

Содержание

1	Введение	3
1.1	Актуальность работы	3
1.2	Цель магистерской работы	4
1.3	Цель курсовой работы	5
2	Определения и обозначения	5
3	Потоки в графах	6
3.1	Определение	6
3.2	Стандартные алгоритмы	7
4	Методы ускорения расчёта	7
4.1	Разделение графа на несколько компонент связности	8
4.2	Стягивание цепи	8
4.3	Обработка мостов	10
4.4	Удаление висячих вершин	11
5	Реализация алгоритмов и численные эксперименты	12
5.1	Алгоритм перебора пар	12
5.2	Алгоритм с использованием редукций	12
6	Реализация алгоритмов и численные эксперименты	13
7	Заключение	14

1 Введение

В данной работе рассматривается такой показатель надёжности сети, как матрица математических ожиданий величин максимальных потоков между всеми парами узлов в сети. Точный расчёт этого показателя требует полного перебора всех возможных вариантов разрушений сети, а следовательно является NP-трудной задачей и, как следствие, сопряжён с большими временными и вычислительными трудностями и требует разработки новых алгоритмов.

При рассмотрении задач, связанных с надёжностью сетей, сеть обычно описывается случайным графом, где рёбра отображают сетевые каналы, а в качестве узлов выступают сенсоры, серверы, переключатели и другие устройства. В нашем случае, в качестве узлов будут выступать рабочие станции, а в качестве рёбер - различные виды связей между ними. Узлы сети мы будем считать абсолютно надёжными, соединения же подверженными отказам.

1.1 Актуальность работы

Различные виды сетей активно применяются для изучения вулканов [1], военного наблюдения [2], среды обитания [3].

Для различных видов сетей можно по разному трактовать значение исследуемого показателя надёжности, например:

1. Для информационных сетей - количество информации, которое мы можем передать между любой парой узлов.
2. Для транспортных сетей - количество грузов, которое мы можем перевезти между любой парой узлов.

В данной работе объектом исследования является система сетевой структуры, в которой в качестве модели используется случайный граф с абсолютно надёжными вершинами, моделирующими узлы, и ненадёжными рёбрами, моделирующими связи в сети. Пропускные способности каналов считаем одинаковыми в обоих направлениях, поэтому граф считаем неориентированным.

В качестве предмета исследования выбран такой показатель надёжности сети как матрица математических ожиданий величин максимальных потоков между всеми парами узлов в сети и соответствующее арифметическое среднее.

1.2 Цель магистерской работы

Целью магистерской работы является разработка и реализация алгоритма точного расчёта описанного показателя. Алгоритм должен абсолютно точно вычислять искомый показатель и завершаться за разумное время для как можно больших размерностей графов.

Для достижения указанной цели были поставлены следующие задачи:

1. Разработать точный алгоритм расчёта искомого показателя надёжности, завершающийся за разумное время.
2. Разработать и исследовать различные методы ускорения.
3. Реализовать алгоритм с помощью средств языка C++.
4. Реализовать параллельную версию алгоритма.

Методы исследования базируются на теории вероятностей и теории графов. Языком реализации выбран C++ для последующей адаптации и тести-

рования алгоритмов на кластере. Практическая значимость заключается в возможности точного расчёта указанного показателя надёжности, что в свою очередь позволяет ускорить и улучшить процесс проектирования надёжных сетей.

1.3 Цель курсовой работы

Целью курсовой работы является точное вычисление матрицы величин максимальных потоков между всеми парами узлов в надёжной сети.

Для достижения указанной цели, были поставлены следующие задачи:

1. Разработать точный алгоритм, вычисляющий искомую величину.
2. Реализовать и исследовать влияние различных методов ускорения.
3. Реализовать алгоритм с помощью средств языка C++.

2 Определения и обозначения

Используется модель неориентированного графа. Введем обозначения:

$G(V, E)$ – неориентированный граф с множеством вершин V и множеством ребер E .

c_{ij} – вес ребра между вершинами i и j .

f_{ij} – величина максимального потока между i и j .

3 Потоки в графах

Впервые задача о максимальном потоке была сформулирована в 1951 году Джорджем Данцигом. В 1955 году Лестер Форд и Делберт Фалкерсон впервые построили алгоритм для решения этой задачи. В течение последующих лет интерес к ней подогревался высокой практической значимостью и её применимостью при исследовании информационных и транспортных сетей. В 1970 году был опубликован частный случай алгоритма Форда-Фалкерсона, именуемый алгоритмом Эдмонда-Карпа. В 1986 году Гольдбергом и Тарьяном был опубликован алгоритм проталкивания предпотока. В дальнейшем решение задачи многократно улучшалось.

3.1 Определение

Прежде всего, необходимо ввести понятие потока.

Выделим две вершины в графе: источник, который будем обозначать s , и сток, который будем обозначать буквой t .

Поток - функция $f : V \times V \rightarrow \mathbb{R}$ со следующими свойствами для любых вершин u и v :

1. Ограничение пропускной способности – поток не может превысить пропускную способность: $f(u, v) \leq c(u, v)$.
2. Антисимметричность – поток из u в v должен быть противоположен потоку из v в u : $f(u, v) = -f(v, u)$.
3. Сохранение потока – $\sum_{w \in V} f(w, v) = 0$ для всех $w \in V \setminus \{s, t\}$.

Величина потока – сумма потоков из источника $|f| = \sum_{v \in V} f(s, v)$.

Максимальный поток между u и v – такой поток f , что его величина максимальна.

3.2 Стандартные алгоритмы

Наиболее стандартными алгоритмами вычисления максимального потока являются алгоритмы Эдмондса-Карпа, Диница [7] и проталкивания предпотока [8]. Асимптотическая сложность этих алгоритмов $O(|V| * |E|^2)$, $O(|V|^2 * |E|)$, $O(|V|^2 * |E|)$ соответственно. Несмотря на то, что сложность алгоритма Эдмондса-Карпа растёт быстрее, он более эффективно расходует память, что может положительно сказаться на производительности.

4 Методы ускорения расчёта

Методы ускорения расчёта различных показателей надёжности, как правило, сводятся к декомпозиции, редукции цепей и висячих вершин, а также рассмотрения других особенностей структур (наличие мостов, и точек сочленения, специальный вид графа, например, циклы или деревья и пр.) [9–11]. При этом возможность использования данных методов и их сложность существенно зависят от рассматриваемых показателей.

Так как исследуемые графы являются неориентированными, то поток в одну сторону равен потоку в другую, следовательно, итоговая матрица будет симметрична и нет необходимости считать поток между всеми парами – достаточно посчитать потоки для пар, находящихся над диагональю матрицы.

4.1 Разделение графа на несколько компонент связности

Исходный граф может иметь несколько компонент связности. Очевидно, между парами вершин в разных компонентах связности поток равен нулю. Таким образом, можно извлечь из графа его компоненты связности, найти потоки внутри них и затем собрать ответ из этих частей, предполагая, что между компонентами связности величины потоков равны нулю.

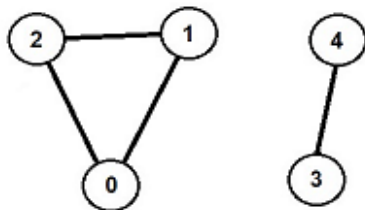


Рис. 1: Разделение графа на несколько компонент связности

Например, на графе, представленном на рисунке 1, $f_{ij} = 0$ для $i \in \{0, 1, 2\}, j \in \{3, 4\}$, так как эти вершины находятся в разных компонентах связности.

4.2 Стягивание цепи

В этом параграфе мы будем следовать обозначениям, введённым на рисунке 2 — $i_0, i_1 \dots i_m$ — вершины цепи, пронумерованные в порядке следования, u и v — любые вершины в графе, не обязательно не принадлежащие цепи.

При обнаружении цепи в графе мы можем применить следующую редукцию: заменить обнаруженную цепь на ребро, пропускная способность равна минимальной пропускной способности всех ребёр цепи, затем посчитать величины потоков в получившемся графе. Чтобы величину максимального потока

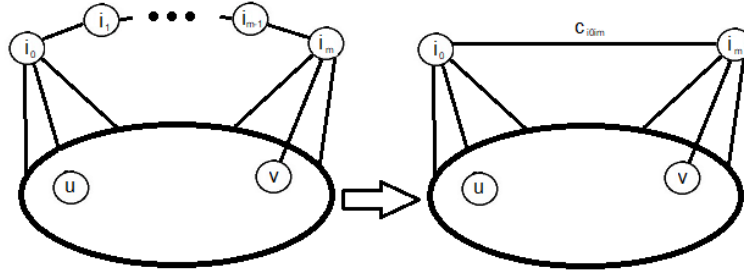


Рис. 2: Разделение графа на несколько компонент связности

между парой вершин в исходном графе, нужно рассмотреть следующие случаи:

1. Обе вершины не находятся в цепи.
2. Одна вершина находится в цепи, другая нет.
3. Обе вершины находятся в цепи.

1. Если обе вершины не находятся в цепи, то величина максимального потока между ними равна величине максимального потока между этими вершинами в преобразованном графе.

2. Если одна вершина (например i_p) находится в цепи, а другая нет, то величина максимального потока между ними равен сумме величин двух потоков - величине потока, который можно доставить до i_p с одной стороны, который равен $\min(f_{vi_0}, \min_{j \in [0 \dots p-1]} c_{ij}i_{j+1})$ и величине потока, который можно доставить с другой стороны, равной $\min(f_{vi_m}, \min_{j \in [p \dots m-1]} c_{ij}i_{j+1})$.

3. Если обе вершины (например i_p и i_k , $p < k$) находятся в цепи, то величина максимального потока между ними равна сумме двух величин потоков: величине максимального потока, который можно доставить по цепи, равная $\min_{j \in [p \dots k]} c_{ij}$, и величине максимального потока, который можно доставить через

остальной граф, равная $\min(\min_{j \in [0, p-1]} c_{ij i_{j+1}}, f_{i_0 i_m}, \min_{j \in [k, m-1]} c_{ij i_{j+1}})$.

4.3 Обработка мостов

Мостом называется такое ребро, удаление которого увеличивает число компонент связности. При обнаружении моста в графе (например, между вершинами i и j , как на рисунке 3) мы можем провести следующую редукцию: разделить на граф на два подграфа, разделённые мостом и посчитать потоки внутри этих подграфов. Чтобы получить величины потоков в исходном графе, нужно перебрать все пары вершин и рассмотреть два случая:

1. Обе вершины лежат в одном подграфе.
2. Вершины лежат в разных подграфах.

Если обе вершины лежат в одном подграфе, то величина потока между ними равна величине потока в этом подграфе. Если вершины лежат в разных подграфах (например u в первом, v во втором), то величина максимального потока между ними равна минимуму из величины максимального потока от u до моста, пропускной способности моста и величине потока от моста до v , т.е. $f_{uv} = \min(f_{ui}, c_{ij}, f_{jv})$.

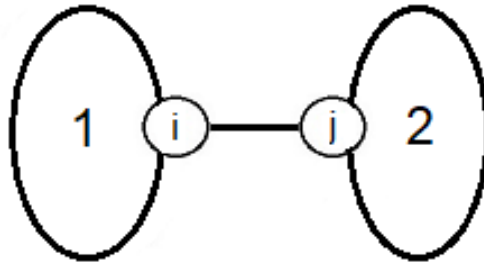


Рис. 3: Мост в графе

4.4 Удаление висячих вершин

Висячие вершины (вершины степени 1) могут быть обработаны следующим образом: удалим вершину и посчитаем величины потоков в оставшемся графе. Чтобы восстановить величины максимальных потоков в исходном графе, нужно рассмотреть два случая:

1. Обе вершины не являются висячими.
2. Одна вершина висячая, вторая нет.

Если обе вершины не висячие, то наличие висячей вершины никак не влияет на поток между ними, а следовательно величина потока между ними равна величине потока между ними в графе с удалённой висячей вершиной.

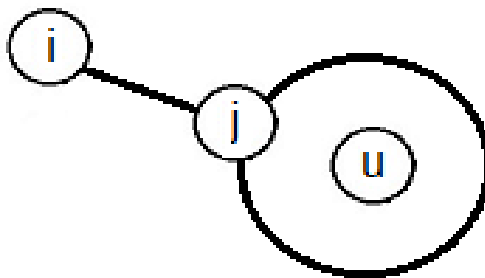


Рис. 4: Висячая вершина

Если одна вершина является висячей, а вторая нет, то величина потока между ними равна минимуму величины потока до опорной вершины и пропускной способности опорного ребра, то есть, следуя обозначениям из рисунка 4, $f_{ui} = \min(f_{uj}, c_{ij})$

5 Реализация алгоритмов и численные эксперименты

5.1 Алгоритм перебора пар

Простейшая версия алгоритма заключается в следующем: будем перебирать пары вершин, учитывая симметричность потоков, и считать величину максимального потока между ними с помощью стандартных алгоритмов.

5.2 Алгоритм с использованием редукций

Алгоритм с использованием редукций будет применять редукции следующим образом:

1. Разделяем граф на несколько компонент связности (описано в пункте 4.1).
2. Стягиваем цепи (описано в пункте 4.2).
3. Обрабатываем мосты (описано в пункте 4.3).
4. Удаляем висячие вершины (описано в пункте 4.4).
5. Пользуясь алгоритмом перебора пар получаем ответ.

Такой порядок действий обусловлен следующими соображениями: в процессе редукций не может быть увеличено число компонент связности, поэтому мы можем выделить их ещё на первом этапе. При обнаружении цепи-моста обработка мостов будет обрабатывать каждую вершину цепи, что существенно уступает обработке стянутой цепи, поэтому сначала идёт стягивание цепей и только за ней обработка мостов.

6 Реализация алгоритмов и численные эксперименты

Было реализовано 6 версий алгоритма:

1. Алгоритм перебора пар на базе алгоритма Эдмондса-Карпа.
2. Алгоритм перебора пар на базе алгоритма Диница.
3. Алгоритм перебора пар на базе алгоритма проталкивания предпотока.
4. Алгоритм с использованием редукций на базе алгоритма Эдмондса-Карпа.
5. Алгоритм с использованием редукций на базе алгоритма Диница.
6. Алгоритм с использованием редукций на базе алгоритма проталкивания предпотока.

Таблица 1: Результаты экспериментов

Версия алгоритма	1	2	3	4	5	6
(50,55)	0,127	0,289	0,65	0,051	0,091	0,15
(60,80)	0,31	0,786	1,52	0,162	0,681	1,25
(100,120)	2,12	5,35	63,61	0,9	1,76	9,59
(150,190)	10,6	29,20	470,56	3,46	10,48	98,35

Используемый процессор - Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz количество оперативной памяти - 8GB. Были случайным образом были сгенерированы 30 графов для каждой пары (количество вершин, количество рёбер) со случайными пропускными способностями в интервале (1, 100). В таблицу 1 занесено среднее арифметическое времени расчёта в секундах.

7 Заключение

В данной работе был рассмотрен такой показатель функциональности сети, как матрица величин поток между всеми парами вершин в графе. Для расчёта этого показателя был реализован алгоритм перебора пар и его модификации, которые ускорили алгоритм.

Таким образом, были решены следующие задачи:

1. Разработан алгоритм точного расчёта.
2. Предложены и реализованы различные методы ускорения вычислений.

В результате численных экспериментов были получены следующие результаты:

1. Алгоритм Эдмондса-Карпа показывает наилучшую производительность как в переборе пар, так и использовании редукций.
2. Получен существенный выигрыш по сравнению с перебором пар, который увеличивается с увеличением размерности.

Была заложена программная основа для дальнейших исследований, которые будут проводиться по следующим направлениям:

1. Решение задач поставленных в магистерской работе.
2. Поиск и анализ новых методов редукций.
3. Усовершенствование разработанных алгоритмов.

Список литературы

- [1] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10:18–25, 2006.
- [2] P. Kikiras and J. Avaritsiotis. Unattended ground sensor network for force protection. *Journal of Battlefield Technology*, 7(3):29–34, 2004.
- [3] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *First European Workshop on Wireless Sensor Networks*, pages 307–322, 2004.
- [4] Liu Sh., Cheng K., Liu X. Network reliability with node failures // *Networks*. 2000. V. 35. P. 109–117
- [5] Leslie G. Valiant The Complexity of enumeration and reliability problems / Valiant Leslie G. // *Society for Industrial and Applied Mathematics*. -1979. -№3. - P.410-421.
- [6] Moore E.F., Shannon C.E. Reliable Circuits Using Less Reliable Relays // *J. Franklin Inst.*, Vol. 262, n. 4b, 1956. P. 191–208.
- [7] Dinic E.A. Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation / E.A. Dinic // *Soviet Math Doklady*, - 1970. Vol. 11. - P. 1277-1280.
- [8] Goldberg A.V. A new approach to the maximum flow problem / A.V. Goldberg // *Journal of the ACM*. - 1988. - Vol. 35. - P. 921-940.

- [9] Shooman A. M. Algorithms for network reliability and connection availability analysis // Electro/95 Intern. Prof. Program Proc. Hynes Convention Center, Boston, MA, June 21–23, 1995. S. 1.: IEEE, 1995. P. 309–333
- [10] Мигов Д.А. Об одном показателе надёжности для сетей с отказами узлов / Д.А. Мигов // Средства и системы защиты информации и сетевых ресурсов. -2013. -№2. -С.43-48.
- [11] Родионов А.С., Родионова О.К. К вопросу практического использования формулы Муры-Шеннона для расчета вероятности связности локальных сетей // Тр. 2 Междунар. науч.-практ. конф. «Информационные технологии и радиосети», ИНФОРАДИО-2000. Омск, 2000. С. 67-69.