x3_{4,3} + 7x3_{5,4} /. \delta 3_{4,1};$$

$$R33_{4,1} = 5x3_{1,2} + 6x3_{4,1} + 0x3_{4,2} + 2x3_{4,3} + 8x3_{5,4} /. \delta 3_{4,1};$$

$$R43_{4,1} = 0x3_{1,2} + 0x3_{4,1} + 0x3_{4,2} + 0x3_{4,3} + x3_{5,4} /. \delta 3_{4,1};$$

$$R13_{5,1} = 2x3_{1,2} + 4x3_{4,2} + 5x3_{4,3} + 9x3_{5,1} + 7x3_{5,4} /. \delta 3_{5,1};$$

$$R23_{5,1} = 6x3_{1,2} + 2x3_{4,2} + 2x3_{4,3} + 7x3_{5,1} + 7x3_{5,4} /. \delta 3_{5,1};$$

$$R33_{5,1} = 5x3_{1,2} + 0x3_{4,2} + 2x3_{4,3} + x3_{5,1} + 8x3_{5,4} /. \delta 3_{5,1};$$

$$R43_{5,1} = 0x3_{1,2} + 0x3_{4,2} + 0x3_{4,3} + 0x3_{5,1} + x3_{5,4} /. \delta 3_{5,1};$$

$$DD = \begin{pmatrix} R11_{5,1} & R11_{5,2} & R12_{4,1} & R13_{4,1} \\ R21_{5,1} & R21_{5,2} & R22_{4,1} & R23_{4,1} \\ R31_{5,1} & R31_{5,2} & R32_{4,1} & R33_{4,1} \\ R41_{5,1} & R41_{5,2} & R42_{4,1} & R43_{4,1} \end{pmatrix};$$

$$A_1 = 556 - (7x1_{1,2} + 4x2_{1,2} + 2x3_{1,2} + 8x2_{2,3} + 9x2_{4,1} + 9x3_{4,1} + 9$$
$$x1_{4,2} + 4x3_{4,2} + 2x2_{4,3} + 5x3_{4,3} + 9x1_{5,1} + 9x3_{5,1} + 5x1_{5,2} + 9x1_{5,4} + 3x2_{5,4} +$$
$$7x3_{5,4} /. \overline{\delta 1} /. \overline{\delta 2} /. \overline{\delta 3};$$

$$A_2 = 501 - (6x1_{1,2} + 7x2_{1,2} + 6x3_{1,2} + 3x2_{2,3} + 7x2_{4,1} + 5x1_{4,2} +$$
$$2x3_{4,2} + 2x3_{4,3} + 10x1_{5,1} + 7x3_{5,1} + 8x1_{5,2} + 9x1_{5,4} + 5x2_{5,4} + 7x3_{5,4}) /. \overline{\delta 1} /. \overline{\delta 2} /. \overline{\delta 3};$$

$$A_3 = 307 - (2x1_{1,2} + x2_{1,2} + 5x3_{1,2} + 5x2_{2,3} + 2x2_{4,1} + 6x3_{4,1} + 3x1_{4,2} +$$
$$3x2_{4,3} + 2x3_{4,3} + 8x1_{5,1} + x3_{5,1} + 8x1_{5,2} + 2x1_{5,4} + 4x2_{5,4} + 8x3_{5,4}) /. \overline{\delta 1} /. \overline{\delta 2} /. \overline{\delta 3};$$

$$A_4 = 13 - (x1_{5,4} + x3_{5,4}) /. \overline{\delta 1} /. \overline{\delta 2} /. \overline{\delta 3};$$

$$\beta_1 = A_1 - (R13_{5,1} y3_{5,1});$$

$$\beta_2 = A_2 - (R23_{5,1} y3_{5,1});$$

$$\beta_3 = A_3 - (R33_{5,1} y3_{5,1});$$

$$\beta_4 = A_4 - (R43_{5,1} y3_{5,1});$$

$$Y = \text{Simplify}\left[\text{Inverse}[DD]. \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}\right];$$

Print[MatrixForm[DD]];

Print[Y];

```
rule1 = {x1_{5,1}->Y[[1]][[1]], x1_{5,2}->Y[[2]][[1]], x2_{4,1}->Y[[3]][[1]], x3_{4,1}->Y[[4]][[1]], x3_{5,1}->y3_{5,1}};
rule2 = {
x1_{1,2}->x1_{5,1} (x1_{1,2}/.δ1_{5,1}) + x1_{5,2} (x1_{1,2}/.δ1_{5,2}) + (x1_{1,2}/.δ1bar),
x1_{4,2}->x1_{5,1} (x1_{4,2}/.δ1_{5,1}) + x1_{5,2} (x1_{4,2}/.δ1_{5,2}) + (x1_{4,2}/.δ1bar),
x1_{5,4}->x1_{5,1} (x1_{5,4}/.δ1_{5,1}) + x1_{5,2} (x1_{5,4}/.δ1_{5,2}) + (x1_{5,4}/.δ1bar),
x2_{1,2}->x2_{4,1} (x2_{1,2}/.δ2_{4,1}) + (x2_{1,2}/.δ2bar),
x2_{2,3}->x2_{4,1} (x2_{2,3}/.δ2_{4,1}) + (x2_{2,3}/.δ2bar),
x2_{4,3}->x2_{4,1} (x2_{4,3}/.δ2_{4,1}) + (x2_{4,3}/.δ2bar),
x2_{5,4}->x2_{4,1} (x2_{5,4}/.δ2_{4,1}) + (x2_{5,4}/.δ2bar),

x3_{1,2}->x3_{4,1} (x3_{1,2}/.δ3_{4,1}) + x3_{5,1} (x3_{1,2}/.δ3_{5,1}) + (x3_{1,2}/.δ3bar),
x3_{4,2}->x3_{4,1} (x3_{4,2}/.δ3_{4,1}) + x3_{5,1} (x3_{4,2}/.δ3_{5,1}) + (x3_{4,2}/.δ3bar),
x3_{4,3}->x3_{4,1} (x3_{4,3}/.δ3_{4,1}) + x3_{5,1} (x3_{4,3}/.δ3_{5,1}) + (x3_{4,3}/.δ3bar),
x3_{5,4}->x3_{4,1} (x3_{5,4}/.δ3_{4,1}) + x3_{5,1} (x3_{5,4}/.δ3_{5,1}) + (x3_{5,4}/.δ3bar)};

solution = Simplify[Join[rule1, rule2/.rule1]];
Print[Simplify[System/.solution]];
Print[solution];
```

ПРИЛОЖЕНИЕ Б

Научная статья

ABOUT MULTIFLOW DECOMPOSITION METHODS IN FRACTIONAL LINEAR PROGRAMMING PROBLEMS

Pilipchuk L. A., Romanovsky Y. V., Hurinovich V. V.
Department of Computer Technology and Systems, Belarusian State University
Minsk, Belarus
E-mail: pilipchuk@bsu.by, yuriy_v_romanovskiy@mail.ru, gliger2004@gmail.com

We consider a nonlinear flow programming problem with a nested network constraint structure. An example of system decomposition, algorithms and technologies for solving large sparse linear systems with matrices of incomplete rank are given.

Introduction

Currently, there has been a significant increase in interest in the use of streaming programming. Network and streaming models are used in virtually all scientific, social and economic spheres of human activity. Network models are used in the analysis of a wide variety of systems, for example: inventory management systems, numerous territorial distribution systems (information, transport, energy). Mathematical models and problems of network flow programming can be formulated in terms of linear and fractional linear programming. Flow models are suitable for analyzing problems that have a network structure that can be conveniently described using certain parameters of arcs and nodes.

I. Mathematical model

For a multinetwork $G = (I, U)$, we consider the following linear-fractional optimization problem with linear constraints

$$f(x) = \frac{p(x)}{q(x)} = \frac{\sum\limits_{(i,j)\in U}\sum\limits_{k\in K(i,j)} p_{ij}^k x_{ij}^k + \beta}{\sum\limits_{(i,j)\in U}\sum\limits_{k\in K(i,j)} q_{ij}^k x_{ij}^k + \gamma} \longrightarrow \max, \tag{1}$$

$$\sum_{j\in I_i^+(U^k)} x_{ij}^k - \sum_{j\in I_i^-(U^k)} x_{ji}^k = a_i^k, i \in I^k, k \in K;$$

$$\sum_{k\in K_0(i,j)} x_{ij}^k \le d_{ij}^0, (i,j) \in U_0; \tag{2}$$

$$\sum_{(i,j)\in U}\sum_{k\in K(i,j)} \lambda_{ij}^{kp} x_{ij}^k = \alpha_p, p = \overline{1,l};$$

$$x_{ij}^k \ge 0, k \in K_0(i,j), (i,j) \in U_0; \tag{3}$$

$$0 \le x_{ij}^k \le d_{ij}^k, \in K_1(i,j), (i,j) \in U;\ x_{ij}^k \ge 0,$$

$$k \in K(i,j) \setminus K_1(i,j), (i,j) \in U \setminus U_0;$$

$$I_i^+(U^k) = \{j \in I^k : (i,j)^k \in U^k\};$$

$$I_i^-(U^k) = \{j \in I^k : (j,i)^k \in U^k\}. \tag{4}$$

Here $K\,(|K| < \infty)$ is a set of different products (types of flow) transported through the multinetwork G. Without loss of generality, let's put $K = \{1, \ldots, |K|\}$. Let us denote the connected network corresponding to a certain type k of flow with $S^k = (I^k, U^k)$, where I^k is the set of nodes and U^k is the set of arcs which are available for the flow of type k, $k \in K$. Also, we define for each node $i \in I$ the set of types of flows $K(i) = \{k \in K : i \in I^k\}$ and for each multiarc $(i,j) \in U$ the set $K(i,j) = \{k \in K : (i,j)^k \in U^k\}$. We assume that the denominator $q(x)$ of the objective function (1) does not change sign on a set of multiflows X, $x \in X$.

We use constructive decomposition theory [1] for constructing solutions of the following sparse linear systems: potencials system, system for appropriate direction of multinetwork change and to calculate the increment of the objective function. The work is devoted to methods, algorithms and technologies for constructing optimal and suboptimal solutions in synthesis with modern innovative technologies of sparse matrix analysis [2], algorithmic graph theory, theoretical computer science. The presented algorithms and computing technologies make it possible to construct solutions to large sparse linear systems with matrices of incomplete rank using parallel computing.

II. Example of linear system decomposition

For a multinetwork $S = (I, U)$, $I = \{1, 2, 3, 4, 5\}$, $U = \{(1,2), (2,3), (4,1), (4,2), (4,3), (5,1), (5,2), (5,4)\}$ consider a sparse underdetermined system of linear algebraic equations (5) – (6). Multinetwork S presented as a combination of networks S^k (Fig. 1): $S^k = (I^k, U^k)$, $k \in K = \{1, 2, 3\}$,

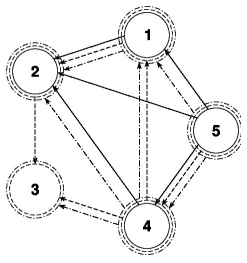

Fig. 1 – The multinetwork $S = (I, U)$

$I^1 = \{1,\ 2,\ 4,\ 5\}$, $I^2 = \{1,\ 2,\ 3,\ 4,\ 5\}$,

$$I^3 = \{1,\ 2,\ 3,\ 4,\ 5\},$$

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

$$x^1_{1,2} - x^1_{5,1} = 5, \quad -x^1_{1,2} - x^1_{4,2} - x^1_{5,2} = -13$$
$$x^1_{4,2} - x^1_{5,4} = -1, \quad x^1_{5,1} + x^1_{5,2} + x^1_{5,4} = 9$$

$$x^2_{1,2} - x^2_{4,1} = -1, \quad x^2_{2,3} - x^2_{1,2} = 7$$
$$-x^2_{2,3} - x^2_{4,3} = -13, \quad x^2_{4,1} + x^2_{4,3} - x^2_{5,4} = -1$$
$$x^2_{5,4} = 8$$

$$x^3_{1,2} - x^3_{4,1} - x^3_{5,1} = -4, \quad -x^3_{1,2} - x^3_{4,2} = -10$$
$$-x^3_{4,3} = -6, \quad -x^3_{4,1} + x^3_{4,2} + x^3_{4,3} - x^3_{5,4} = 12$$
$$x^3_{5,1} + x^3_{5,4} = 8 \tag{5}$$

$$x^1_{1,2} + 4x^2_{1,2} + 7x^3_{1,2} + 4x^2_{2,3} + 5x^2_{4,1} +$$
$$+4x^3_{4,1} + 3x^1_{4,2} + 6x^3_{4,2} + 6x^2_{4,3} + 2x^3_{4,3} +$$
$$+9x^1_{5,1} + 3x^3_{5,1} + 10x^1_{5,2} + 4x^1_{5,4} + 2x^2_{5,4} + 9x^3_{5,4} = 328$$

$$10x^1_{1,2} + 6x^2_{1,2} + 5x^3_{1,2} + 2x^2_{4,1} + 4x^3_{4,1} + 9x^1_{4,2} +$$
$$+10x^2_{4,3} + 4x^3_{4,3} + 4x^1_{5,1} + 2x^3_{5,1} + 4x^1_{5,2} + 7x^1_{5,4} +$$
$$+7x^2_{5,4} + 10x^3_{5,4} = 412$$

$$5x^1_{1,2} + 8x^2_{1,2} + x^3_{1,2} + 7x^2_{2,3} + 9x^2_{4,1} + 5x^3_{4,1} +$$
$$+2x^3_{4,2} + 6x^2_{4,3} + 5x^3_{4,3} + 4x^1_{5,1} + +x^3_{5,1} + 7x^1_{5,2} +$$
$$2x^1_{5,4} + 8x^2_{5,4} + 5x^3_{5,4} = 359 \tag{6}$$

Support $U^k_T \bigcup U^k_C, k \in K = \{1,2,3\}$ for the network $S = (I,U)$ for the system (5) – (6) [1] is represented on figures 2 – 4, where $U^1_T = \{(1,2)^1,\ (4,2)^1,\ (5,4)^1\}$, $U^2_T = \{(1,2)^2, (2,3)^2, (4,3)^2, (5,4)^2\}$, $U^3_T = \{(1,2)^3, (4,2)^3, (4,3)^3, (5,4)^3$ – sets of arcs of spanning trees U^1_T, U^2_T, U^3_T of the graphs $S^1 = (I^1,U^1)$, $S^2 = (I^2,U^2)$, $S^3 = (I^3,U^3)$ respectively (marked with bold lines), $U_C = U^1_C \bigcup U^2_C \bigcup U^3_C$ – set of cyclic arcs, $U^1_C = \{(5,1)^1, (5,2)^1\}$, $U^2_C = \{(4,1)^2\}$, $U^3_C = \emptyset$.

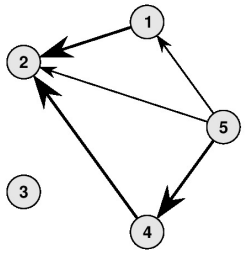

Fig. 2 – Support $U^1_T \bigcup U^1_C$

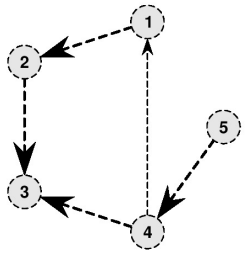

Fig. 3 – Support $U^2_T \bigcup U^2_C$

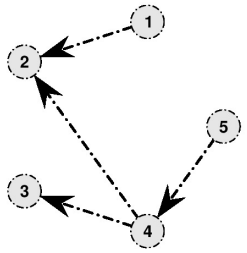

Fig. 4 – Support $U^3_T \bigcup U^3_C$

Construct a general solution to a sparse underdetermined system (5) – (6) relative to the support (Fig. 2 – 4) $U^k_T \bigcup U^k_C, k \in K = \{1,2,3\}$ of the network $S = (I,U)$ for the system (5) – (6).

General solution to sparse underdetermined system (5) – (6) relative to the reference set of arcs $U^k_T \bigcup U^k_C$, $k \in K = \{1,2,3\}$, which is shown in Fig. 2 – 4, has the form:

$$x^1_{5,2} \to \frac{1}{3,4}\left(-449 + 123y^3_{4,1} - 85y^3_{5,1}\right),$$
$$x^1_{5,1} \to \frac{1}{3,4}\left(3311 - 889y^3_{4,1} + 697y^3_{5,1}\right),$$
$$x^2_{4,1} \to -\frac{504}{17} + \frac{152y^3_{4,1}}{17} - 7y^3_{5,1},$$
$$x^3_{5,1} \to y^3_{5,1},\ x^3_{4,1} \to y^3_{4,1},$$
$$x^1_{1,2} \to \frac{1}{3,4}\left(3481 - 889y^3_{4,1} + 697y^3_{5,1}\right),$$
$$x^1_{4,2} \to \frac{1}{17}\left(-1295 + 383y^3_{4,1} - 306y^3_{5,1}\right),$$
$$x^1_{5,4} \to \frac{1}{1,7}\left(383y^3_{4,1} - 18\left(71 + 17y^3_{5,1}\right)\right)$$
$$x^2_{1,2} \to -\frac{521}{17} + \frac{152y^3_{4,1}}{17} - 7y^3_{5,1},$$
$$x^2_{2,3} \to -\frac{402}{17} + \frac{152y^3_{4,1}}{17} - 7y^3_{5,1},$$
$$x^2_{4,3} \to \frac{623}{17} - \frac{152y^3_{4,1}}{17} + 7y^3_{5,1}, x^2_{5,4} \to 8,$$
$$x^3_{1,2} \to -4 + y^3_{4,1} + y^3_{5,1},$$
$$x^3_{4,2} \to 14 - y^3_{4,1} - y^3_{5,1},$$
$$x^3_{4,3} \to 6,\ x^3_{5.4} \to 8 - y^3_{5,1}.$$

III. References

1. Pilipchuk L. A. Linear-fractional extremal inhomogeneous problems network flow programming. – Minsk : BSU, 2013. – 235 p. (in Russian).
2. Pilipchuk L.A. Sparse Linear Systems and Their Applications. – Minsk : BSU, 2013. – 235 p.