Глава 1. Клики и их поиск в графе

Клика в графе – подмножество вершин графа, любые две из которых соединены ребром. Другими словами, клика – это полный подграф первоначального графа. Для нахождения всех возможных образованных в графе клик используются различные алгоритмы решения данной задачи.

1.1 Алгоритм Брона-Кербоша

Алгоритм Брона-Кербоша – алгоритмический метод для нахождения оптимального решения для поиска всех клик неориентированного графа. Был разработан математиками Броном и Кербошем в 1973 и до сих пор является одним из самых эффективных алгоритмов поиска клик.

В общей сложности алгоритм может применяться для двух различных целей: максимальной клики (основная цель) и поиск некой клики с четко заданным размером. Большинство модификаций и подвидов алгоритма связаны именно с решением задачи под первую цель.

Начиная с одиночной вершины, образующей полный подграф, алгоритм на каждом шаге пытается увеличить уже построенный полный подграф, добавляя в него вершины из множества кандидатов. Алгоритм использует тот факт, что всякая клика в графе является его максимальным по включению полным подграфом.

Высокая эффективность и скорость обеспечивается отсечением при переборе вариантов, которые заведомо не приведут к построению клики, для чего используется дополнительное множество, в которое помещаются вершины, которые уже были использованы для увеличения полного подграфа. Вычислительная сложность линейна относительно количества клик в графе. В худшем случае алгоритм работает за О(), где n – количество вершин в графе.

Хотя другие алгоритмы для решения данной задачи теоретически лучше работают на входах, которые имеют несколько максимальных независимых множеств, алгоритм Брона-Кербоша и последующие его усовершенствования на практике являются более эффективными, нежели другие альтернативы.

Данный алгоритм хорошо известен и широко используется в прикладных областях графовых алгоритмов (например, вычислительная химия).

1.1.1 Алгоритм Брона-Кербоша без поворота

Базовая форма алгоритма Брона-Кербоша представляет собой рекурсивный алгоритм поиска с [возвратом](https://en.wikipedia.org/wiki/Backtracking), который ищет все максимальные клики в данном графе G. В более общем смысле, для трех непересекающихся наборов вершин R, P и X он находит максимальные клики, включающие все вершины в R, некоторые вершины в P и ни одну из вершин в X. В каждом вызове алгоритма P и X являются непересекающимися множествами, объединение которых состоит из тех вершин, которые образуют клики при добавлении к R. Когда P и X оба пусты, к R нельзя добавить никаких дополнительных элементов, поэтому R является максимальной кликой, и алгоритм выводит именно R.

Базовая форма алгоритма неэффективна в случае графов с большим количеством немаксимальных клик: он делает рекурсивный вызов для каждой клики, максимальной или нет, что значительно увеличивает время работы алгоритма.

1.1.2 Алгоритм Брона-Кербоша с поворотом

Понимая, что алгоритм базовый алгоритм (без поворота) тратит много времени на рекурсивный вызов каждой клики в графе, Брон и Кербош ввели вариант алгоритма, включающий некую “осевую вершину” u, выбранную из P. Любая максимальная клика должна включать либо u, либо одного из ее несоседей, иначе клику можно было бы увеличить, добавив к ней u. Поэтому только u и ее несоседи должны быть проверены как варианты для вершины v, которая добавляется к R при каждом рекурсивном вызове алгоритма.

При дополнении базовой формы алгоритма Брона-Кербоша поворотом минимизируется количество рекурсивных вызовов, выполняемых алгоритмом, экономия времени выполнения по сравнению с версией алгоритма без поворота может быть значительной.

1.1.3 Алгоритм Брона-Кербоша с порядком вершин

Также существует альтернативный метод улучшения базовой формы алгоритма Брона-Кербоша. Под ним подразумевается отказ от поворота на самом внешнем уровне рекурсии, а также тщательный выбор порядка рекурсивных вызовов. Тщательный выбор порядка основан на порядке вырождения графа. Вырожденность графа G – наименьшее число d такое, что каждый подграф графа G имеет вершину степени d или меньше. Порядок вырождения – порядок вершин такой, что каждая вершина имеет d или меньше соседей, которые следуют позже по порядку. Если порядок вершин v, по которым проходит алгоритм Брона-Кербоша, является порядком вырождения, то множество P вершин-кандидатов в каждом вызове (соседи v, которые находятся позже в порядке) гарантированно будут иметь размер не более d.

Порядок вырождения находится за линейное время, многократно выбирая вершину минимальной степени среди оставшихся вершин. Данный метод достаточно эффективен для графов с малым вырождением и используется на практике в разных областях реального мира (например, социальные сети).

1.2 Алгоритм MaxCliqueDyn

Алгоритм MaxCliqueDyn – алгоритм поиска максимальной клики в неориентированном графе. Он основан на базовом алгоритме (алгоритме MaxClique), который находит максимальную клику ограниченного размера. Граница находится с использованием улучшенного алгоритма раскраски. MaxCliqueDyn расширяет алгоритм MaxClique, включая динамически изменяющиеся границы. Алгоритм был разработан Жанезом Концем в 2007 году.

По сравнению с предыдущими алгоритмами MaxCliqueDyn улучшен алгоритмом приближенной окраски (алгоритмом ColorSort) и применением более жестких и более затратных в вычислении верхних границ для пространства поиска. Однако оба улучшения сокращают время на поиск максимальной клики. В дополнение к сокращению времени улучшенный алгоритм раскраски также уменьшает количество шагов, необходимых для поиска максимальной клики.

Базовый алгоритм рекурсивно ищет максимальную клику, добавляя и удаляя вершины из Q – набора вершин растущей в данный момент клики.

Алгоритм окраски ColorSort раскрашивает вершины одну за другой в том же порядке, в котором они появляются в наборе вершин-кандидатов R, так что, если следующая вершина p не является смежной со всеми вершинами в некотором цветовом классе, она добавляется в этот класс; и если p смежна хотя бы с одной вершиной в каждом из существующих цветовых классов, она помещается в новый цветовой класс.

На каждом шаге алгоритма MaxClique алгоритм также пересчитывает степени вершин в P относительно вершины, в которой алгоритм находится в данный момент. Затем эти вершины сортируются в порядке убывания относительно их степеней в графе G. Затем алгоритм ColorSort рассматривает вершины в P, отсортированные по их степеням. Таким образом, количество шагов, необходимых для нахождения максимальной клики, сокращается до минимума.

Алгоритм имеет постоянные вычислительные затраты O(|P|2) на определение степеней и сортировку вершин