МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Раскраски и ориентации графов

АВТОРЕФЕРАТ
дипломной работы

студентки 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Сажиной Елизаветы Максимовны

Научный руководитель
д.ф.-м.н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Саратов 2024

ВВЕДЕНИЕ

С каждым годом появляется всё больше новых технологий в области биологии, медицины, физики, информатики. Это требует разработки новых эффективных алгоритмов для работы с задачами различной сложности и инновационных подходов к формированию структур данных. В связи с этим особую важность приобрели те разделы математики, которые имеют отношение к развитию цифровых устройств и цифровых вычислительных машин.

При обучении будущих специалистов, которые смогут работать с перспективными технологиями и развивать их и дальше, применяется такая дисциплина, как дискретная математика. Её особым разделом для изучения стала теория графов.

Из широкого аспекта тем в теории графов, в настоящей работе будут рассматриваться те области, которые касаются ориентаций, автоморфизмов, раскрасок и чисел различимости графов. Это активно исследуемые темы, и многое изучено в данных областях. Различные алгоритмы для нахождения вышеуказанных характеристик графа уже существуют и доказаны, но остается много нерешенных проблем и вариаций.

Актуальность темы данной работы заключается в исследовательском интересе, так как теория графов предлагает множество математических концепций, что способствует развитию новых теоретических результатов, которые затем могут быть применены на практике.

Целью настоящей работы является теоретическое изучение общих сведений о графах и практическая реализация программы на языке C#, которая будет ориентировать графы, подсчитывать число автоморфизмов и раскрасок графов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить теоретический материал в области ориентаций, автоморфизмов, раскрасок и чисел различимости графа;

2. Изучить теоретический материал в области алгоритмов ориентации графов, поиске их автоморфизмов и раскраски графов;

3. Осуществить практическую реализацию программного кода, который позволит провести исследования, необходимые для данной работы.

Дипломная работа состоит из введения, 7 разделов, заключения, списка использованных источников и 3 приложений. Общий объем работы – 70 страниц, из них 41 страница – основное содержание, включая 39 рисунков и 1 таблица, список использованных источников из 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

В первом разделе приводятся теоретические понятия, необходимые для дальнейшего рассмотрения темы. Здесь излагаются основные определения и обозначения, относящиеся к теории графов.

Граф – математическая абстракция реальной системы объектов любой природы, обладающих парными связями. Граф как математический объект есть совокупность двух множеств – множества самих объектов, называемого множеством вершин и множеством их парных связей, называемой множеством рёбер. Элемент множества рёбер есть пара элементов множества вершин.

Ориентированным графом или *орграфом* называется пара $G = (V, \alpha)$, где α – отношение на множестве вершин V, называемое *отношением смежности.* Элементы множества α называются *дугами*. Если $(u, v) \in \alpha$, то говорят, что u – *начало дуги*, а v – *конец дуги*. Дуга вида $(u, u) \in \alpha$ называется *петлей*. Вершина, не являющаяся началом никакой дуги, кроме, быть может, петли, называется *стоком*, а вершина, не являющаяся концом никакой дуги, кроме, быть может, петли, называется *источником*. Орграф $G = (V, \alpha)$ называется *направленным графом* или *диграфом*, если отношение α антисимметрично. Полный направленный граф называется *турниром.*

Неориентированным графом (далее *графом*) называется пара $G = (V, \alpha)$, где α – симметричное и антирефлексивное отношение на множестве вершин V. Если $(u, v) \in \alpha$, то говорят, что вершины u и v *смежны* и эти вершины соединены *ребром* (u, v). При этом (u, v) и (v, u) это одно и то же ребро, которое обозначают $\{u, v\}$. Говорят, что ребро $\{u, v\}$ *инцидентно* каждой из вершин u и v и эти вершины называются *концевыми вершинами* или *концами* ребра $\{u, v\}$. Два ребра называются *смежными*, если они имеют общую концевую вершину.

Два графа $G_1 = (V_1, \alpha_1)$ и $G_2 = (V_2, \alpha_2)$ называются *изоморфными*, если можно установить взаимно однозначное соответствие $f: V_1 \to V_2$, сохраняющее

отношение смежности: $(u, v) \in \alpha_1 \Rightarrow \big(f(u), f(v)\big) \in \alpha_2$ для любых $u, v \in V_1$. В этом случае пишут $G_1 \cong G_2$.

Автоморфизмом графа G называется изоморфизм графа G на себя. Автоморфизмы графа образуют группу. Любой граф имеет по крайней мере один тождественный автоморфизм. Граф, который имеет только один тождественный автоморфизм называется *тождественным* или *ассиметричным*.

Раскраской графа G в k цветов называется функция $\rho: v(G) \to M$, где $|M| = k$. Раскраска ρ называется правильной, если $\rho(v) \neq \rho(u)$ для любой пары смежных вершин u и v.

Во втором разделе рассматривается программное обеспечение Cygwin с пакетом программ nauty и программой geng. Данный программный комплекс необходим для выполнения поставленных в работе задач. Программная утилита geng, непосредственно, позволяет генерировать неориентированные графы с заданным числом вершин.

В третьем разделе рассматриваются способы ориентации графа. Данный раздел содержит два подраздела. Это обусловлено наличием в работе теоретического описания вариантов ориентации рёбер графа и практической реализации ориентации графов.

Существует несколько вариантов ориентации рёбер графа. Первый случай, когда ребро заменяется дугой, тогда существует два варианта ориентаций. Второй случай, когда ребро заменяется парой дуг. В данной работе второй случай рассматриваться не будет. Для первого случая ориентация будет производиться следующим образом: рёбра неориентированного графа будут ориентироваться сначала в одну, а затем в другую сторону. При этом, сначала будут рассмотрены изоморфные, а затем неизоморфные ориентации графа. Иллюстрация ориентирования графа проводится на примере графа с пятью вершинами, которые образуют цикл.

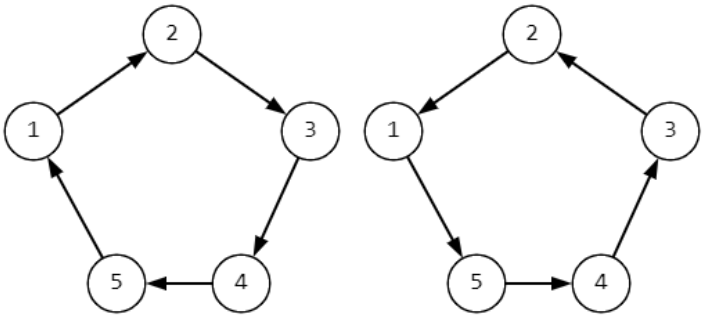

Рисунок 1 – Вариант изоморфной ориентации графа с пятью вершинами, которые образуют цикл

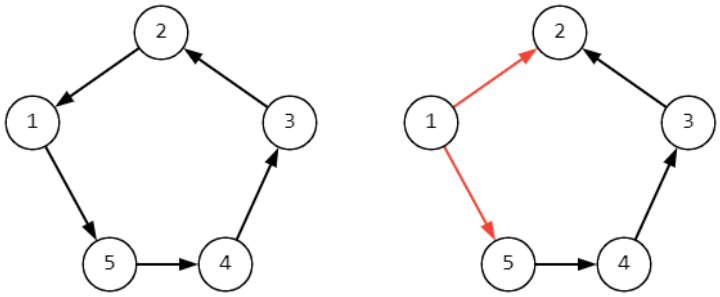

Рисунок 2 – Вариант неизоморфной ориентации графа с пятью вершинами, которые образуют цикл

При проверке изоморфных копий, происходит отображение всех вершин на все. То есть, получается большое количество графов, а следовательно, получается большое количество кодов. Далее, среди всех кодов выбирается наибольший. Он отображается в число и далее работа происходит с числом. Это наибольшее число и является каноническим кодом. Также стоит отметить, что среди всех рассмотренных кодов может встретиться несколько кодов, равных каноническому. Количество таких графов будет являться количеством автоморфизмов искомого ориентированного графа с каноническим кодом.

Таким образом, вместе с поиском неизоморфных ориентаций графа производится подсчёт автоморфизмов этих графов.

Раздел четыре, аналогично третьему разделу, состоит из двух частей, в которых теоретически рассматривается способ правильной раскраски графа, а затем, описывается практическая реализация раскраски графа путём применения Жадного алгоритма. Раскраска рассматривается на примере полного неориентированного графа K_5. Для выполнения правильной раскраски

графу K_5 понадобилось количество цветов, равное 5. Иллюстрация раскраски графа представлена на рисунке 3.

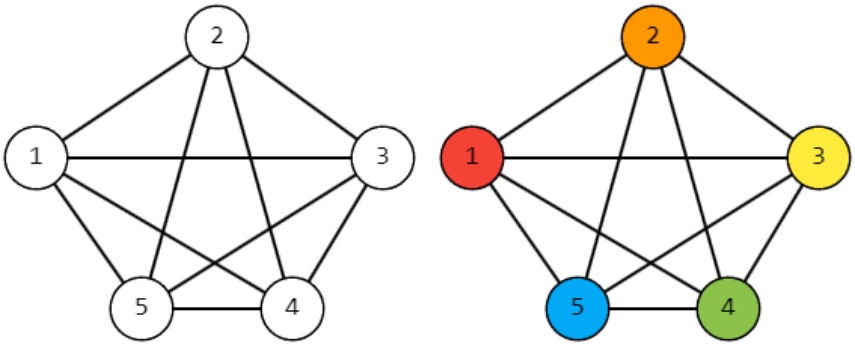

Рисунок 3 – Полный неориентированный граф с пятью вершинами. Пример его раскраски в пять цветов

Раскраска графов является инвариантным свойством под действием группы автоморфизмов. Это означает, что если граф имеет автоморфизм, который переводит одну раскраску в другую, то эти две раскраски считаются эквивалентными. В свою очередь, количество неэквивалентных раскрасок графа является важной характеристикой, так как она показывает, насколько граф симметричен.

Раскраска и ориентация графа – это два разных представления графа. Связь между ними может быть неочевидной. Однако, одно из возможных направлений исследования – рассмотреть ориентированный граф, полученный из раскраски графа, где ребра направляются от вершин с меньшими номерами цветов к вершинам с большими номерами цветов. Это может привести к интересным связям между характеристиками ориентированных графов и раскрасок исходного графа.

В пятом разделе работы описывается разработанная и реализованная программа «GraphOrientations», исполнимым файлом которой является «GraphOrientationsWithGroups». Программа позволяет ориентировать графы, подсчитывать количество их автоморфизмов и раскрасок. Программа написана на языке программирования C# в среде разработки Microsoft Visual Studio 2022. Иллюстрация работы программы представлена на рисунке 4.

Помимо основной программы, для визуализации получившихся результатов и построения графиков, позволяющих анализировать зависимости различных показателей графов, была разработана вспомогательная программа. Она выполнена на языке программирования Python в среде разработки PyCharm 2022.2.1. Для удобства в данной работе использовался бесплатный облачный сервис от Google – Google Colab, предназначенный для программистов, которые работают на языке Python. Интерфейс вспомогательной программы можно видеть на рисунке 5.

Также, на языке программирования Си++ была реализована ещё одна вспомогательная программа, позволяющая переводить графы из исходного формата G6 в матрицу смежности.

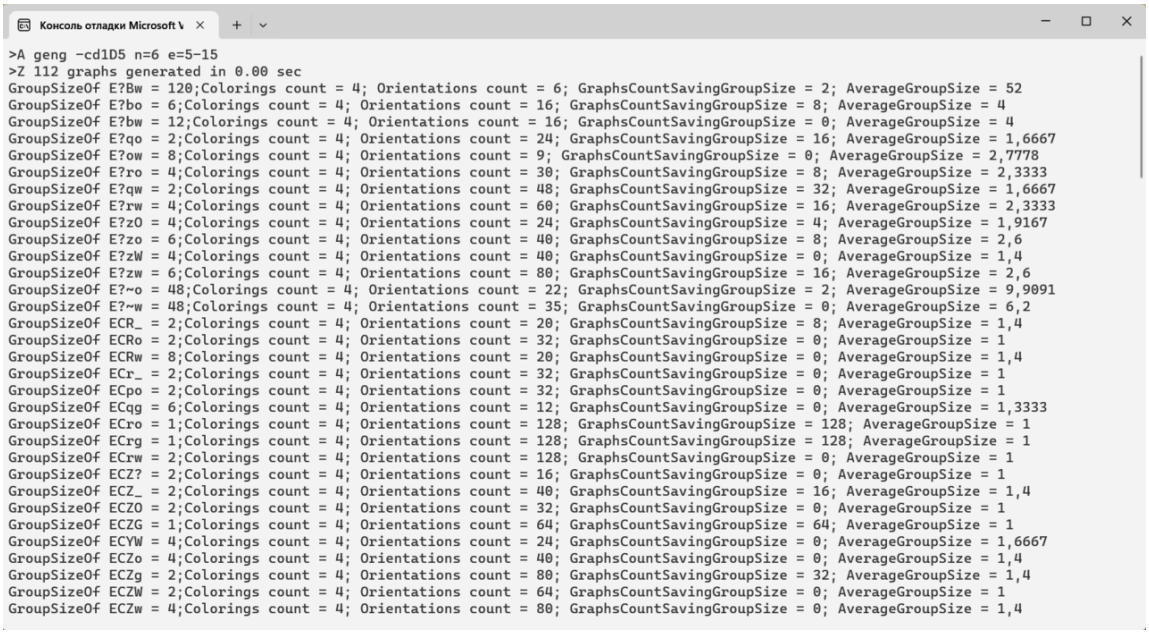

Рисунок 4 – Пример работы программы для графов с шестью вершинами

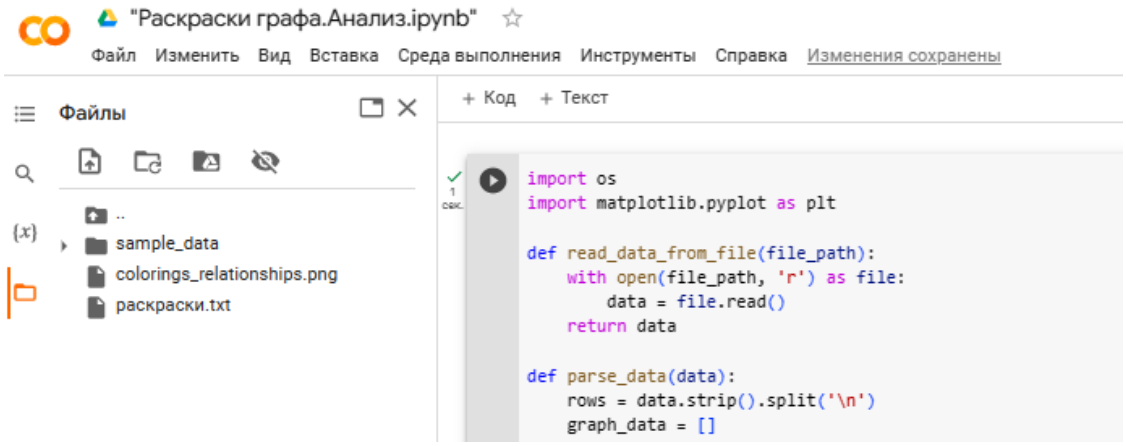

Рисунок 5 – Интерфейс блокнота Google Colab и файл «раскраски.txt»

В разделе шесть осуществляется проверка работоспособности программы «GraphOrientations». Для осуществления проверки работоспособности программы необходимо сравнить значения, полученные в ходе работы программы без использования утилиты nauty со значениями, полученными в ходе работы программы с использованием утилиты nauty.

В пакете Cygwin есть программа directg, которая строит все ориентации для заданного неориентированного графа. На входе программа получает файл с неориентированными графами, на выходе программа составляет файл с ориентированными графами.

Также, в пакете Cygwin есть программа pickg, которая позволяет смотреть различные свойства графов, в том числе, находить количество автоморфизмов.

Для осуществления раскрасок графа, в пакете Cygwin находится программа vcolg.

По итогам проверки, выявлено, что программа работает корректно. Единственным отличием является время работы программы. При применении утилиты nauty программа работает медленнее. Такое отличие обуславливается тем, что при не большом числе вершин у графа, утилита nauty затрачивает время на обмен данными через файлы.

В случае графов с большим числом вершин, программа с утилитой nauty будет работать быстрее программы без утилиты nauty.

В разделе семь выполняется поиск зависимостей между параметрами графа. Сначала исследуется зависимость числа раскрасок графов от количества цветов для разукрашивания. Данное исследование проводилось на восьмивершинном графе, количество цветов для разукрашивания равно 4. По итогам проведённой работы был получен вывод, что для одних и тех же графов разное количество цветов даёт одно, константное, количество раскрасок.

Другим направлением для проведения исследований является поиск зависимости между ориентациями, раскрасками и автоморфизмами графов. Для нахождения данной зависимости можно осуществить подсчёт количества ориентированных графов, сохраняющих размер группы, то есть, сохраняющих

количество автоморфизмов. Подсчёт предлагается осуществить на графах из восьми вершин. Результаты исследований представлены на графике на рисунке 6.

Рисунок 6 – График зависимости среднего количества ориентаций и раскрасок от числа автоморфизмов для графов с восемью вершинами

Было установлено, что значение количества ориентаций графа зависит от числа автоморфизмов следующим образом: чем больше число автоморфизмов, тем меньше количество ориентаций в целом.

Другой вывод, полученный в результате эксперимента: ориентация имеет число автоморфизмов не больше, чем исходный неориентированный граф.

Также, было выявлено, что не у каждого графа есть ориентация тождественная, то есть с одним автоморфизмом. Примером может послужить следующий неориентированный граф, который представлен на рисунке 38. Если ориентировать данный граф, как показано на рисунке 39, то в любом случае хотя бы две ориентации будут изоморфны.

Кроме вышеперечисленного, по данной работе стоит отметить, что ориентации тождественного графа являются тождественными графами.

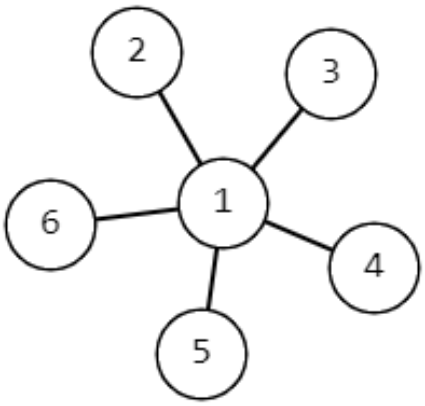

Рисунок 7 – Пример неориентированного графа

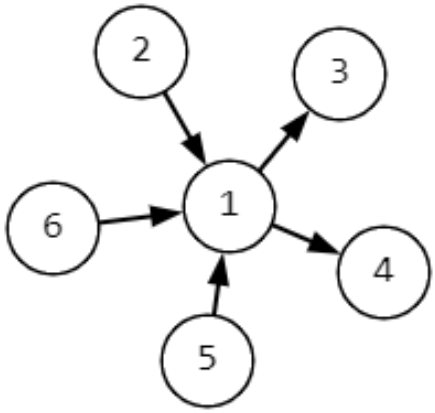

Рисунок 8 – Пример ориентации графа

Исходя из полученных данных, можно сделать вывод, что раскраски, автоморфизмы и ориентации графа являются различными аспектами изучения графов, и их связи могут быть неочевидными и сложными. Однако, комбинированный анализ этих характеристик может привести к более глубокому пониманию структуры и свойств графов. Программа, реализованная в настоящей работе, предоставляет значения этих характеристик для различных графов, что может быть полезным для дальнейшего анализа и изучения связей между этими показателями.

ЗАКЛЮЧЕНИЕ

В данной работе рассматривался материал, касающийся ориентации графов, их автоморфизмов и раскрасок.

В результате проделанной работы, разработана и реализована на языке программирования C# программа «GraphOrientations», которая ориентирует сгенерированные графы, считает количество их автоморфизмов и осуществляет раскраску графа.

Учитывая полученные в работе выводы, можно сказать, что решение задачи включает в себя все возможные альтернативы и варианты, чтобы полностью охватить различные сценарии, следовательно, оно представлено в полном объеме, без упущений или пропусков.

Результаты данной работы могут быть полезны при дальнейшем изучении теории графов, а именно, разделов, которые касаются ориентации графов, поиска их автоморфизмов и раскрасок. На практике, полученные результаты могут быть использованы в сфере логистики, машинном обучении, телекоммуникациях, биоинформатике.

Предложенные решения являются дополнениями к существующим решениям и достижениям в данной области. Они показывают высокую эффективность и смогут позволить сэкономить многие ресурсы при решении различных задач.

Таким образом, цели, поставленные в начале работы, были достигнуты.