**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования**

**"Национальный исследовательский университет "Высшая школа экономики"**

***Отделение программной инженерии***

***Кафедра Управления разработкой программного обеспечения***

УТВЕРЖДАЮ

Зав. кафедрой УРПО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014г.

***ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА***

***по направлению 231000.62 Программная инженерия***

***подготовки бакалавра***

На тему: Программа вычисления стохастической степени связности неориентированных графов

Студента группы № 472ПИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чинкова С. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата)

Научный руководитель

д.т.н., профессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ульянов М. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Дата)

Москва, 2014 г.

# Реферат

## Сведения о работе:

Данная работа состоит из 51 страниц и содержит три основные главы:

* «Обзор решения задач в выбранной области»,
* «Используемые методы и алгоритмы»,
* «Реализация и анализ результатов».

Помимо этого в работе представлены 19 иллюстраций, 11 таблиц, 5 приложений и список источников. Было изучено и использовано 8 источников, в том числе и иностранных авторов и публицистов. Приложение содержит полный пакет документации из 5 документов. Список и приложения приведены в советующих разделах.

## Аннотация:

В работе исследовалось такое свойство графов как степень связности. При этом граф рассматривался как стохастическая система со случайным взаимодействием.

Целью работы является описание и разработка алгоритмов проверки связности графов, алгоритмов подсчета количества графов и экспериментов для них для предоставления достоверных результатов; предложить способы хранения и представления данных, реализовать графическую часть для демонстрации процессов потери связности.

Работа содержит описания теоретической части и практической реализации. Так же представлены результаты работы, заключение и документация. Конечный результат работы представляет собой программный продукт, написанный на языке программирования С++ с использованием SQLite (для хранения данных на компьютере для дальнейшего их использования) и OpenGL (для визуального оформления демонстрации) в среде разработки MS Visual Studio 2013. Программа предоставляет возможность выполнять задачи, описанные в техническом задании (Приложение А). А использование вышеперечисленных технологий повышает качество приложения.

## Ключевые слова:

Теория графов, невзвешенный неориентированный граф, связность графа, теория вероятностей, стохастическая система, математическая статистика, доверительный интервал, поиск пути в графе, матрица смежности.

**Содержание**

[Реферат 1](#_Toc389220770)

[Сведения о работе: 1](#_Toc389220771)

[Аннотация: 1](#_Toc389220772)

[Ключевые слова: 1](#_Toc389220773)

[Определения, обозначения, сокращения 4](#_Toc389220774)

[Введение 5](#_Toc389220775)

[Глава 1. Обзор решения задач в выбранной области 6](#_Toc389220776)

[Глава 2. Используемые методы и алгоритмы 8](#_Toc389220777)

[Условия и ограничения 8](#_Toc389220778)

[Первая подзадача 9](#_Toc389220779)

[Проблема представления графов 9](#_Toc389220780)

[Вторая подзадача 9](#_Toc389220781)

[Дополнительные задачи 10](#_Toc389220782)

[Глава 3. Реализация алгоритма и анализ результатов 12](#_Toc389220783)

[Использованные средства и технологии 12](#_Toc389220784)

[Измерения 12](#_Toc389220785)

[Основные функции 13](#_Toc389220786)

[Второстепенные функции 15](#_Toc389220787)

[Заключение 17](#_Toc389220788)

[Список использованных источников 18](#_Toc389220789)

# Определения, обозначения, сокращения

Далее мы будем говорить о **графе** как о **стохастической системе** со случайными процессами. Связано это с тем, что в графе в случайном порядке будут удаляться связи между узлами. Это означает, что заранее узнать, какое ребро будет удалено, невозможно.

# Введение

Связность графов является одним из основных понятий в теории графов [1]. В этой работе рассматривается степень связности неориентированных графов. А считать граф связным можно, только если для любой пары различных вершин существует цепь, соединяющая эти вершины. Если же для графа можно указать хоть одну пару различных вершин, которые не соединяются цепью (простой цепью), то граф называется несвязным. Этот вопрос тесно связан с проблемами теории сетевых потоков [2], [3]. Задавая любые сетевые системы в виде графа, можно проводить эксперименты по поиску надежной связности системы или проверки уже существующей структуры. Таким образом, исследование сводится к проведению определенного количества экспериментов над графом и поиску среднего числа ребер, которые можно удалить, прежде чем он распадется на несвязанные части.

Тем не менее, никогда нельзя узнать заранее, между какими двумя узлами в сети будет нарушена связь в дальнейшем (причина нарушения связи не имеет значения в данной работе). Таким образом, граф рассматривается как стохастическая система со случайным воздействием, приводящим к непредсказуемым нарушениям соединений. Процесс называется стохастическим, если он описывается случайными переменными, значения которых меняются во времени. А стохастичность системы в теории вероятностей характеризует ее как недетерминированную. Подобные системы играют фундаментальную роль в математических моделях.

Следовательно, эта работа лежит на пересечении таких областей как теория графов и теория вероятностей. Но при этом основная цель этой работы заключается в создании программного продукта, позволяющего пользователю сконструировать надежный неориентированный граф в качестве модели сетевой системы или проверить степень связности уже существующей. Основными целями являются: предложить и внедрить методику, описывающую класс графов, для реализации генерации графов на основе заданного описания и разработать эффективный по времени алгоритм динамической проверки связности графа на каждом этапе удаления ребер, основанный на изучении достоинств и недостатков существующих алгоритмов поиска пути. Кроме того, есть несколько несущественных задач, таких как: анализ и определение нужного размера выборки графов, относящихся к общему описанию, и экспериментов для каждого графа для получения надежных результатов; добавить демонстрационный режим для визуального ознакомления со стохастическими процессами потери связности в неориентированных графах.

В результате будет возможно оценить среднее число ребер (из определенного набора экспериментов), которые необходимо удалить, прежде чем граф распадается на отдельные и несвязанные между собой части и сравнить модели на основе полученных средних значений.

# Глава 1. Обзор решения задач в выбранной области

Проанализируем некоторые существующие способы, пути, алгоритмы, технологии решения подзадач, которые являются полезными и важными для изучения и понимания стохастической степени связности неориентированных графов.

Самые первые задачи теории графов восходят еще к Эйлеру (XVIII в.), хотя впервые термин «граф» появился в книге выдающегося венгерского математика Д. Кёнига в 1936 г. [4]. С тех пор появилось множество новых задач, терминов, понятий, решений, связанных с этим разделом дискетной математики. Теория графов содержит довольно большое количество проблем, нерешённых и по сей день, и пока не доказанных гипотез.

В данной работе под вниманием оказывается одна из многих характеристик графа: связность. Связный граф — это граф, в котором между любой парой вершин существует как минимум один путь. Прямым применением теории графов и, в частности, такой его характеристики как связность выступает теория сетей. Рассмотрим один всем известный пример: Интернет. Все компьютеры и устройства, которые подключены во всемирную сеть, образуют по своей сути некий связный граф. Отдельно взятая пара компьютеров или любых других устройств может и не быть соединена между собой напрямую, однако от каждого из них можно передать информацию друг другу или же любому другому. Этот пример как раз наглядно иллюстрирует определение связности.

Существует множество способов проверки связности графов, в том числе основанных на алгоритмах поиска пути в графах [5]. Самый распространенный – алгоритм маркировки вершин. Эти алгоритмы просты в исполнении, и некоторым достаточно один раз посетить каждый узел. Однако в проблеме вычисления стохастической степени связности неориентированных графов не все так просто и эти алгоритмы будут недостаточно продуктивными.

Рассматриваемая задача по вычислению стохастической степени связности неориентированных графов содержит в себе набор подзадач, относящиеся не только к теории графов. Чтобы разобраться более детально, приведем следующий пример. Возьмем пример с железнодорожным сообщением в городах какого-либо региона (уже упомянутый пример из теории сетей тоже может быть наглядным). Если по каким-либо причинам (причина не существенна в данном изучении) были нарушены железнодорожные пути между городами A и B. При этом каждый из них связан с другими городами C, D, E и т. д. В данной ситуации есть вероятность, что город A или B (и возможно некоторые другие города, имевшие возможность попасть в дальний город только через один из этих двух) оказались изолированными от общей системы железнодорожной сети. Значит, связанность оказалась нарушенной. Так же и в компьютерных сетях при разрыве, например, кабеля один или несколько узлов могут оказаться изолированными. Однако в обоих случаях разрыв имел непредсказуемый характер. Невозможно было предугадать, какая связь будет выведена из строя. Это значит, что процесс имеет случайный характер. В данном ключе можно рассматривать граф, описывающий модель железнодорожного сообщения и компьютерной сети, как стохастическую систему. Ведь стохастическая система – это система, изменения в которой носят случайный характер. При случайных воздействиях, данных о состоянии системы недостаточно для предсказания событий в последующий момент времени. В данной работе граф рассматривается как стохастическая система со случайными воздействиями, приводящими к потере ребер и, в конечном счете, связности.

Вышеприведенные доводы приводят к необходимости совершенствования алгоритмов, проверяющих связность графов. Это открывает возможности для избегания алгоритмов обхода всех звеньев в графе, таких как поиски в глубину и в ширину.

Так как получаемые данные являются экспериментальными, то для их достоверности необходимо верно определить размер выборки. В теории вероятностей существую методы поиска объема выборки и доверительных интервалов для них.

Основываясь на приведенные выше доводы, анализы, алгоритмы, методы можно создать программу, удовлетворяющую всем требованиям и позволяющую вычислять стохастическую степень связности неориентированных графов.

# Глава 2. Используемые методы и алгоритмы

## Условия и ограничения

Прежде всего, следует уточнить условия и ограничения в поставленной задаче, что позволит найти направление для дальнейшей работы.

* Во-первых, важно обратить внимание, что рассматриваемые графы являются неориентированными и невзвешенными. Это значит, что в данной работе рассматриваются только самые простейшие виды графов. Это позволяет избежать лишних затрат памяти и времени при расчете весов ребер и их направлений. Так же узлы графов не имеют петель, связей элементов самих в себя.
* Во-вторых, граф рассматривается как стохастическая система со случайными процессами, приводящими к потере ребер. Это значит, что следует учитывать случайный характер удаления связей при проведении экспериментов.
* В-третьих, графы в основном будут задаваться в виде некоторого описания: $S=\{a,b,c,d,…\}$, где порядковый номер параметров равен номеру узла, $a,b,c,d,…$ - количество узлов, с которыми i-тый узел имеет связи причем $a\geq b\geq c\geq d\geq …$. К тому же каждое значение в описание указывает не уникальное количество связей, а общее. Это значит, что значения в описании полностью соответствуют узлам и их связям. Это значит, что каждая связь в описании указывается дважды. А значит, сумма всех значения должна быть четной, чтобы граф можно было построить, следуя инструкции *S*. Эти условия необходимо учесть при разработке проверок формата описания, введенного пользователем.
* В-четвертых, по описанию S можно построить большое количество разных графов и с каждым из них нужно провести определенное количество экспериментов. Объемы этих выборок нельзя выбирать случайно, по желанию. Для этого необходимо прибегнуть к методам теории вероятностей по поиску оптимального объема выборок графов для описания S и оптимального количество необходимых прогонов каждого графа по алгоритму удаления ребер. Это предоставляет нам возможность выдавать достоверные статистические данные с точностью, которую пожелает указать сам пользователь.

Проблема вычисления стохастической степени связности неориентированных графов включает в себя несколько подзадач. Рассмотрим их в порядке прохождения в программе.

## Первая подзадача

Первая подзадача заключается в нахождении оптимального объема выборки графов для описания S. Выборочная совокупность (или выборка) — это отобранное по особому строго заданному правилу определенное число элементов генеральной совокупности. Это число обеспечивает некоторую достоверность и надёжность результатов того явления, которое мы рассматриваем. Объем выборки может быть вычислен по формуле $N=\frac{t^{2}δ^{2}}{∆^{2}}$ [6], где *N* — объем выборки; δ — найденное среднее квадратичное отклонение; δ2 — дисперсия; *t* — t-критерий, зависящий от числа планируемых опытов (определяется по стандартной таблице значений t-критерия Стьюдента); Δ — ожидаемое значение предельной ошибки выборки. Такой параметр как относительная погрешность в программе будет указываться пользователем в настройках для регулировки подходящей точности, которая будет необходима пользователю. Однако необходимо понимать, что уменьшение значения погрешности приведет к дополнительным временным затратам. В то же время, в этом случае статистические данные будут более точными. В дальнейшем, используя эту же методологию, необходимо найти оптимальное количество проводимых экспериментов для каждого сгенерированного графа. Тогда можно будет гарантировать точность результата.

## Проблема представления графов

Чтобы работать с графами их необходимо как-то представлять в программе. «Есть два стандартных способа представить граф G = (V, E): как набор списков смежности или как матрицу смежности. Оба способа применимы как к ориентированным, так и к неориентированным графам» [7] (стр. 527-531). Однако только с матрицей смежности можно быстро сказать, что существует связь, соединяющая две вершины. К тому же возможно использовать треугольную матрицу смежности по причине отсутствия ориентации в связях графов. В этом случае матрица будет симметричной относительно главной диагонали. Это позволит ускорить работу алгоритмов, работающих непосредственно с графами.

## Вторая подзадача

Следующий этап – случайное удаление ребер и проверка связности графов при каждой итерации. Необходимость случайного удаления была пояснена выше в начале главы (Глава 2) и в главе «Определения, обозначения, сокращения». После того, как связь между двумя любыми узлами, которые имели эту связь ранее, была потеряна, необходимо проверить граф на критерий связности, чтобы узнать, не распался ли он на две части. В связи с тем, что оба звена графа, потерявшие связь, известны, проверка связности сводится к поиску пути в графе от первого узла ко второму. Это значит, что алгоритм освобождается от необходимости искать путь к каждому узлу и проверять все пути. Можно отказаться от тяжелых алгоритмов, таких как поиски в глубину и в ширину. Теперь можно обратить внимание на такие алгоритмы как: алгоритмы Дейкстры, Флойда, поиска А\*. Это значительно увеличивает общую эффективность и скорость работы модуля по поиску стохастической связности в группе графов, описываемых через *S*.

Последовательность шагов в этом этапе будет повторяться до тех пор, пока граф не распадется на несвязанные части, т.е. появится хотя бы одна пара вершин, которые не будут иметь пути между собой при разрыве связи между ними. Однако на каждом этапе удаления мы будем получать каждый раз разный граф и пару элементов, между которыми раньше была связь. При этом на каждой итерации этот граф будет похож на граф из предыдущего шага цикла с разницей лишь в одну связь. Запись уже пройденных путей при поиске пути в предыдущих шагах цикла позволит быстрее выявить потерю связности. Для этого достаточно найти те пути из списка, в которых участвовали ребра удаленной связи. После чего не составит проблем составить путь по совпадающим элементам. Таким образом, можно быстрее искать пути между узлами, если они еще имеются в графе

В конечном счете, результатом работы будет предоставление среднего числа удаленных ребер по всем проведенным экспериментам для всех сгенерированных графов, общего количества сгенерированных графов, найденного с помощью методов поиска объема выборки и суммарное количество экспериментов, которые были проведены со всеми графами.

## Дополнительные задачи

Так же в программе предусмотрен наглядный демонстрационный режим, в котором можно проследить визуально за процессом разделения графа на несвязанные части. Достаточно будет задать конкретный граф, и после программа приступит к удалению ребер в нем вплоть до потери связности. Это позволит проводить разовые наглядные визуальные испытания определенных и конкретных систем (будь то компьютерные или транспортные системы, которые пользователь моделирует в виде неориентированных графов). Этот модуль реализуется на основе указанных выше алгоритмов создания, испытания и проверки графов с использованием графических низкоуровневых библиотек OpenGL для графического оформления работы модуля.

При всем при этом в локальную базу данных будут записываться несколько последних (объем пользователь сможет указать в настройках) созданных описаний графов S и самих графов для демонстрационного модуля. Сообщение с БД производится с помощью библиотек SQLite на языке SQL-запросов.

# Глава 3. Реализация алгоритма и анализ результатов

## Использованные средства и технологии

Для реализации был выбран язык программирования С++, т.к. он является одним из самых лучших, востребованных языков. К тому же С++ является кроссплатформенным языком программирования. В качестве среды разработки использовалась Microsoft Visual Studio 2013 Professional.

Помимо этого в приложении использована структура хранения данных в виде локальной базы данных для получения более удобной и организованной формы хранения информации. БД создана с помощью SQLite, а в программе сообщение с БД выполняется посредством языка SQL-запросов.

Для графического оформления демонстрационного режима были задействованы графические библиотеки OpenGL.

## Измерения

*Рисунок 1*

### Основные функции

После реализации основных алгоритмов программы (Рис.1) было необходимо провести эксперименты для замеров и оценки эффективности алгоритмов. После проведения некоторого числа экспериментов над программой были получены данные.

*Рисунок 2*

Среднее время работы программы представлено на рисунке 2. Вертикальная ось отображает время в тиках, горизонтальная – количество узлов в графе. Три кривые отображают зависимость времени работы от количества узлов при относительной погрешности $ε=0,1; ε=0,05 и ε=0,01$. Таким образом, видно, что, чем ниже относительная погрешность, тем больше времени затрачивается на расчеты, однако при этом растет точность.

*Рисунок 3*

На рисунке 3 представлены данные проведения экспериментов с одним графом. Точно так же по горизонтали указано количество узлов в графе, по вертикали – время в тиках. В данном случае в расчеты вошло время вычисления объема выборки, случайное удаление ребер в графе и проверка связности графа на каждом этапе удаления. Не трудно заметить, что уже при $ε=0,01$(зеленая линия на графике) на вычисления времени уходит гораздо больше. Это связано с тем, что для большей точности данных требуется больший объем выборки.

Для наглядности рассмотрим пример графа с описанием $S=\{4,3,3,2,2\}$ (Рисунок 4).

*Рисунок 4*

Горизонтальная ось показывает значения относительной погрешности. Вертикальная ось так же отображает время в тиках. В данном случае четко видно, как резко возрастает время работы из-за повышения точности результата.

### Второстепенные функции

Одной из второстепенных функций программы является предоставление демонстрационного режима для наглядного ознакомления или отслеживания процесса потери связности в графе. Было решено реализовать это с использованием графической библиотеки OpenGL (#include <GL/gl.h> и #include <GL/glu.h>). В этом режиме можно пошагово удалять случайные ребра до тех пор, пока граф не распадется на две несвязанные части.

*Рисунок 5*

# Заключение

Главной целью выпускной квалификационной работы являлось создание программного обеспечения, позволяющего вычислять стохастическую степень связности неориентированных графов, что открывает для пользователя возможность сконструировать свою модель и протестировать ее на устойчивость к дроблению (этими системами могут быть компьютерные сети, транспортное сообщение и др.). Для достижения этой цели было изучено множество материала, в том числе иностранных авторов и публицистов, множество подходов решения подзадач данной работы. Это позволило проработать все алгоритмы, вошедшие в работу и позволяющие в совокупности вычислять требуемые данные.

А именно программа позволяет находить среднее число ребер, которое достаточно удалить из системы, чтобы она распалась. И не менее важно, что предоставляемые данные можно считать достоверными, так как объемы выборок для нахождения статистических данных вычисляются по всем правилам математической статистики. А так же программа позволяет ознакомиться с принципами изучаемого явления в наглядном демонстрационном режиме.

Для реализации всех подзадач было задействовано определенное количество технологий, средств разработки и стороннего ПО, что в результате положительно сказалось на программе, и она продемонстрировала ожидаемые от нее результаты даже с учетом скачков практических данных. Программа достаточно быстро считает результат на небольших графах, чтобы не доставлять пользователю дискомфорт при работе. Однако для работы с большими данными и ожидаемой высокой точностью есть большой шанс, что времени будет затрачено много, т.к. при этом условии объемы вычислений резко возрастают.

Данная работа может иметь дальнейшее развитие для ее последующего внедрения в решение реальных практических задач. В частности, автоматизация данных, или внедрение ПО в определенную систему для периодического мониторинга ее состояния для более точных и частных результатов. Работа затрагивает сразу несколько областей науки. Это позволяет ей развиваться в дальнейшем по многим направлениям сразу, повышая свою ценность и полезность.

# Список использованных источников

1. Diestel, R. (2005). *Graph Theory*, Electronic Edition
2. Nagamochi, H., Ibaraki, T. (2008). *Algorithmic Aspects of Graph Connectivity*. Cambridge University Press.
3. Блог iZerO.Ru: [http://](http://www.izero.ru/windows/kompyuternye-seti/kompyuternye-seti-kak-oni-ustroeny-topologii-setej.html)www.izero.ru/windows/kompyuternye-seti/kompyuternye-seti-kak-oni-ustroeny-topologii-setej.html - Компьютерные сети. Как они устроены. Топологии сетей
4. М.Э. Рояк, С.Х. Рояк. «Теория графов. Методические указания к практическим занятиям и выполнению РГР по курсу «Дискретная математика»» Новосибирский Государственный Технический Университет
5. Bryan, S. (1997). Looking Before You Leap. *Smart Moves: Intelligent Pathfinding,* Game Developer Magazine, July.
6. Шведов А. С. «Теория вероятностей и математическая статистика» Изд. дом ГУ-ВШЭ, 2005 г.
7. Cormen, Thomas, H. Leiserson, Charles, E. Rivest, Ronald, L. Stein, Clifford. (2001). "Section 22.1: Representations of graphs". *Introduction to Algorithms* (Second edition), MIT Press and McGraw-Hill.
8. Антонова Анастасия Анатольевна. «Исследование сложных нестационарных телекоммуникационных систем и разработка метода управления потоками данных» Автореферат диссертации на соискание ученой степени кандидата технических наук. ФГБОУ ВПО НИУ Московский энергетический институт