DOI 10.37882/2223-2966.2024.7.35

СВЯЗЬ ГАМИЛЬТОНОВОСТИ И ПЛОТНОСТИ ОТНОШЕНИЯ СМЕЖНОСТИ ГРАФА

RELATION OF HAMILTONIAN AND DENSITY OF THE GRAPH ADJACENCY RELATION

D. Smirnov

Summary. In the process of considering pairs of different characteristics of graphs, a search was made for a possible correlation between any one pair of such characteristics. As a result, a conjecture about pancyclic graphs was formed, which is the main conclusion of the experiments. The calculations were carried out using the parallelization of the calculation process, all source codes were written in the object-oriented language C ++ and executed under the control of a Unix-like Debian open source community system. In order for the calculations not to become too time-consuming, dynamic programming techniques were used to search for the Hamiltonian cycle, however, even with this optimization, the graphs generated by the Nauty package are considered only up to 12 vertices, since it becomes too laborious to carry out calculations on a larger number of vertices even with optimizations.

Keywords: graph, cycle, pancyclicity, dynamic programming, parallel programming.

Смирнов Данила Антонович
учитель, Физико-технический лицей №1, г. Саратов
sad.97@list.ru

Аннотация. В процессе рассмотрения пар различных характеристик графов был произведен поиск возможной связи между какой-нибудь одной парой таких характеристик. В результате сформирована гипотеза о панциклических графах, являющаяся основным выводом экспериментов. Расчеты проводились с применением распараллеливания процесса вычисления, все исходные коды написаны на объектно-ориентированном языке C++ и исполнялись под управлением системы Unix-подобной Debian opensource-сообщества. Для того, чтобы расчеты не становились слишком требовательны по времени применены техники динамического программирования для поиска Гамильтонова цикла, однако даже при данной оптимизации генерируемые пакетом Nauty графы рассмотрены только до 12 вершин, так как на большем числе вершин расчеты даже с оптимизациями проводить становится чересчур трудоемко.

Ключевые слова: граф, цикл, панцикличность, динамическое программирование, параллельное программирование.

Введение

Теория графов в современном мире нашла применение в таких областях как компьютерные сети, построение оптимальных маршрутов, строительство железных дорог и мостов, создание схем электронно-вычислительных машин, схем авиалиний, карт звездного неба, построение генеалогического древа.

В данной научной работе было рассмотрены характеристики графов с целью выявить возможную связь каких-либо характеристик.

Основным результатом является гипотеза о панциклических графах.

Тема панцикличности была рассмотрена в [5], [6], [7], [8].

Теоретическая часть

Рассмотрим используемую в работе теорему, полученную в ходе работы гипотезу и используемые алгоритмы. Будем считать, что все вершины пронумерованы с 0 до $n-1$ и n — число вершин в рассматриваемом графе. Если говорится о вершине i, подразумевается вершина с номером i. Во всех алгоритмах полагается $n \geq 3$, причем все графы связные, следовательно и число ребер $m \geq n-1$, так как минимальное число ребер, необходимое, чтобы соединить n вершин равно $n-1$.

Отношение ϕ на множестве A называется плотным если для любых $x, y \in A$, отличных друг от друга и таких, что $(x, y) \in \phi$, найдется $z \in A$, отличный и от x и от y, такой, что $(x, z) \in \phi$ и $(z, y) \in \phi$ [4].

Граф называется панциклическим, если содержит все циклы от треугольника до гамильтонова [2].

Теорема Дирака

Пусть G — неориентированный граф и w — минимальная степень его вершин. Если $n \geq 3$ и $w \geq \frac{n}{2}$, то G — гамильтонов граф [2].

Гипотеза о панциклических графах

С благодарностью Богу, в ходе работы была получена следующая гипотеза: все графы, удовлетворяющие теореме Дирака и имеющие плотное отношение смежности — панциклические.

Проверка проводилась на всех связных графах имеющих от 3 до 12 вершин включительно.

Условия теоремы Дирака

Сложность проверки на условия теоремы Дирака составляет $O(n)$, где n — число вершин графа, так как для нахождения минимальной степени вершин надо перебрать все n вершин.

Алгоритм проверки отношения смежности графа на плотность

Рассмотрим все ребра графа. Зафиксируем ребро (u,v). Если не существует вершины t, которая смежна с обоим вершинам u,v, то отношение смежности графа не является плотным. Вершину t достаточно перебрать среди всех вершин, смежных вершине u. Общая сложность функции равна $O(nm)$, где n — число вершин графа, m — число ребер графа. Это справедливо, так как в худшем случае каждая вершина имеет $n-1$ смежную, а число ребер равно m.

Алгоритм проверки графа на гамильтоновость

В данном разделе под m подразумевается представление некоторого множества в виде числа в двоичной системе счисления. Например, в графе из трех вершин множество, состоящее из вершин с номерами 0 и 2 вершин, имеет двоичное представление 101, так как 0 и 2 разряды равны 1. Нумерация разрядов от младшего к старшему идет справа налево, начиная с нуля. Под n подразумевается число вершин графа.

Предположим, что в графе есть гамильтонов цикл, заканчивающийся в вершине h (то есть в вершине с номером h). Положим значения таблицы динамического программирования $D_{i,j}=0$ для всех $i=\overline{0,n-1}$ и $j=\overline{0,2^n-1}$. Ячейки таблицы содержат численные значения: 1 или 0. Пусть $m_0=2^n-1$, то есть во множестве m_0 есть все вершины графа. Будем последовательно рассматривать все вершины, смежные с вершиной h и находящиеся во множестве m_0. Пусть одна из таких вершин — это вершина j. Тогда удалим из текущего множества m_0 вершину i и перейдем в вершину j рекурсивно. Для того, чтобы удалить вершину i из множества m_0, достаточно просто вычесть из его двоичного представления число 2^i. В этом рекурсивном переходе и состоит весь алгоритм. Рассмотренное множество m_0 является начальным, то есть функция запускается с таким значением $m_0=2^n-1$.

Рассмотрим рекурсивную функцию $F(m,i)$, где m — маска текущего множества вершин, а i — текущая вершина. Функция будет задаваться последовательностью следующих шагов:

1. Если $D_{i,m}\neq 0$ завершить выполнение функции.
2. Положить $D_{i,m}=1$.
3. Если $m=2^i$, то выполнить следующее: если вершина h смежна с вершиной i, то завершить выполнение функции с результатом: «найден гамильтонов цикл»; иначе завершить выполнение функции с результатом: «найден гамильтонов путь».
4. Рассмотрим номера всех вершин, смежных с вершиной i. Пусть одна из них — вершина с номером j. Тогда, если $m\&2^j\neq 0$ (где операция & подразумевает операцию побитового «И»), то выполнить следующее:
   4.1. Вызвать функцию $F(m-2^i,j)$.
   4.2. Если результат вызова — «найден гамильтонов цикл», завершить выполнение функции с результатом: «найден гамильтонов цикл».
   4.3. Если результат вызова — «найден гамильтонов путь», запомнить, что на этапе 4.1 был найден хотя бы 1 гамильтонов путь.
5. Если на этапе 4.1 был найден хотя бы один гамильтонов путь, завершить выполнение функции с результатом: «найден гамильтонов путь». Иначе завершить выполнение функции с результатом: «ничего не найдено».

Для того, чтобы узнать, имеет ли граф хотя бы один гамильтонов цикл или хотя бы один гамильтонов путь, необходимо запустить функцию $F(2^n-1,i)$ для всех $i=\overline{0,n-1}$. Если был найден хотя бы один гамильтонов цикл — значит граф гамильтонов. Если цикл не был найден, но найден хотя бы один гамильтонов путь — значит граф содержит гамильтонов путь. Если не был найден и гамильтонов путь — значит граф не содержит ни гамильтонова пути ни гамильтонова цикла. Общая сложность функции $O\left(\left(2^n\right)\cdot n^2\right)$. Это справедливо, так как заполняемая таблица для динамического программирования имеет размер $\left(2^n\right)\cdot n$, а в худшем случае каждая вершина имеет $n-1$ смежную.

Алгоритм проверки существования простого цикла заданной длины

Если добавить в проверку графа на гамильтоновость условие, при котором функция будет завершаться не когда в текущем множестве m осталась одна вершина, а $n-c$ вершин, где c — длина цикла, то получится именно функция проверки существования простого цикла заданной длины. Положим значения $D_{i,j}=0$ для всех $i=\overline{0,n-1}$ и $j=\overline{0,2^n-1}$. Рассмотрим рекурсивную функцию $F(i,m,l,c)$, где m— маска текущего множества вершин, i — текущая вершина, l — число пройденных вершин, c — длина искомого цикла. Функция будет задаваться последовательностью следующих шагов:

1. Если $D_{i,m} \neq 0$ завершить выполнение функции.
2. Положить $D_{i,m} = 1$.
3. Если $l = c$, то выполнить следующее: если вершина h смежна с вершиной i, то завершить выполнение функции с результатом: «найден цикл».
4. Рассмотрим номера всех вершин, смежных с вершиной i. Пусть одна из них — вершина с номером j. Тогда, если $m \& 2^j \neq 0$ (где операция & подразумевает операцию побитового «И»), то выполнить следующее:
   4.1. Вызвать функцию $F\left(j, m - 2^i, l + 1, c\right)$.
   4.2. Если результат вызова — «найден цикл», завершить выполнение функции с результатом: «найден цикл».
5. Завершить выполнение функции с результатом: «ничего не найдено».

Для того, чтобы узнать, имеет ли граф хотя бы один цикл длины c, необходимо запустить функцию $F\left(i, 2^n - 1, 1, c\right)$ для всех $i = \overline{0, n-1}$. Если был найден хотя бы один цикл — значит граф содержит искомый цикл.

Общая сложность функции $O\left(n^2\left(C_n^1 + C_n^2 + ... + C_n^c\right)\right)$, так как число возможных подмножеств множества вершин графа из 1 элемента C_n^1, число возможных подмножеств множества вершин графа из 2 элементов C_n^2, и так далее до числа возможных подмножеств множества вершин графа из c элементов C_n^c. Кроме того каждая вершина в худшем случае имеет $n - 1$ смежную, а сама функция запускается n раз из каждой вершины графа.

Это справедливо, так как алгоритм последовательно рассматривает все подмножества, пока их мощность не станет равна c. Как только мощность множества становится равной c алгоритм проверяет совпадение последней рассмотренной вершины со стартовой. В случае совпадения результат — в данном графе есть некоторый цикл длины c. Сам цикл в точности находить не требуется, достаточно установить его наличие.

Каталог графов

Ниже приведены изображения графов, удовлетворяющих рассмотренной гипотезе.

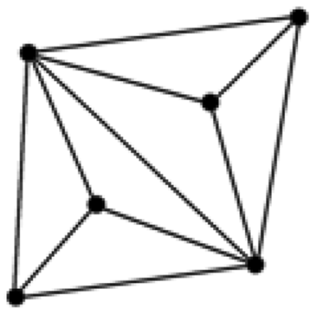

Рис. 1

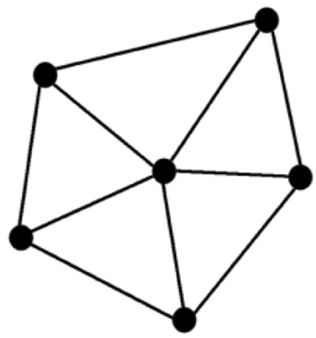

Рис. 2

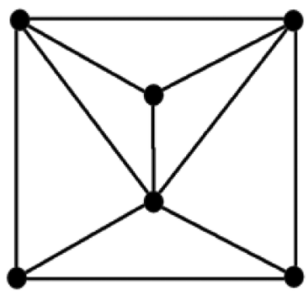

Рис. 3

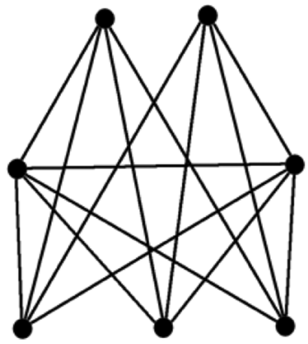

Рис. 4

ЛИТЕРАТУРА

1. Абросимов, М.Б. Практические задания по графам: учебное пособие / М.Б. Абросимов, А.А. Долгов. Сар.: Научная книга, 2009. 76 с.
2. Карпов Д.В. Теория графов. — Москва: МЦНМО, 2022. 560 с.
3. Н.К. Верещагин, А. Шень. Лекции по математической логике и теории алгоритмов. Часть 1. Начала теории множеств. 2-е изд., исправленное. М.:МЦНМО, 2002, 128 с.
4. Специальные свойства бинарных отношений [Электронный ресурс] // Математический форум Math Help Planet. URL: http://mathhelpplanet.com/static.php?p=spetsialnyye-svoystva-binarnykh-otnosheniy (дата обращения: 01.05.2020).
5. J.A Bondy, Pancyclic graphs I, Journal of Combinatorial Theory, Series B, Volume 11, Issue 1, 1971, Pages 80–84, ISSN 0095-8956, https://doi.org/10.1016/0095-8956(71)90016-5.
6. Pancyclic graphs and conjecture of Bondy and Chvatal. E.F. Schmeichel and S.L. Hakimi. Northwestern University, Evanston, Illinois 60201. Communicated by W.T. Tutte. Received December 28, 1973
7. Pancyclicity of Hamiltonian graphs. Nemanja Draganić, David Munhá Correia, Benny Sudakov.
8. Chvátal-Erdős condition for pancyclicity. Nemanja Draganić, David Munhá Correia, Benny Sudakov

Журнал «Современная наука: актуальные проблемы теории и практики»