

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК 681.5

Отчет об исследовательском проекте на тему:
Построение оптимальных маршрутов передачи данных в сети

Выполнил студент:

группы #БПМИ229, 2 курса

Владимирова Екатерина Евгеньевна

Принял руководитель проекта:

Посыпкин Михаил Анатольевич

Научный сотрудник

Факультет компьютерных наук НИУ ВШЭ

Москва 2024

Содержание

Аннотация	3
1 Введение	4
1.1 Описание предметной области	4
1.2 Постановка задачи	4
1.3 Структура работы	4
2 Обзор литературы	5
3 Формальная постановка задачи	6
4 Решение проблемы подключения (CP - connectivity problem)	6
4.1 Полностью однородная матрица (completely uniform traffic matrix)	7
4.2 Однородная матрица трафика с нулевыми диагональными элементами (uniform traffic matrix with zero diagonal elements)	8
4.3 Квазиоднородная матрица трафика (a quasiuniform traffic matrix)	9
4.4 Матрица трафика кольцевого типа (ring type traffic matrix)	10
4.5 Матрица трафика центрального типа (centralized type traffic matrix)	11
4.6 Матрица трафика несвязанного типа (disconnected type traffic matrix)	12
5 Решение проблемы маршрутизации (RP - routing problem)	13
5.1 Формулировка проблемы	13
5.2 Собственный пример	14
5.3 Улучшение наивного алгоритма с помощью статьи [2]	18
5.4 Результаты для матриц	19
6 Итоговые результаты	20
Список литературы	21

Аннотация

Построение оптимальных маршрутов передачи данных в сети имеет огромное значение для локальных сетей и сетей больших городов. Цель проектирования заключается в том, чтобы свести к минимуму наибольший поток по любому каналу сети. Цель же этого курсового проекта - разобраться в теме оптимизации потоков и начать проектировать подобные пути для любой матрицы трафика и любого запроса (from, to), где from - отправитель сигнала, to - получатель сигнала. За основу эта работа будет брать статью: “Logically Rearrangeable Multihop Lightwave Networks” [1]. В ней авторы изучают задачу оптимизации многооперационных сетей с настраиваемыми передатчиками и приемниками.

Ключевые слова

Узлы, пропускная способность, трафик, приемники, передатчики, длина волны (bitrates), схема подключения, проблема подключения (CP), проблема маршрутизации (RP), линейное программирование.

1 Введение

1.1 Описание предметной области

В настоящее время коммуникационные сети имеют большое значение: через них ежедневно проходит огромное количество данных. При этом за последние десятилетия объем трафика передачи данных в сетях постоянно рос, поэтому для обеспечения высокого уровня обслуживания и сохранения скорости передачи данных были разработаны новые методы. Так, современные телекоммуникационные линии используют световые сигналы для передачи данных по направленным каналам, которые называют оптическими волокнами. С появлением этой технологии пропускная способность сетей увеличилась в несколько раз. Вместе с этим увеличилась потребность в передаче большего объема данных и при этом в короткие сроки. Сети разрастались и продолжают разрастаться, поэтому исследование построения оптимальных маршрутов передачи данных имеет большое значение.

1.2 Постановка задачи

В работе за основу берется сеть "multihop". Она состоит из физически распределенной оптической топологии и узлов, генерирующих и завершающих трафик, каждому из которых выделено небольшое количество передатчиков и приемников. Длина волны назначается передатчикам и приемникам, посредством чего создается несколько независимых каналов, каждый из которых определяет логическую связь между узлами.

На вход дается матрица трафика, которая описывает количество трафика, проходящее между всеми узлами сети, а также запрос $(from, to)$ - где *from* - отправитель, *to* - получатель.

Необходимо научиться строить граф исходя из матрицы трафика между узлами и научиться находить оптимальные потоки для определенного запроса $(from, to)$.

1.3 Структура работы

В разделе [2] представлен обзор литературы по изученной теме. Далее, в разделе [3] будет представлена более формальная постановка задачи. Решение отдельных подзадач, которые появляются в ходе решения основной задачи, будут описаны в разделах [4] и [5]. Конечные результаты исследования находятся в разделе [6].

2 Обзор литературы

Для решения поставленной задачи была использована статья [1]. Анализ этой статьи помог формализовать ее, а также найти метод решения.

В этой статье авторы используют сеть "multihop". Длина волны рассматривается как канал, который передает сигналы, исходящие от передатчиков. Канал доставляет сигнал от передатчиков к приемникам, которые имеют одинаковую длину волны. Канал может быть общим для многих соединений, ведь несколько передатчиков и приемников могут иметь одну и ту же длину волны. Однако в статье такой вариант не рассматривается. Каналы не являются общими, и каждая длины волны предназначена для уникального соединения.

Авторы предлагают алгоритм, демонстрирующий начальное решение, которое затем улучшается, с помощью локального поиска. Локальный поиск уточняет схему, при этом используются операции ветвления - обмена данными по наименее используемым звеньям схемы подключения. Именно здесь авторы разделяют проблему на две не зависимые: проблему подключения (CP) и проблему маршрутизации (RP). Сначала необходимо назначить длины волн соединениям, причем таким образом, чтобы соответствовать любой базовой структуре матрицы трафика. Соответственно, формулируется проблема подключения (CP).

После получения схемы подключения (т.е. графа) необходимо оптимально распределить потоки по ссылкам логической схемы. Это приводит к решению проблемы маршрутизации (RP). После чего выполняется локальный поиск в пространстве возможных решений, посредством применения операций обмена ветвями к недостаточно используемым ссылкам.

В отличие от поставленной в работе задачи, в статье [1] авторы хотят построить сеть сразу между всеми узлами, чтобы эффективно распространять информацию. Это задача является более сложной, по сравнению с той, которая решается в этой работе.

3 Формальная постановка задачи

Рассматривается матрица $N \times N$:

$$\Lambda = (\lambda_{st}) \quad (1)$$

, где λ_{st} – объем внешнего трафика в единицах потока в секунду, исходящий из узла s и заканчивающийся в узле t .

Также даны два узла $from$ и to , где $from$ - отправитель сигнала и to - получатель сигнала. Необходимо найти для матрицы Λ граф, на котором можно будет построить оптимальный поток для пары $(from, to)$. Далее вводятся несколько обозначений:

f_{ij}^{st} – трафик, который течет от узла i к узлу j , обусловленный парой (s, t) .

f_{ij} - общий поток по каналу (i, j) , который получается путем суммирования трафиков f_{ij}^{st} по всем парам (s, t) .

z_{ij} – количество направленных подключений от узла i к узлу j . T - количество приемников/передатчиков (неформально: количество ребер входящих в узел и исходящих из него).

После этого мы декомпозируем нашу задачу на две подзадачи, как и в статье: проблему подключения (CP) и проблему маршрутизации (RP), которые подробно будут описаны в разделах 4 и 5 соответственно.

4 Решение проблемы подключения (CP - connectivity problem)

В ходе решения этой задачи нам хотелось бы получить граф, в котором будут соединены узлы с наибольшим значением трафика, учитывая ограничения T .

Задача CP описывается следубщим образом:

$$maximize \sum_{i,j} \lambda_{ij} z_{ij} \quad (2)$$

При этом выполняется

$$\sum_i z_{ij} = T, \forall i \quad (3)$$

$$\sum_j z_{ij} = T, \forall j \quad (4)$$

$$z_{i,j} \text{ -- целое число в пределах } [0 - 1] \quad \forall i, j \quad (5)$$

Исходя из формулировки проблемы СР, можно понять, что ее можно решить с помощью линейного программирования, используя симплекс-метод.

Для работы используются стандартные матрицы с разной структурой.

4.1 Полностью однородная матрица (completely uniform traffic matrix)

Ее матрица трафика:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

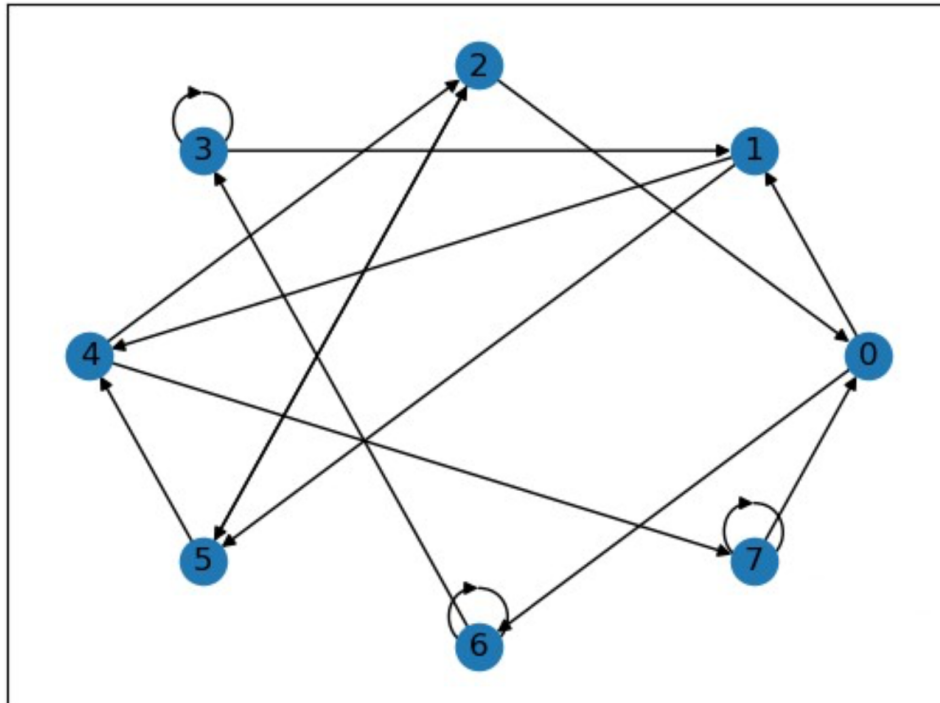


Рис. 4.1: Граф для полностью однородной матрицы, который был получен после решения задачи СР

На рисунке 4.1 представлен граф, который был получен после решения задачи СР.

Для этой матрицы получился такой граф, так как матрица полностью состоит из единиц, значит необходимо выполнить только ограничение на количество приемников и передатчиков. Поэтому матрица не имеет каких-либо характерных признаков, кроме того, что из-за единиц на диагонали возможен случай, когда ребро из вершины ведет в нее же.

4.2 Однородная матрица трафика с нулевыми диагональными элементами (uniform traffic matrix with zero diagonal elements)

Ее матрица трафика:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

На рисунке 4.2 представлен граф, который был получен после решения задачи СР для однородной матрицы с нулями на диагонали. В этом случае матрица так же не имеет каких-либо отличительных признаков. Главное чтобы соблюдались ограничения на количество приемников и передатчиков. Также в этом случае точно не возникнет ребер, которые идут из вершины в ту же вершину.

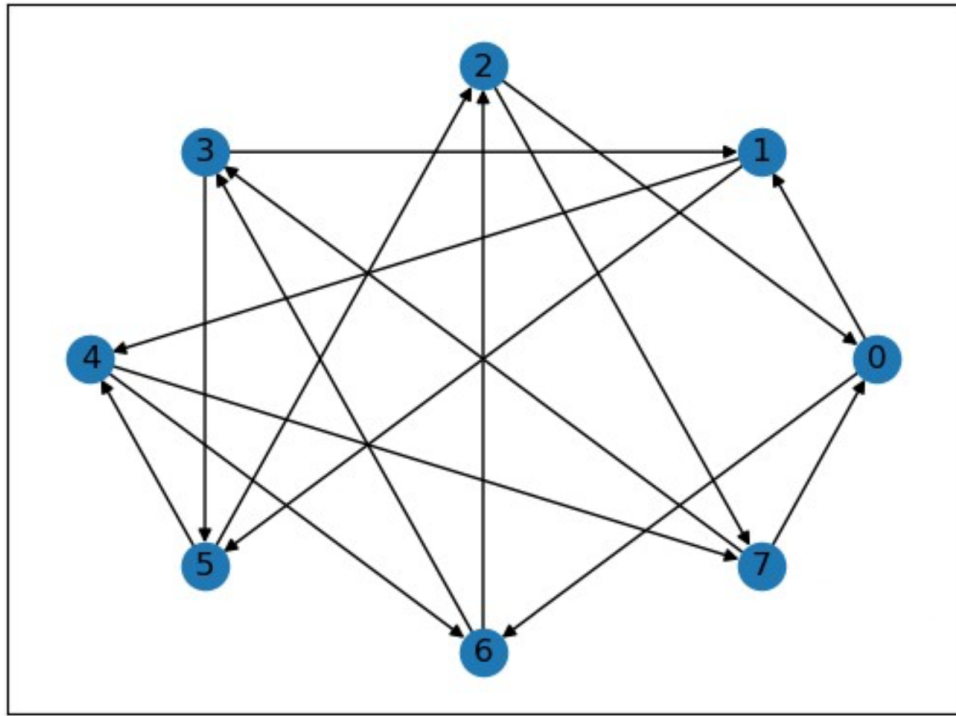


Рис. 4.2: Граф для однородной матрицы с нулями на диагонали, который был получен после решения задачи СР

4.3 Квазиоднородная матрица трафика (a quasiuniform traffic matrix)

Ее матрица трафика:

$$A = \begin{pmatrix} 0.0 & 1.1 & 1.0 & 0.9 & 0.9 & 1.0 & 0.8 & 0.7 \\ 0.8 & 0.0 & 0.7 & 1.0 & 1.1 & 1.0 & 0.9 & 0.9 \\ 0.9 & 1.0 & 0.0 & 1.1 & 1.2 & 0.8 & 0.9 & 0.8 \\ 1.1 & 1.1 & 1.0 & 0.0 & 0.7 & 0.8 & 0.9 & 0.9 \\ 0.9 & 1.0 & 1.1 & 0.7 & 0.0 & 0.8 & 0.8 & 1.0 \\ 1.0 & 0.8 & 0.8 & 0.9 & 1.0 & 0.0 & 1.0 & 1.0 \\ 1.1 & 1.0 & 0.9 & 0.9 & 0.8 & 0.7 & 0.0 & 0.9 \\ 0.8 & 0.8 & 1.0 & 1.0 & 1.1 & 0.9 & 0.9 & 0.0 \end{pmatrix}$$

На рисунке 4.3 представлен граф, который был получен после решения задачи СР для квазиоднородной матрицы. В итоге мы получили граф, в котором ребра из вершины идут в те вершины, которые имеют наибольшее значение в матрице трафика. Конечно эти значения не для всех вершин максимальны, однако выбираются максимально возможные, чтобы учитывались ограничения на количество приемников и передатчиков.

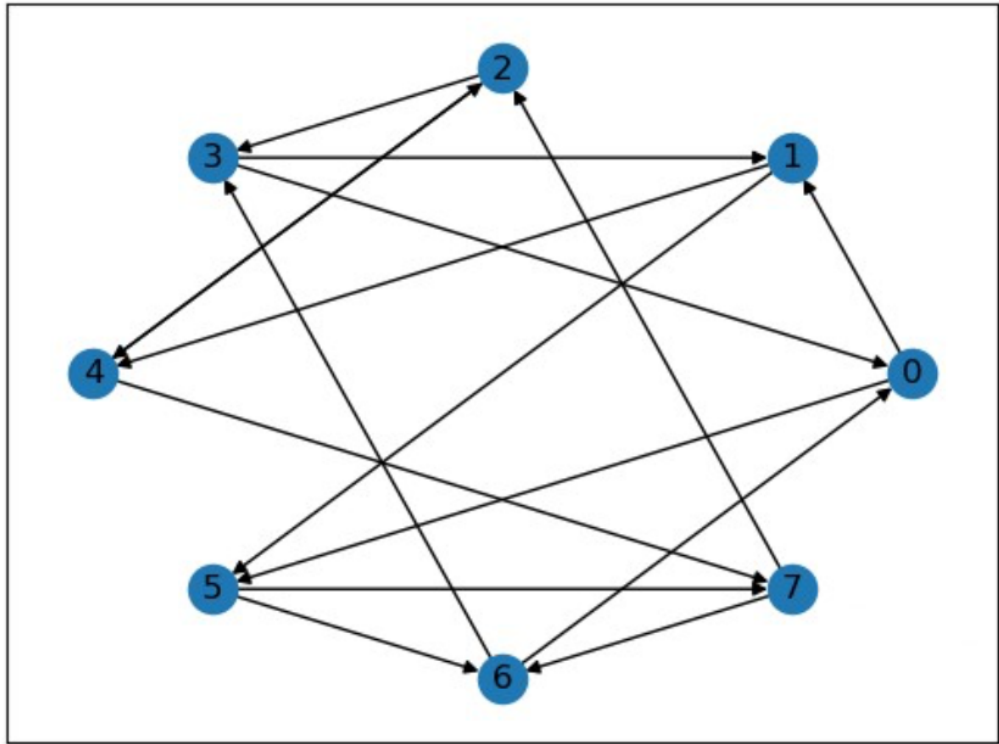


Рис. 4.3: Граф для квазиоднородной матрицы, который был получен после решения задачи СР

4.4 Матрица трафика кольцевого типа (ring type traffic matrix)

Ее матрица трафика:

$$A = \begin{pmatrix} 0.0 & 10.0 & 0.9 & 1.0 & 0.8 & 1.0 & 1.1 & 0.8 \\ 0.9 & 0.0 & 11.0 & 0.9 & 1.0 & 1.0 & 0.8 & 1.0 \\ 1.1 & 0.9 & 0.0 & 8.0 & 0.8 & 0.9 & 1.0 & 0.9 \\ 1.1 & 1.0 & 1.0 & 0.0 & 9.0 & 0.9 & 0.9 & 1.0 \\ 0.8 & 1.0 & 0.8 & 0.9 & 0.0 & 10.0 & 1.1 & 0.9 \\ 0.9 & 0.9 & 1.0 & 1.0 & 0.9 & 0.0 & 12.0 & 1.1 \\ 0.8 & 1.1 & 1.1 & 1.0 & 0.8 & 0.9 & 0.0 & 9.0 \\ 11.0 & 0.9 & 0.8 & 0.7 & 1.0 & 1.1 & 0.8 & 0.0 \end{pmatrix}$$

На рисунке 4.4 представлен граф, который был получен после решения задачи СР для матрицы кольцевого типа. Как мы и предполагали, граф имеет кольцевую структуру, а остальные ребра идут в вершины со значениями, которые имеют максимально возможные значения трафика, при данных ограничениях.

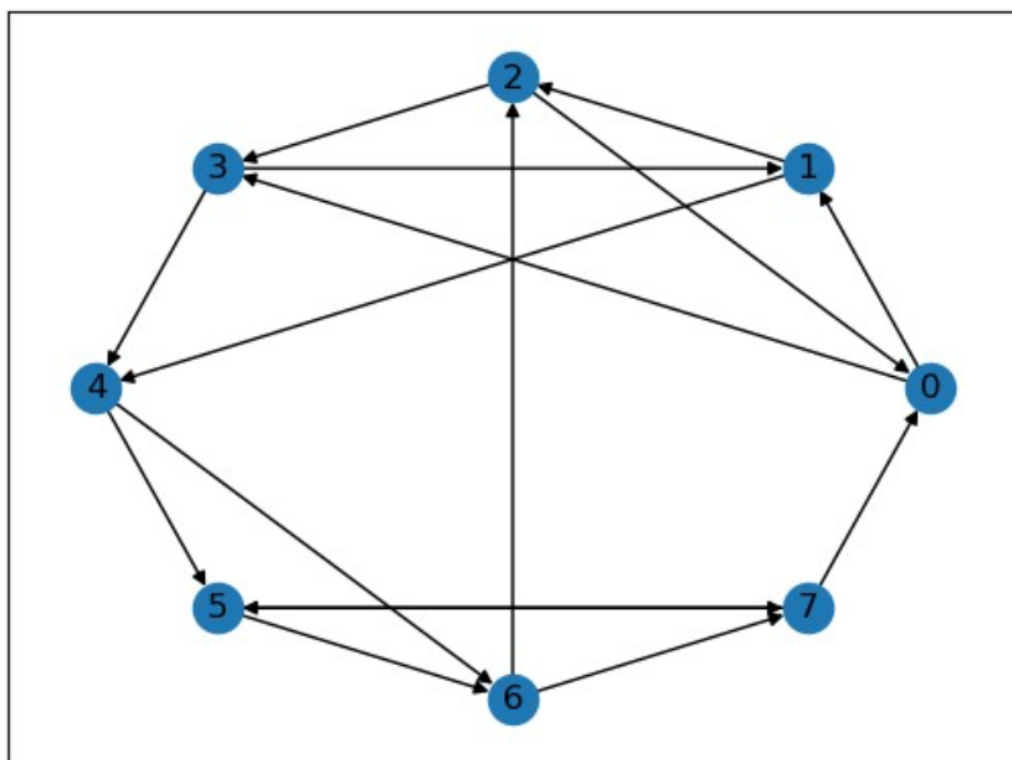


Рис. 4.4: Граф для матрицы кольцевого типа, который был получен после решения задачи СР

4.5 Матрица трафика центрального типа (centralized type traffic matrix)

Ее матрица трафика:

$$A = \begin{pmatrix} 0.0 & 10.0 & 8.0 & 7.0 & 11.0 & 9.0 & 11.0 & 10.0 \\ 12.0 & 0.0 & 0.8 & 1.0 & 0.9 & 1.0 & 1.0 & 0.9 \\ 8.0 & 1.1 & 0.0 & 0.7 & 0.9 & 1.0 & 1.1 & 1.0 \\ 9.0 & 1.0 & 1.1 & 0.0 & 0.8 & 0.8 & 0.7 & 0.9 \\ 10.0 & 0.9 & 1.0 & 0.9 & 0.0 & 0.9 & 1.0 & 1.1 \\ 9.0 & 1.1 & 0.8 & 1.1 & 1.1 & 0.0 & 1.0 & 0.9 \\ 11.0 & 0.8 & 0.9 & 0.9 & 1.0 & 1.0 & 0.0 & 0.7 \\ 8.0 & 0.7 & 1.0 & 1.1 & 0.8 & 0.8 & 0.9 & 0.0 \end{pmatrix}$$

На рисунке 4.5 представлен граф, который был получен после решения задачи СР для матрицы трафика центрального типа. В этом графе от нулевой вершины и в нулевую вершину ребра идут по максимальным значениям трафика. В остальных вершинах (подматрице из вершин 1 - 7) ребра также идут по максимально возможным значениям трафика, ровно как в квазиоднородной матрице.

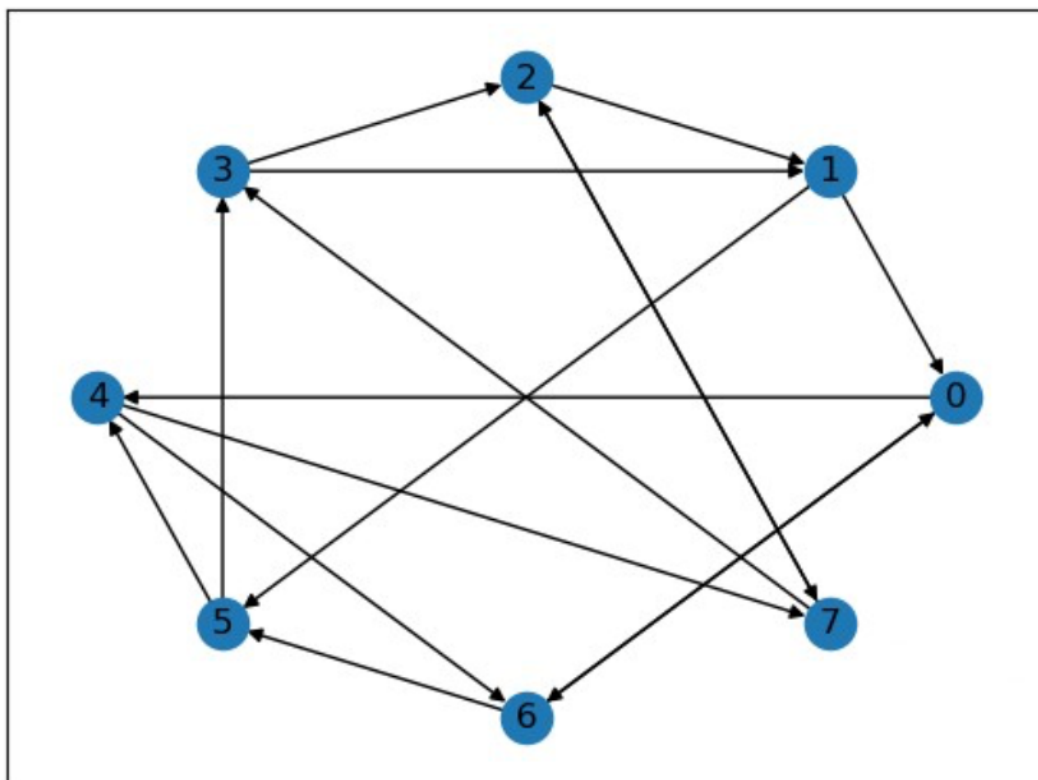


Рис. 4.5: Граф для матрицы кольцевого типа, который был получен после решения задачи СР

4.6 Матрица трафика несвязанного типа (disconnected type traffic matrix)

Ее матрица трафика:

$$A = \begin{pmatrix} 0.0 & 10.0 & 11.0 & 9.0 & 0.9 & 0.8 & 1.0 & 1.0 \\ 9.0 & 0.0 & 11.0 & 9.0 & 1.0 & 0.8 & 1.0 & 0.9 \\ 10.0 & 12.0 & 0.0 & 8.0 & 0.9 & 0.9 & 1.1 & 1.0 \\ 8.0 & 9.0 & 10.0 & 0.0 & 1.1 & 1.0 & 0.8 & 0.7 \\ 0.7 & 0.8 & 1.1 & 1.0 & 0.0 & 10.0 & 11.0 & 8.0 \\ 1.2 & 0.8 & 0.9 & 0.9 & 9.0 & 0.0 & 9.0 & 8.0 \\ 0.8 & 1.1 & 1.0 & 1.1 & 10.0 & 11.0 & 0.0 & 11.0 \\ 0.9 & 1.1 & 1.0 & 1.0 & 11.0 & 8.0 & 9.0 & 0.0 \end{pmatrix}$$

На рисунке 4.6 представлен граф, который был получен после решения задачи СР для матрицы трафика несвязанного типа. В итоге граф получился такой, как и ожидалось - он имеет две компоненты связности, где ребра идут по максимально возможным значениям трафика.

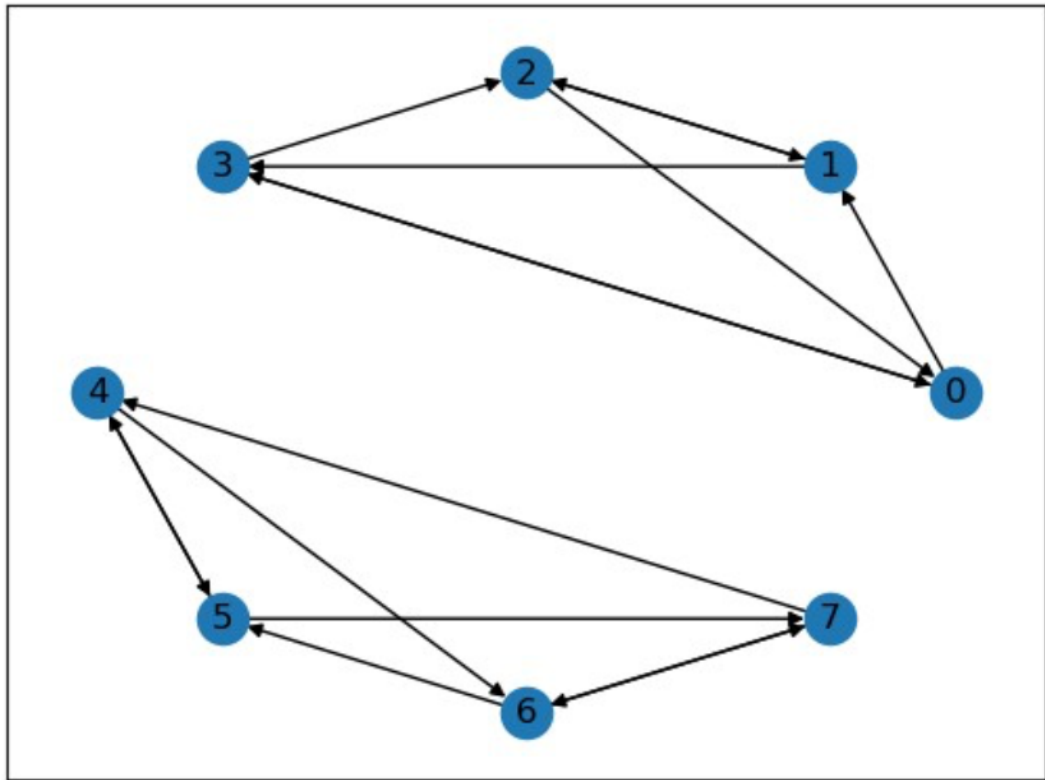


Рис. 4.6: Граф для матрицы несвязанного типа, который был получен после решения задачи СР

5 Решение проблемы маршрутизации (RP - routing problem)

5.1 Формулировка проблемы

После получения схемы подключения необходимо оптимально распределить потоки по ссылкам логической схемы. Это приводит к решению проблемы маршрутизации (RP). Она формулируется следующим образом:

$$\text{minimize } \max_{(i,j)} \left\{ \frac{f_{ij}}{C_{ij}} \right\} \quad (6)$$

И должны выполняться следующие ограничения:

$$\sum_j f_{ij}^{st} - \sum_j f_{ji}^{st} = \begin{cases} \lambda_{st}, & \text{if } s = i \\ \lambda_{st}, & \text{if } t = i, \\ 0, & \text{иначе} \end{cases} \quad \forall i, s, t \quad (7)$$

$$f_{ij} = \sum_{s,t} f_{ij}^{st}, \quad \forall i, j \quad (8)$$

$$f_{ij}^{st}, f_{ij} \geq 0 \quad \forall i, j, s, t \quad (9)$$

При этом, если $C_{i,j} = 0$, то дробь $\frac{f_{ij}}{C_{ij}}$ принимается равной нулю. Стоит отметить, что в формулировке проблемы нет ограничений по емкости, поэтому, как только решение получено, необходимо проверить, что общий поток по каналу f_{ij} не превышает пропускную способность канала - C .

5.2 Собственный пример

Для того чтобы разобраться в проблеме RP, был рассмотрен пример для графа на четырех вершинах. Матрица траффика имеет вид:

$$\Lambda = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Для этой матрицы мы решаем задачу СР, после чего получаем матрицу:

$$Z = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Значит можем построить граф:

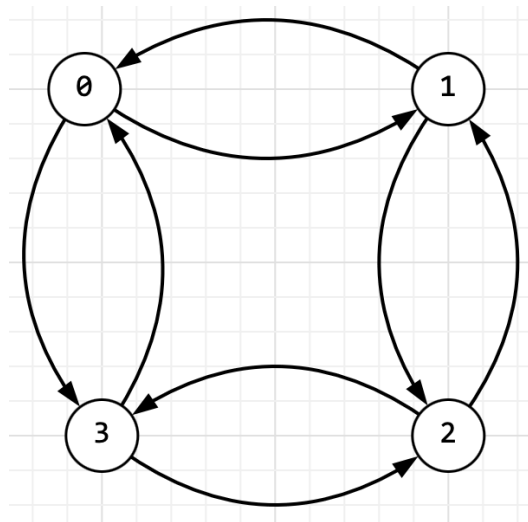


Рис. 5.1: Граф для примера на 4 вершины

На рисунке 5.1 представлен граф для составленного примера.

Теперь мы хотим решить проблему RP. Для этого посмотрим на условия (7) - (9). Так

как мы пока что опускаем значение C , то $C_{i,j}$ - равны 1, если $z_{i,i} \neq 0$.

В нашем примере: $C_{0,2} = C_{0,3} = C_{1,2} = C_{1,3} = C_{2,0} = C_{2,1} = C_{3,0} = C_{3,1} = 1$

Теперь нам хотелось бы насчитать $f_{i,j}$. Посчитать их можно с помощью формулы (8) из статьи. Однако они не постоянные. Значит сначала допустим, что волна от вершины i к вершине j может прийти через всевозможные пути. Значит она может пройти через все возможные каналы. Тогда можем посчитать $f_{i,j}$.

Так как $C_{i,j}$ и $z_{i,j}$ равны либо 0, либо 1, то нам достаточно посчитать наличие пути, который проходит через канал. Обозначать это будет как $0 \rightarrow 3$ (волна от 0 до 3 проходит через данный канал)

Теперь посчитаем:

$$f_{0,2} = (0 \rightarrow 2) + (0 \rightarrow 1) + (0 \rightarrow 3) + (1 \rightarrow 2) + (3 \rightarrow 2) + (3 \rightarrow 1) = 6$$

$$f_{0,3} = (0 \rightarrow 3) + (0 \rightarrow 1) + (0 \rightarrow 2) + (1 \rightarrow 3) + (2 \rightarrow 3) + (2 \rightarrow 1) = 6$$

$$f_{1,2} = (1 \rightarrow 2) + (1 \rightarrow 3) + (1 \rightarrow 0) + (3 \rightarrow 0) + (3 \rightarrow 2) + (0 \rightarrow 2) = 6$$

$$f_{1,3} = (1 \rightarrow 3) + (1 \rightarrow 0) + (1 \rightarrow 2) + (2 \rightarrow 3) + (0 \rightarrow 3) + (2 \rightarrow 0) = 6$$

$$f_{2,0} = (2 \rightarrow 0) + (2 \rightarrow 1) + (2 \rightarrow 3) + (1 \rightarrow 0) + (3 \rightarrow 0) + (1 \rightarrow 3) = 6$$

$$f_{2,1} = (2 \rightarrow 1) + (2 \rightarrow 3) + (2 \rightarrow 0) + (0 \rightarrow 1) + (3 \rightarrow 1) + (0 \rightarrow 3) = 6$$

$$f_{3,0} = (3 \rightarrow 0) + (3 \rightarrow 2) + (3 \rightarrow 1) + (1 \rightarrow 0) + (2 \rightarrow 0) + (1 \rightarrow 2) = 6$$

$$f_{3,1} = (3 \rightarrow 1) + (3 \rightarrow 2) + (3 \rightarrow 0) + (0 \rightarrow 1) + (2 \rightarrow 1) + (0 \rightarrow 2) = 6$$

И посмотрим, выполняются условие (7) из статьи:

Взяв в качестве s, t вершины 1 и 2 соответственно, получим, что условие не выполняется:

$$st = 12$$

$$i = 1$$

$$\sum_j f_{1,j}^{12} - \sum_j f_{j,1}^{12} = f_{1,2} + f_{1,3} - 0 = 2$$

Получим, что

$$2 \neq 1 = \lambda_{1,2}$$

Когда этот процесс выполнялся в ручную, то было замечено, что условие (7) будет выполняться, если между двумя вершинами будет существовать только один путь. Тогда для начальной вершины будет только один канал, и λ_{ij} будет входить в него с положительным значением. У последней вершины также будет один канал, из которого она будет выходить, а значит будет иметь отрицательное значение. Все остальные вершины в пути будут иметь два канала, а так как мы суммируем по всем j , фиксируя i , то значение превратится в 0.

Посмотрим на пример.

Для начала, построим для каждого s, t единственные пути:

Вершины: Путь

$$0 \rightarrow 1 : \quad 0 \rightarrow 2 \rightarrow 1$$

$$0 \rightarrow 2 : \quad 0 \rightarrow 2$$

$$0 \rightarrow 3 : \quad 0 \rightarrow 3$$

$$1 \rightarrow 0 : \quad 1 \rightarrow 3 \rightarrow 0$$

$$1 \rightarrow 2 : \quad 1 \rightarrow 2$$

$$1 \rightarrow 3 : \quad 1 \rightarrow 3$$

$$2 \rightarrow 0 : \quad 2 \rightarrow 0$$

$$2 \rightarrow 1 : \quad 2 \rightarrow 1$$

$$2 \rightarrow 3 : \quad 2 \rightarrow 1 \rightarrow 3$$

$$3 \rightarrow 0 : \quad 3 \rightarrow 0$$

$$3 \rightarrow 1 : \quad 3 \rightarrow 1$$

$$3 \rightarrow 2 : \quad 3 \rightarrow 1 \rightarrow 2$$

Можем обозначить $f_{i,j}$.

Так как очень много $f_{i,j}^{st}$ имеют значение 0, нам проще обозначить те, которые равны 1.

$$f_{0,2}^{01} = 1, \quad f_{2,1}^{01} = 1, \quad f_{0,2}^{02} = 1, \quad f_{0,3}^{03} = 1, \quad f_{1,3}^{10} = 1, \quad f_{3,0}^{10} = 1,$$

$$f_{1,2}^{12} = 1, \quad f_{1,3}^{13} = 1, \quad f_{2,0}^{20} = 1, \quad f_{2,1}^{21} = 1, \quad f_{2,1}^{23} = 1, \quad f_{1,3}^{23} = 1,$$

$$f_{3,0}^{30} = 1, \quad f_{3,0}^{30} = 1, \quad f_{3,1}^{32} = 1, \quad f_{1,2}^{32} = 1$$

$$\forall i, j, s, t \neq \quad f_{i,j}^{s,t} = 0$$

Теперь построим $f_{i,j}$ по этому правилу и проверим условие (7)

$$f_{0,2} = (0 \rightarrow 2) + (0 \rightarrow 1) = 2$$

$$f_{0,3} = (0 \rightarrow 3) + (2 \rightarrow 3) = 2$$

$$f_{1,2} = (1 \rightarrow 2) + (3 \rightarrow 2) = 2$$

$$f_{1,3} = (1 \rightarrow 3) + (1 \rightarrow 0) = 2$$

$$f_{2,0} = (2 \rightarrow 0) + (2 \rightarrow 3) = 2$$

$$f_{2,1} = (2 \rightarrow 1) + (0 \rightarrow 1) = 2$$

$$f_{3,0} = (3 \rightarrow 0) + (1 \rightarrow 0) = 2$$

$$f_{3,1} = (3 \rightarrow 1) + (3 \rightarrow 2) = 2$$

Теперь проверим условие (7).

$$st = 12$$

$$i = 1$$

$$\sum_j f_{1,j}^{12} - \sum_j f_{j,1}^{12} = f_{1,2}^{12} - 0 = 1 = \lambda_{1,2}$$

$$i = 2$$

$$\sum_j f_{2,j}^{12} - \sum_j f_{j,2}^{12} = 0 - f_{1,2}^{12} = -1 = -\lambda_{1,2}$$

$\forall i \neq 1, i \neq 2$ получим то, что я описывала выше, либо не будет таких значений $f_{i,j}^{12}$, либо они она будут равны $\lambda_{1,2}$, поэтому разность сумм будет равна 0. Для всех остальных s и t значения будут такими же.

Однако, можно понять, что выстраивать единственный путь между отправителем и получателем неоптимально. Намного лучше разделить потоки.

Это демонстрирует следующий пример:

Рисунок 5.2 демонстрирует, что если между вершинами необходимо провести трафик, который равен 1, то его лучше разделить между двумя путями. И тогда максимальное

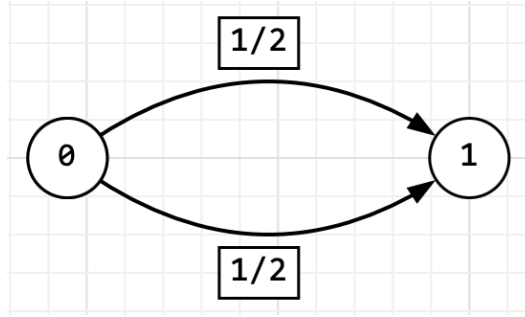


Рис. 5.2: Пример , который демонстрирует ошибочность начальных суждений

значение ребра стало меньше чем могло быть.

Это означает, что первое предположение оказалось неверным, поэтому необходимо найти другой способ оптимизировать значения ребер так, чтобы максимальное значение было минимальным.

5.3 Улучшение наивного алгоритма с помощью статьи [2]

Чтобы решить эту проблему, авторы из статьи [1] используют статью [2]. В ней в основном описываются две задачи, которые авторы решают. Задачи заключаются в том, чтобы оптимально распределить потоки на графе.

Поток представляется как вектор, состоящий из всех ребер графа, где элемент вектора описывает значение потока на ребре.

Тогда

$$f^{from,to} = \sum_k f_k^{from,to} \quad (10)$$

где from - отправитель, to - получатель. И при этом $f^{from,to} = \lambda_{from,to}$.

Теперь необходимо найти оптимальный такой поток.

В статье [2] приводится алгоритм, который дает нам решение. Этот алгоритм использует линейный оператор $FD(v, \lambda)$, который применяется к уже найденному потоку f .

Линейный оператор описывается следующим образом:

$$FD(v, \lambda) = (1 - \lambda)f + \lambda v \quad (11)$$

где v — другой поток, а λ — значение в пределах $0 \leq \lambda \leq 1$.

Почему же это может улучшить поток? Рассмотрим поток f , который был найден до

этого. А также найдем другой поток v . При этом

$$f^{from,to} = \sum_k f_k^{from,to} = \lambda_{from,to} \quad (12)$$

$$v^{from,to} = \sum_k v_k^{from,to} = \lambda_{from,to} \quad (13)$$

Тогда

$$\begin{aligned} f' &= FD(v, \lambda) = (1 - \lambda)f + \lambda v = f - \lambda f + \lambda v = \\ &= \sum_k f_k^{from,to} - \lambda \sum_k f_k^{from,to} + \lambda \sum_k v_k^{from,to} = \lambda_{from,to} - \lambda \lambda_{from,to} + \lambda \lambda_{from,to} = \lambda_{from,to} \end{aligned}$$

Значит, чтобы рассеять поток, можно применить следующий алгоритм:

- 1) Найти какой-то путь между вершинами from и to
- 2) Случайным образом присвоить значения на ребра в пути в этом потоке
- 3) Найти второй путь между вершинами from и to
- 4) Случайным образом присвоить значения на ребра в пути в этом потоке
- 5) Применить оператор FD

После применения оператора к двум потокам, будет получен новый поток, у которого значения на ребрах будут меньше, чем в двух предыдущих. При этом поток останется равным значению $\lambda_{from,to}$.

5.4 Результаты для матриц

Для каждой матрицы, описанной в разделе [4] была решена задача RP, где в качестве from, to рассматривались все пары вершин i, j , таких что $i \neq j$. Для каждого запроса находились два потока, которые удовлетворяют условиям (7) - (9). Чтобы посмотреть на результат, находилось среднее значение максимального ребра, а также минимальное и максимальное значение для каждой матрицы трафика.

Результаты приведены в таблице 5.1.

Таблица 5.1: Среднее, максимальное и минимальное значения для максимального ребра все всех запросов матрицы

	F — average	F — min	F — max
a uniform matrix	0.35824	0.233712	0.5
a uniform matrix with zero	0.350228	0.218954	0.470943
a quasiuniform matrix	0.35821	0.181447	0.552
a ring type matrix	0.388079	0.176127	6
a centralized type matrix	0.346114	0.154798	5.21586
a disconnected type matrix	0.389942	0.157224	5.64

6 Итоговые результаты

Цели исследования достигнуты. В течение работы над проектом удалось погрузиться в тему построения оптимальных маршрутов и оптимизации потоков между узлами. Однако для полного понимания темы необходимо больше времени.

Было решено две подзадачи. На вход каждой подзадачи подавалось 6 матриц, которые более подробно были описаны в разделе [4]. Каждая матрица имеет 8 вершин, а также ограничение на T : количество приемников и передатчиков, которое для каждой матрицы равно 2.

В ходе решения проблемы подключения (CP) были найдены графы для шести стандартных матриц, с помощью применения симплекс-метода. Решение, код и графы представлены в Jupyter Notebook. Число строк кода: 1419.

В ходе решения проблемы маршрутизации (RP) был написан код на C++, который находит два пути и определяет два возможных потока между узлами запроса, после чего находит новый поток, который является рассеиванием предыдущих двух, при этом уменьшая значение максимального ребра. Число строк кода: 304.

Перспективы дальнейшей деятельности вполне очевидны. В данной работе проблема RP была решена только для отдельного запроса. Далее можно попытаться построить оптимальный маршрут и найти оптимальные потоки для двух запросов, а потом и для матрицы запросов. Для этого необходимо будет научиться решать проблему RP для матрицы запросов, т.е. понять, как применять линейный оператор FD к нескольким потокам. После того как эта задача решена, можно будет пытаться искать новый метод оптимизации маршрутов данных в сети.

Список литературы

- [1] J.-F.P. Labourdette и A.S. Acampora. “Logically rearrangeable multihop lightwave networks”. В: *IEEE Transactions on Communications* 39.8 (1991), с. 1223—1230.
- [2] L. Fratta, M. Gerla и L. Kleinrock. “The flow deviation method: An approach to store-and-forward computer-communication network design”. В: *Networks* 3.2 (1973), с. 97—133.