ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет Санкт-Петербургская школа
физико-математических и компьютерных наук

Епифанов Владислав Николаевич

ЗАДАЧА ПОИСКА МАКСИМАЛЬНОГО РАЗРЕЗА, ПАРАМЕТРИЗОВАННАЯ НАД НИЖНЕЙ ОЦЕНКОЙ, НА СПЕЦИАЛЬНЫХ КЛАССАХ ГРАФОВ

Выпускная квалификационная работа - БАКАЛАВРСКАЯ РАБОТА
по направлению подготовки *01.03.02 Прикладная математика и информатика*
образовательная программа «Прикладная математика и информатика»

Рецензент
м.н.с.
Д.Г. Сагунов

Руководитель
канд.физ.-мат.наук
И.А. Близнец

Санкт-Петербург 2021

Оглавление

Аннотация

MAX-CUT заключается в поиске разбиения вершин графа на два множества с максимальным количеством ребер между ними. Будучи NP-полной, эта задача вероятно не решается за полиномиальное время, потому в качестве фреймворка для эффективного решения некоторых экземпляров задачи используются параметризированные алгоритмы. В случае MAX-CUT одной из самых естественных параметризаций является параметризация над нижними оценками. Формально, задача MAX-CUT, параметризованная над нижней оценкой $\mathrm{LB}(G)$ заключается в следующем: по графу G и параметру k проверить, есть ли в графе разрез размера $\mathrm{LB}(G) + k$.

Известен алгоритм для MAX-CUT над оценкой $|V| - 1$ с асимптотикой $2^{\mathcal{O}(k)} \operatorname{poly}(|G|)$. Наша работа развивает эту параметризацию MAX-CUT. Мы приводим полиномиальный алгоритм препроцессинга, позволяющий уменьшить размер графа до $\mathcal{O}(k)$, работающий для произвольных графов, а также предлагаем алгоритмы с асимптотикой $2^{\mathcal{O}(\sqrt{k})} \operatorname{poly}(|G|)$ для хордальных и кодвудольных графов. Предложенный алгоритм оказывается асимптотически оптимальным для хордальных графов и их подкласса расщепляемых графов. Для трехдольных графов мы доказываем невозможность асимптотического улучшения алгоритма относительно общего случая. Мы также рассматриваем специфичную для трехдольных графов оценку в $\frac{2}{3}|E|$ и приводим полиномиальный алгоритм, достигающий этой оценки.

Ключевые слова: параметризованные алгоритмы, максимальный разрез, параметризация над нижней оценкой, графы

Max-Cut is a problem of partitioning vertices of a graph into two sets with the maximal number of edges between them. Despite this problem being NP-complete, hence probably impossible to solve in polynomial time, parameterized algorithms framework can be employed to solve some of its instances efficiently. The most natural way to parameterize Max-Cut is parameterization above lower bounds. Formally Max-Cut parameterized above a lower bound $\mathrm{LB}(G)$ is defined as follows: given a graph G and a parameter k, check whether the graph admits a cut of size $\mathrm{LB}(G) + k$.

There is a known algorithm for Max-Cut above lower bound of $|V| - 1$ with a running time of $2^{\mathcal{O}(k)} \operatorname{poly}(|G|)$ [1]. We develop this parameterization further by presenting a polynomial preprocessing algorithm that reduces graph size to $\mathcal{O}(k)$ and algorithms with running time of $2^{\sqrt{k}} \operatorname{poly}(|G|)$ for chordal and co-bipartite graphs. The algorithm for chordal graphs is asymptotically optimal even when restricted to split graphs. For tripartite graphs we prove asymptotical optimality of general case algorithm as well as provide a polynomial algorithm obtaining a tripartite-specific lower bound of $\frac{2}{3}|E|$.

Keywords: parameterized algorithms, max cut, parameterization above lower bound, graphs

Введение

Max-Cut, или задача поиска максимального разреза, является одной из первых известных NP-полных задач. Она имеет множество приложений от дизайна микросхем [2] до анализа социальных сетей [3].

Мы рассматриваем невзвешенную версию с бинарным ответом, а именно

Определение (Max-Cut)**.** *Дан граф G и число c. Существует ли разбиение вершин $X \sqcup Y = V(G)$ такое, что между X, Y проходит хотя бы c ребер.*

Поскольку задача NP-трудна, в предположении P $\neq$ NP для ее решения не существует алгоритма, работающего за полиномиальное время. Но это недостаточно точная оценка. Чтобы лучше понять минимальное время работы алгоритма прибегнем к *гипотезе экспоненциального времени* (exponential time hypothesis, ETH), сформулированной Импальяццо и Патури в 1999 году [4]. Эта гипотеза постулирует существование такого $\delta > 0$, что задача 3-SAT не может быть решена за $o(2^{\delta n})$, где $n =$ количеству переменных. Из этой гипотезы имеется множество следствий. Например доказано, что если она верна, 3-SAT нельзя решить за $o(2^{\delta' n})$ даже в случае когда каждая переменная используется константное число раз [5]. Это позволяет доказать, что Max-Cut не решается за $2^{o(|V|)}$ даже в случае когда $|E| = \mathcal{O}(|V|)$.

Однако несмотря на сложность задачи в общем случае, некоторые экземпляры могут быть решены быстрее. Для формализации этого подхода существуют параметризованные алгоритмы. У задачи выделяется параметр k и время работы алгоритма оценивается как функция не только от размера входных данных, но и от значения параметра. Если для параметризованной задачи существует алгоритм с временем работы вида $\mathcal{O}^*(f(k))$, эта задача лежит в классе FPT.

Кроме того, эффективному решению NP-трудных задач значительно помогает препроцессинг данных. Многие задачи могут быть существенно упрощены перед запуском основного алгоритма решения. Полиномиальный алгоритм препроцессинга почти всегда будет выполняться относительно быстро и может дать значительный прирост к скорости поиска точного ответа.

Эффективность препроцессинга оценивается по размеру получившейся задачи. Однако очевидно полиномиальный алгоритм не сможет уменьшать все экземпляры NP-трудной задачи. В связи с этим для оценки эффективности препроцессинга опять же используется параметризация.

Определение (Ядро)***.*** *Алгоритм A называется ядром параметризованной задачи L, если существует функция $f\colon \mathbb{N} \to \mathbb{N}$ такая, что $\forall (x,k)$ если $A(x,k) = (x',k')$, то $(x,k) \in L \Leftrightarrow (x',k') \in L$, $|x'|, k' \le f(k)$. Функция f называется размером ядра.*

Естественным параметром для Max-Cut был бы размер искомого разреза. К сожалению, на размер разреза существуют нижние оценки. Например известно что в любом графе есть разрез, содержащий хотя бы половину его ребер. Так как размер всего графа оказывается линеен от размера разреза, добавление такого параметра не позволяет лучше различать хорошо решаемые экземпляры задачи. В связи с этим рассматривается параметризация над нижней оценкой.

Определение (Параметризация над нижней оценкой)***.*** *Если* LB(G) *– нижняя оценка на размер разреза в графе G, то задача* Max-Cut *параметризованная над оценкой* LB(G) *заключается в следующем: По данным графу G и параметру k определить, есть ли в графе G разрез размера* $\mathrm{LB}(G) + k$*.*

Данная работа исследует сложность задачи на нескольких семействах графов, на которых она остается NP-полной [6]. Далее приведены их определения.

Определение (Хордальные графы)***.*** *Граф называется хордальным, если в нем нету индуцированных циклов длины > 3.*

Это одно из многих эквивалентных определений хордальных графов.

Определение (Расщепляемые графы)***.*** *Граф G называется расщепляемым (англ. split), если существует разбиение его вершин на два множества, что вершины первого образуют клику, а вершины второго – независимое множество. При этом между множествами могут быть проведены произвольные ребра.*

Нетрудно заметить, что все расщепляемые графы также являются хордальными.

Определение (Кодвудольные графы)***.*** *Граф называется кодвудольным, если его дополнение – двудольный граф.*

Определение (Трехдольные графы)***.*** *Граф называется трехдольным, если его вершины делятся на три независимых множества с ребрами между собой.*

Несмотря на то, что для двудольных графов задача Max-Cut является тривиальной, добавление третьей доли делает её NP-полной.

Цели и задачи

Основной целью данной работы была разработка более эффективных алгоритмов и меньших ядер для параметризаций Max-Cut над нижними оценками.

Для параметризации над $|V| - 1$ стояла задача улучшить алгоритм и найти меньшее ядро в случае когда семейство графов сужено до

- хордальных графов
- расщепляемых графов
- кодвудольных графов
- трехдольных графов

Также стояла задача изучить вычислительную сложность параметризации над $\frac{2}{3}|E|$ для трехдольных графов.

Основные результаты

Для параметризации над оценкой $|V| - 1$ мы находим алгоритм с асимптотикой $2^{\mathcal{O}(\sqrt{k})}$ для хордальных и расщепляемых графов, оказывающийся асимптотически оптимальным. Для кодвудольных также найден алгоритм с той же асимптотикой, но его оптимальность неясна. Для трехдольных графов оказывается невозможным найти алгоритм более эффективный чем алгоритм для общего случая

Также мы предоставляем ядро с $\mathcal{O}(k)$ вершин и ребер для произвольных графов, тем самым улучшая лучший текущий результат $\mathcal{O}(k^5)$ для всех рассматриваемых классов графов.

Относительно оценки $\frac{2}{3}|E|$ на трехдольных графах мы доказываем достижимость оценки за полиномиальное время, а также что сравнить размер максимального разреза с $\left(\frac{2}{3}+\varepsilon\right)|E|$ NP-трудно для сколь угодно малого ε.

Нотация

Все графы, рассматриваемые в данной работе являются простыми, если не указано обратное. Для графа G множество вершин обозначается $V(G)$, а множество ребер $E(G)$. В случае, когда граф однозначен, возможно написание V и E. Для $S \subseteq V(G)$ запись $G - S$ обозначает подграф G на вершинах $V(G) \setminus S$. Для $u, v \in V(G)$ ребро между этими вершинами обозначается uv. $N(v) = \{u \mid uv \in E\}$ – множество соседей вершины.

Из нестандартных обозначений используется $\mathrm{mc}(G)$ как максимальный разрез графа и $\langle a, b\rangle$ как скалярное произведение векторов.

Для указания асимптотики с точностью до полиномиального множителя используется $\mathcal{O}^*$. Формально, $f = \mathcal{O}^*(g)$ тогда и только тогда, когда $\exists c, k, N \in \mathbb{N}\colon \forall n > N\ f(n) \le c \cdot n^k \cdot g(n)$. Также полиномиальный от размера входных данных множитель обозначается за poly.

Структура работы

Обзор литературы описывает историю развития параметризаций Max-Cut над нижними оценками и известные результаты этой области.

Глава 1 описывает найденные алгоритмы для параметризации над $|V|-1$ для хордальных и кодвудольных графов, а также нижние оценки на теоретически возможное время работы подобных алгоритмов.

В главе 2 приведено описание найденного для параметризации над $|V|-1$ ядра линейного размера.

Последняя глава 3 посвящена результатам для трехдольных графов: невозможности улучшения алгоритмов для оценки $|V|-1$ и описанию достигнутых результатов по оценке $\frac{2}{3}|E|$.

Обзор литературы

NP-полнота взвешенного варианта Max-Cut была доказана в оригинальном списке 21 NP-полных задач Карпа [7]. Вскоре была доказана NP-полнота невзвешенной версии [8].

Бодлендер и Янсен рассматривают ограничения Max-Cut на различные классы графов и доказывают NP-полноту этих сужений [6]. Среди этих классов хордальные, расщепляемые, трехдольные и кодвудольные графы. Они также приводят алгоритм с асимптотикой $2^{\mathcal{O}(\mathrm{TW})}\,\mathrm{poly}$, где TW = древесная ширина графа (при условии данной древесной декомпозиции).

Параметризация над нижней оценкой была предложены в статье 1999 года, которая рассматривает оценку в $\frac{|E|}{2}$ [9]. Аналогичные параметризации многих других задач были предложены Махаджан и другими в статье 2009 года, в частности поднимающей вопрос о вычислительной сложности параметризации максимального разреза над оценкой Эдвардса-Эрдёша $\frac{|E|}{2}+\frac{|V|-1}{4}$, справедливой для связных графов [10]. Тремя годами позже Кроустон и другие доказывают что эта задача принадлежит классу FPT и приводят ядро с $\mathcal{O}(k^5)$ вершин [11]. Вскоре этот результат доказан для задачи Наибольший Сбалансированный Подграф – обобщения Max-Cut, вводящего ребра которые нужно не разрезать, а сохранять в одной доле [12]. Та же статья улучшает ядро до $\mathcal{O}(k^3)$ вершин.

Затем это ядро далее улучшено для конкретного класса графов. Этшайд и Мнич рассматривают (r, l)-графы – графы, разбивающиеся на r независимых множеств и l клик и строят ядро на $\mathcal{O}\left((r+l)k^2\right)$ вершин. Однако этот результат быстро устаревает с появлением ядра с линейным количеством вершин для общего случая [13].

Параметризация на другой нижней оценкой для связных графов – оценкой остовным деревом в $|V|-1$ была рассмотрена в более поздней работе, доказавшей что задача так же лежит в FPT и нашедшей ядро на $\mathcal{O}(k^5)$ вершин и ребер [1].

На трехдольных графах для Max-Cut нет результатов вычислительной сложности помимо её NP-трудности. Соответственно оценка $\frac{2}{3}|E|$ не была рассмотрена в отношении параметризации над нижними оценками.

1. Хордальные и кодвудольные графы

Мы приводим субэкспоненциальный алгоритм для Max-Cut над оценкой остовным деревом на кодвудольных и хордальных графах.

1.1. Кодвудольные графы

***Теорема* 1.1.** *На связном кодвудольном графе можно найти разрез размера $|V|-1+k$ или заключить что его нет за время $2^{\mathcal{O}(\sqrt{k})}\operatorname{poly}$.*

Доказательство. Кодвудольные графы состоят из двух клик, а потому они всегда плотные, то есть содержат много ребер. Из этого следует, что оценка $|V|-1$ не точна, поскольку ее превосходит другая оценка $\frac{|E|}{2}$.

Хотя бы одна из клик содержит половину вершин, а значит в ней $\geq \frac{\frac{|V|}{2}\left(\frac{|V|}{2}-1\right)}{2}$ ребер. Взяв гарантированный разрез в половину ребер одной этой клики, мы уже получаем $\geq \frac{|V|^2}{9}$ ребер.

Таким образом ответ на задачу будет положительным если $|V|-1+k \leq \left(\frac{|V|^2}{9}\right)$.

Иначе мы получаем $k \geq \left(\frac{|V|^2}{9}\right) - |V| + 1 = \mathcal{O}\left(|V|^2\right)$ и решение пребором за $\mathcal{O}^*(2^{|V|})$ укладывается в рамки $2^{\mathcal{O}(\sqrt{k})}\operatorname{poly}$. □

1.2. Хордальные графы

В этой секции нам понадобится пара разнородных относительно широко известных фактов.

***Лемма* 1.1.** *В хордальном графе всегда найдется минимальное древесное разложение, в котором каждая сумка – клика, причем это разложение можно найти полиномиальным алгоритмом [14].*

***Лемма* 1.2.** *По графу G и его древесной декомпозиции ширины k можно найти максимальный разрез за $2^{\mathcal{O}(k)}\operatorname{poly}$ [6].*

Прежде чем переходить к самому алгоритму, мы докажем следующую лемму:

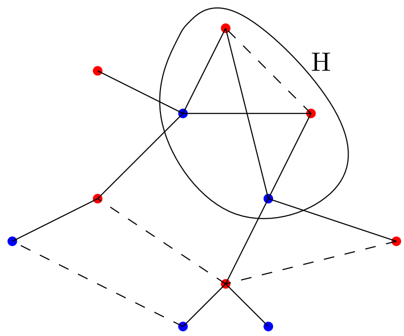

Рис. 1: Частичная окраска остовным деревом. Доли обозначены цветом; ребра, не вошедшие в разрез, проведены пунктиром

Лемма 1.3. *Пусть в связном графе G известен непустой подграф H и его разрез $A \sqcup B = V(H)$ на l ребер. Тогда в G можно построить разрез размера $\geq |V(G)| - |V(H)| + l$.*

Доказательство. Достроим разрез (A, B) до разреза всего графа. Рассмотрим лес произвольного обхода G с корнями в H. В каждом дереве возьмем вершины на четной глубине в ту же долю что и корень, а на нечетной – в другую (см. рис. 1).

Для каждой вершины вне H ребро в ее предка будет лежать в разрезе, следовательно таких ребер будет $V(G) - V(H)$. Добавляя l ребер из разреза H, получаем искомые $V(G) - V(H) + l$.

□

Теперь мы можем перейти к доказательству алгоритма.

Теорема 1.2. *На связном хордальном графе G можно найти разрез размера $V(G) - 1 + k$ за время $2^{\mathcal{O}(\sqrt{k})}$* poly *или определить что его нет.*

Доказательство. Согласно лемме 1.1 найдем древесную декомпозицию из клик и ее ширину l. Поскольку для любого графа есть разрез, содержащий не меньше половины его ребер, существует разрез клики размера l на $\frac{l(l-1)}{4}$ ребер.

Если $\frac{l(l-1)}{4} \geq k + l - 1$, то применив лемму к клике размера l и её разрезу мы получаем разрез G размера $\geq k + l - 1 + (|V(G)| - l) = |V(G)| - 1 + k$, давая положительный ответ на задачу.

В противном случае $\frac{l(l-1)}{4} < k + l - 1$. Оценим l при этих условиях.

$$\begin{aligned}
\frac{l(l-1)}{4} &< k + l - 1 \\
\frac{l(l-1)}{4} - l &< k - 1 \\
\frac{l(l-5)}{4} &< k - 1 \\
l(l-5) &< 4k - 4 \\
(l-5)^2 &< 4k - 4 \\
(l-5) &< \sqrt{4k-4} \\
l &< 5 + \sqrt{4k-4} \\
l &< 5 + 2\sqrt{k}
\end{aligned}$$

Значит $l = \mathcal{O}(\sqrt{k})$ и мы можем решить задачу точно за $2^{\mathcal{O}(l)}\,\mathrm{poly} = 2^{\mathcal{O}(\sqrt{k})}\,\mathrm{poly}$.

□

***Следствие* 1.3.1.** *На связном расщепляемом графе G можно найти разрез размера $V(G) - 1 + k$ за время $2^{\mathcal{O}(\sqrt{k})}\,\mathrm{poly}$ или заключить что его нет.*

Доказательство. Так как расщепляемые графы являются подмножеством хордальных, тот же алгоритм подходит и для них. □

1.3. Нижние оценки времени работы

В предположении гипотезы экспоненциального времени приведенный алгоритм оказывается асимптотически оптимальным для хордальных и расщепляемых графов

***Теорема* 1.3.** *Существование алгоритма для* Max-Cut *над $|V| - 1$ на расщепляемых графах с асимптотикой $2^{o(\sqrt{k})}\,\mathrm{poly}$ противоречит гипотезе экспоненциального времени.*

Доказательство. Известно сведение Max-Cut на произвольных графах к хордальным, превращающее граф G с n вершинами в граф G' с $\mathcal{O}(n^2)$ вершин и ребер [6], так что по максимальному разрезу G' можно понять максимальный разрез G. Так как параметр k не превышает количество ребер, алгоритм для с асимптотикой $2^{o(\sqrt{k})}$ poly позволит найти максимальный разрез в G' за время $2^{o(\sqrt{n^2})}$ poly $= 2^{o(n)}$ poly и таким образом понять максимальный разрез G. Но в предположении гипотезы экспоненциального времени, Max-Cut не решается за такую асимптотику. □

Для кодвудольных графов оценка получается хуже.

***Теорема* 1.4.** *В предположении гипотезы экспоненциального времени для* Max-Cut *над* $|V|-1$ *на кодвудольных графах не существует алгоритма с асимптотикой* $2^{o(\sqrt[4]{k})}$ poly.

Доказательство. Доказательство NP-полноты Max-Cut на кодвудольных графах описывает сведение задачи для расщепляемых графов к кодвудольным, превращающее граф с n вершинами в граф с $\mathcal{O}(n)$ вершин и $\mathcal{O}(n^2)$ ребер [6]. Поскольку алгоритм с асимптотикой $2^{o(\sqrt[4]{k})}$ poly решит задачу в получившемся графе за $2^{o(\sqrt[4]{n^2})}$ poly $= 2^{o(\sqrt{n})}$ poly, такой алгоритм невозможен по теореме 1.3. □

1.4. Результаты главы

В этой главе посредством леммы 1.3 был продемонстрирован способ достичь и превзойти оценку $|V|-1$ с помощью достраивания разреза подграфа до разреза полного графа. Эта лемма затем использована для конструирования алгоритма с асимптотикой $2^{\mathcal{O}(\sqrt{k})}$ poly для хордальных графов, который оказывается асимптотически оптимальным даже когда семейство ограничено до расщепляемых графов.

Также приведен алгоритм для кодвудольных графов с такой же асимптотикой $2^{\mathcal{O}(\sqrt{k})}$ poly. Однако его оптимальность уже непонятна так как оценка говорит только что асимптотика $2^{o(\sqrt[4]{k})}$ poly недостижима

2. Ядро для произвольных графов

2.1. Построение ядра

Попытки создания ядра для хордальных графов привели к работающему для произвольных графов ядру.

***Теорема* 2.1.** *Для задачи* Max-Cut *параметризованной над оценкой* $|V|-1$ *существует ядро с* $\mathcal{O}(k)$ *вершин и ребер.*

Доказательство. Мы добиваемся этого результата, построив набор редукционных правил, уменьшающих граф. Алгоритм заключается в последовательном применении первого подошедшего правила. Мы доказываем что в случае, когда ни одно правило не применимо, граф или оказывается достаточно маленьким, или в нем существует искомый разрез. В процессе применения правил могут образоваться кратные ребра. В итоге алгоритм избавится от них, получив на выходе простой граф.

Мы воспользуемся результатом из предыдущей работы по кернелизации максимального разреза над оценкой $|V| - 1$.

***Лемма* 2.1.** *В графе за полиномиальное время можно или найти разрез размера* $|V| - 1 + k$ *или найти* $S \subseteq V$, *что в* $G - S$ *каждая компонента двусвязности (далее называемая блок) является циклом или кликой и* $|S| \leq 3k$ *[1].*

Найдем такое S. Вершину $G-S$ будем называть *особой*, если она является точкой сочленения $G - S$, или если у нее есть ребро в S.

***Редукция* 2.1.1.** *Если в* $G - S$ *найдется блок* U *с единственной особой вершиной* v, *удалим* $U \setminus \{v\}$ *из графа, уменьшим* k *на* $\mathrm{mc}(U) - (|U| - 1)$, *где* $\mathrm{mc} =$ *максимальный разрез графа.*

Если v *не является точкой сочленения и имеет ребра только в одну вершину из* S, *удалим и ее, уменьшив* k *на количество убранных сверх одного ребер.*

Доказательство корректности. Поскольку любая компонента двусвязности $G - S$ является кликой или циклом, нахождение $\mathrm{mc}(U)$ тривиально. Эта редукция корректна, поскольку для любого разделения $G - (U \setminus \{v\})$ на доли, существует дополняющее его оптимальное разделение U. В оптимальном ответе среди ребер U в разрезе лежит не меньше и не больше $\mathrm{mc}(U)$, значит можно учесть это и убрать $U \setminus \{v\}$ из графа.

В случае когда v не точка сочленения и имеет ребро в одну вершину S, она остается висячей (возможно на кратном ребре) вершиной. Удаление такой вершины уменьшает разрез на количество ребер в вершину S, и уменьшает оценку на один т.к. удаляет одну вершину.

Заметим, что эта редукция всегда удаляет какой-то блок $G - S$. □

После применения правила 2.1.1 не останется компонент связности $G-S$, имеющих < 2 ребер в S. Это произойдет т.к. любой блок с одной точкой сочленения должен иметь свою особую вершину, имеющую ребро в S чтобы не быть удаленным по редукции 2.1.1. Если же в компоненте нет двух листовых блоков (блоков с одной точкой сочленения), то вся компонента связности связности будет одним блоком, причем с хотя бы двумя особыми вершинами. Раз у него есть две особые вершины, компонента имеет два ребра в S.

Мы отдельно оценим компоненты связности, имеющие 2 и ≥ 3 ребра в S. Начнем с последних.

Мы можем ограничить количество ребер ведущих из S в компоненты связности $G-S$ с ≥ 3 ребрами в S. Пусть между S и компонентами связности $G-S$, имеющими хотя бы 3 ребра в S проходит $f \geq 24k$ ребер. Тогда в графе существует большой разрез. Построим его.

Лемма 2.2. *Если в $G-S$ найдутся $f \geq 24k$ ребер между S и компонентами связности, имеющими ≥ 3 ребра в S, то в графе найдется разрез размера $|V(G) - 1 + k|$.*

Доказательство корректности. Рассмотрим компоненты связности $G - S$. Обозначим их за $U_1, \ldots, U_m$. Пусть из них l имеет хотя бы 3 ребра в S, а остальные – только два. Заметим, что $l \leq f/3$.

Построим разрез размера $|V(G)|+k$. В каждой компоненте U_i и построим разрез размера $\geq |V(U_i)|-1$ по оценке остовным деревом. Зафиксировав эти разрезы, можно взять все вершины S в одну долю или в другую, разрезав хотя бы половину ребер между S и $G - S$.

Между компонентами с ≥ 3 ребрами в S и S всего f ребер по условию, а из остальных в S выходит по 2 ребра, значит всего их $2(m-l)$. Так мы оценим, что проведенная операция увеличит разрез на $\geq \frac{f+2(m-l)}{2} = \frac{f}{2}+(m-l)$ ребер.

Итого максимальный разрез не меньше

$$\begin{aligned}
\mathrm{mc}(G) &\geq \sum_{i=1}^{m}(|V(U_i)|-1)+(m-l)+\frac{f}{2} \\
&= \sum_{i=1}^{m}|V(U_i)|-l+\frac{f}{2} \\
&\geq \sum_{i=1}^{m}|V(U_i)|-\frac{f}{3}+\frac{f}{2} \\
&= \sum_{i=1}^{m}|V(U_i)|+\frac{f}{6}
\end{aligned}$$

В то же время вершин в графе

$$V(G) = \sum_{i=1}^{m}|V(U_i)|+|S| \leq \sum_{i=1}^{m}|V(U_i)|+3k$$

Разница между разрезом и оценкой $|V|-1$ равняется $\frac{f}{6}-3k \geq \frac{24k}{6}-3k = k$. В графе нашелся разрез превышающий оценку на k, значит в случае $f \geq 24k$ ответ на задачу положительный.

□

Приведем еще несколько редукций чтобы ограничить размер некоторых компонент.

Редукция 2.2.1. *Пусть в $G-S$ найдется блок-клика U с особыми вершинами W такой, что $|W| \leq \lceil |U|/2 \rceil$. Удалим все его ребра и все его не особые вершины и уменьшим k на $\mathrm{mc}(U)-|U \setminus W|$.*

Если граф потерял связность, вернем минимальное необходимое число ребер между оставшимися вершинами W так, чтобы граф стал снова связен, увеличивая k на количество добавленных ребер.

Доказательство корректности. Рассмотрим G' – граф после удаления ребер U и вершин $U \setminus W$. Так как максимальный разрез клики достигается когда ее вершины поделены на доли поровну (в случае нечетной клики почти поровну), любой разрез G' достраивается до разреза G, содержащего $\mathrm{mc}(U)$ ребер U правильным выбором долей $U \setminus W$.

Значит $\mathrm{mc}(G') + \mathrm{mc}(U) = \mathrm{mc}(G)$. Максимальный разрез уменьшается на $\mathrm{mc}(U)$, количество вершин и оценка $|V| - 1$ уменьшаются на $|U \setminus W|$, значит k уменьшается на $\mathrm{mc}(U) - |U \setminus W|$.

Чтобы поддержать связность графа и достижимость оценки $|V| - 1$ мы добавляем несколько ребер в граф во второй части редукции. Поскольку добавленные ребра будут являться мостами, все они увеличивают максимальный разрез графа, тем самым увеличивая необходимое k.

Заметим, что таким образом было добавлено $\leq |W| - 1$ ребер. Значит если $|W| = a, |U \setminus W| = b$, то по итогу редукции k изменилось на $\leq -\mathrm{mc}(U) + b + (a-1) \leq -ab + b + (a-1) = -ab + a + b - 1 = -(a-1)(b-1) \leq 0$ то есть k не увеличилось. □

Редукция 2.2.2. *Пусть в $G - S$ есть ребро bc, причем $\deg_G b = \deg_G c = 2$. Пусть $N(b) = \{c, a\}$, $N(c) = \{b, d\}$ (возможно $a = d$). Тогда удалим вершины b, c и, если $a \neq d$, добавим ребро ad.*

Доказательство корректности. В максимальном разрезе из ребер ab, bc, cd будет все 3 если a, d в разных долях, и только 2 если в одной. В измененном графе разрез оказывается на 2 меньше в обоих случаях. Разрез на два меньше и удалены две вершины, так что параметр k не меняется.

Заметим, что каждая компонента двусвязности $G - S$ все еще является кликой. Для этого разберем два случая

1. $a, d \in S$: тогда удаление b, c полностью удаляет одну компоненту связности $G - S$ и не затрагивает другие.

2. НУО $a \in G - S$: тогда редукция выражается как стягивание ребер ab, bc. Стягивание ребра в $G - S$ или полностью уничтожает компоненту двусвязности (если это единственное ребро в ней), или уменьшает размер

клики, или уменьшает размер цикла. В любом случае остальные компоненты не затрагиваются и структура сохраняется.

□

Редукция 2.2.3. *Если среди блоков $G - S$ есть k четных циклов, в графе есть разрез размера $|V(G)| - 1 + k$.*

Доказательство корректности. Рассмотрим подграф H состоящий только из блоков, являющихся четными циклами. Выкинем из каждого цикла по одному ребру, получив некоторый лес.

Достроим этот лес до остовного дерева графа G. Получившееся дерево, как двудольный граф, имеет разрез $|V(G)| - 1$. Однако данный разрез разрезает еще и все выкинутые ребра H так как они были выкинуты из четного цикла, вошедшего в дерево. Значит в G разрез с таким разделением на доли имеет размер $\geq |V(G)| - 1 + k$. □

Чтобы оценить суммарный размер компонент $G - S$, имеющих ≥ 3 ребер в S, разделим их блоки на следующие типы

1. Блоки, имеющие ≥ 3 точки сочленения
2. Блоки, имеющие < 3 точек сочленения, но имеющие другую вершину с ребром в S.
3. Четные циклы с ровно двумя особыми вершинами, являющимися точками сочленения.
4. Остальные блоки с ровно двумя особыми вершинами, являющимися точками сочленения.

Это покрывает все варианты блоков так как блоки с одной особой вершиной удалены редукцией 2.1.1.

Заметим, что после редукций, в каждом оставшемся блоке $G - S$ хотя бы половина вершин – особые. Для клик это верно из-за редукции 2.2.1, а в цикле не бывает двух подряд идущих не особых вершин по итогу редукции

2.2.2. Так что достаточно ограничить сумму количеств особых вершин в этих блоках.

Оценим суммарное количество вершин в каждом отдельно

1. Рассмотрим двудольный лес F блоков и точек сочленения графа $G-S$, построенный следующим образом: для каждой компоненты двусвязности X создадим вершину a_X и каждой точки сочленения y создадим вершину b_y. Если $y \in X$, соединим их ребром. Все листовые и изолированные вершины в этом лесу будут соответствовать компонентам, причем имеющим свое ребро в S (остальные удалены редукциями 2.1.1). Таким образом в лесе оказывается $\leq 24k$ листьев согласно лемме 2.2.

   Заметим, что в лесе суммарная степень вершин степени ≥ 3 не превосходит утроенного количества листьев. Это верно т.к. средняя степень вершины $= \frac{2|E|}{|V|} \leq \frac{2(|V|-1)}{|V|} < 2$, значит на каждую вершину со степенью $2+t$ приходится t вершин со степенью 1.

   Значит суммарное количество точек сочленения в блоках первого типа $\leq 3 \cdot 24k = 72k$, а вершин с ребрами в S всего $24k$. Суммарно особых вершин $\leq 72k + 24k = 96k$, а всего вершин может быть в два раза больше, то есть $\leq 192k$.

2. В каждом блоке этого типа есть хотя бы одна вершина с ребром в S, не являющаяся точкой сочленения и, так как точек сочленения ≤ 2, такие вершины составляют хотя бы треть от всех особых вершин блоков этого типа. Значит особых вершин $\leq 3 \cdot 24 = 72k$, а всего вершин $\leq 144k$.

3. По редукции 2.2.3 таких $\leq k$, а по редукции 2.2.2 каждый из них размером ≤ 4

4. Посмотрим, какие вообще блоки могли оказаться в этой категории. Если это клика, то в ней должно быть всего 2 вершины или она будет удалена редукцией 2.2.1. Если же это цикл, то в нем ≤ 4 вершины из-за редукции 2.2.2. Поскольку цикл длины 3 является кликой, а четные циклы не относятся к этой группе, единственный вариант – клика K_2, то есть просто ребро.

Чтобы ограничить их количество опять прибегнем к лесу блоков и точек сочленения F. Пометим в нем вершины со степенью ≥ 3; вершины, соответствующие блокам из предыдущих пунктов; и точки сочленения с ребром в S. Как мы показали вершин каждого из этих типов $\mathcal{O}(k)$, а значит и всех их $\mathcal{O}(k)$.

Остались непомеченными блоки этого пункта и точки сочленения без ребра в S, лежащие в двух блоках. Заметим, что благодаря редукции 2.2.2 в F не бывает пути из непомеченных вершин длиной ≥ 6 (т.к. в таком пути чередуются точки сочленения и блоки, а редукция как раз запрещает такой путь длины 5, начинающийся с ребра).

Поскольку если сжать все непомеченные вершины в ребра граф останется лесом, в сжатом графе окажется $\mathcal{O}(k)$ вершин и следовательно ребер. Каждое сжатое ребро соответствует пути длины ≤ 5 из которых некоторые – блоки этого типа. Значит таких блоков тоже $\mathcal{O}(k)$.

Осталось оценить компоненты связности $G - S$, имеющие два ребра в S. Каждая такая компонента или является блоком с двумя вершинами с ребром в S, или цепочкой блоков, два крайних из которых имеют по ребру в S.

Так как каждый блок имеет две особые точки, это может быть только цикл длины 2 или 4, или просто ребро. Как мы уже оценили, четных циклов в $G-S$ всего $< k$. Рассмотрим $S' = S \cup$(четные циклы), тогда $|S'| \leq 3k+4k = 7k$ и оценим количество оставшихся ребер. По редукции 2.2.2 все оставшиеся пути имеют длину < 3. Путь длины 1 не содержит новых вершин, так что остаются пути длины 2, причем концы каждого такого пути – разные вершины S' (опять же по правилу 2.1.1).

Лемма 2.3. *Пусть таких путей оказалось $q > 16k$. Тогда в графе найдется разрез размера $|V(G)| + k$.*

Доказательство корректности. Построим такой разрез:

- Разделим S' на две доли так, что каждая вершина выберает долю равновероятно независимо от остальных.

- Достроим разрез в оставшемся графе согласно оценке остовным деревом как в лемме 2.1.

В компонентах $G - S$ с ≥ 3 ребрами в S получится разрез размера $\geq |V(G)| - |S'| - q$.

Заметим, что из-за случайного разделения S' для каждой из q вершин с двумя ребрами в S' есть шанс 0.5 что конца ребер окажутся в одной доле и получится взять два ребра в разрез, и шанс 0.5 что они окажутся в разных и в разрезе окажется только одно ребро. Значит матожидание ребер $1.5q \geq q + 8k$. Значит всего получается разрез $\geq |V(G)| - |S'| - q + q + 8k \geq |V(G)| + k$. $\square$

В каждом типе блоков $\mathcal{O}(k)$ вершин, значит всего вершин тоже $\mathcal{O}(k)$. Осталось ограничить количество ребер, но с данной параметризацией это не проблема. Если в графе $|E| > 2k + 2|V|$, гарантированный разрез $\frac{|E|}{2} \geq k + |V|$ окажется достаточным чтобы ответить на задачу положительно.

Также в графе могли остаться кратные ребра. Чтобы оставить граф простым, заменим каждое кратное ребро на путь длины 3 (эта операция корректна согласно редукции 2.2.2). Это опять же добавит линейное число вершин и ребер, оставив ядро линейным.

После применения всего вышесказанного алгоритм не увеличивает параметр k и или получает эквивалентный экземпляр задачи с графом размера $\mathcal{O}(k)$, или заключает что в графе точно есть искомый разрез, что позволяет вернуть в качестве эквивалентного тривиальный экземпляр с положительным ответом, например граф K_2 и параметр 0. Получившийся алгоритм является ядром линейного размера, что завершает доказательство теоремы. $\square$

Приведенное ядро является асимптотически оптимальным в предположении гипотезы экспоненциального времени.

Теорема 2.2. *В предположении гипотезы экспоненциального времени не существует ядра размера $o(k)$ для задачи поиска максимального разреза над оценкой $|V| - 1$ на произвольных графах.*

Доказательство. Так как Max-Cut решается тривиальным перебором за $\mathcal{O}^*(2^{V(G)})$, ядро размера $o(k)$ вершин позволило бы решать Max-Cut за время $\mathcal{O}^*(2^{o(k)}) = \mathcal{O}^*(2^{o(|E|)})$, что противоречит гипотезе экспоненциального времени. $\square$

2.2. Результаты главы

Эта глава была целиком посвящена улучшению ядра для задачи Max-Cut параметризованной над $|V| - 1$ для произвольных графов. Было получено ядро с $\mathcal{O}(k)$ вершин и ребер, что улучшает предыдущий результат в $\mathcal{O}(k^5)$ вершин и ребер от Мадафил и других [1], причем полученное ядро асимптотически оптимально.

3. Трехдольные графы

3.1. Параметризация над $|V| - 1$

Было выяснено, что ограничение семейства графов до трехдольных не позволяет существенно улучшить время работы алгоритма.

***Лемма* 3.1.** *Если существует алгоритм для поиска разреза над над оценкой $|V| - 1$ в трехдольных графах с временем работы $T(k, |G|)$, то в произвольном графе можно решать ту же задачу за время $T(k, 3|G|) + \mathcal{O}(G)$.*

Доказательство. Сведем задачу поиска максимального разреза на произвольных графах к трехдольным. Для этого по данному графу G построим граф G' следующим образом: Каждое ребро $uv \in E(G)$ заменим на путь длины 3 из ребер $ux_{uv}, x_{uv}y_{uv}, y_{uv}v$, где y_{uv}, y_{uv} – две новые вершины.

Докажем, что $\mathrm{mc}(G') = \mathrm{mc}(G) + 2|E(G)|$. Рассмотрим $\mathrm{mc}(G')$. Из каждой тройки ребер, соответствующих $uv \in E(G)$ будет лежать в разрезе или три, если доли u, v различаются, или два, если u, v лежат в одной доле разреза. Тогда оставив то же разделение вершин G на доли получится разрез G отличающийся на $2|E(G)|$. Аналогично по разрезу G строится разрез G' на $2|E(G)|$ больший.

Получается в G есть разрез размера $|V(G)| - 1 + k$ тогда и только тогда, когда в G' найдется разрез $|V(G)| - 1 + 2|E(G)| + k = |V(G')| - 1 + k$. Так как параметр не изменился и граф увеличился не более чем в 3 раза, алгоритм, решающий задачу в получившемся трехдольном графе за $T(k, |G'|)$, позволит решать задачу за $T(k, 3|G|)$ в произвольном случае.

□

В частности, в предположении гипотезы экспоненциального времени для трехдольных графов невозможен субэкспоненциальный алгоритм.

***Следствие* 3.1.1.** *В предположении гипотезы экспоненциального времени не существует алгоритма для поиска разреза над над оценкой $|V| - 1$ в трехдольных графах за асимптотику $2^{o(k)}\,\mathrm{poly}$.*

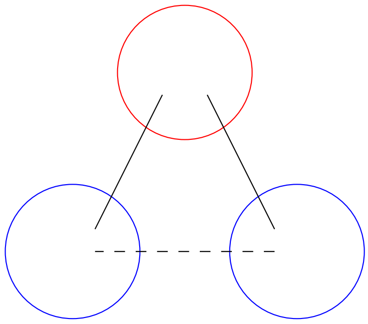

Рис. 2: оценка $\frac{2}{3}|E|$

Доказательство. По лемме 3.1 такой алгоритм позволил бы решать задачу на произвольном графе за $2^{o(k)}$ poly, что противоречие гипотезе экспоненциального времени. $\square$

Получается невозможно значительно улучшить асимптотику алгоритма для трехдольных графов относительно этой оценки.

3.2. Параметризация над $\frac{2}{3}|E|$

Мы рассматриваем другую оценку в $\frac{2}{3}|E|$. Доказательство этой оценки тривиально: в трехдольном графе можно взять разрез, содержащий ребра между любыми двумя парами долей из трех. Иллюстрация представлена на рис. 2.

Однако чтобы построить разрез таким образом необходимо узнать разбиение вершин графа на доли, что NP-трудно. Мы приводим полиномиальный алгоритм, достигающий этой оценки.

Данная оценка достигается с помощью полуопределенного программирования, которое позволяет решать задачи вида

$$\min_{v_1,\ldots,v_n\in\mathbb{R}^n} \sum_{1\leq i,j\leq n} c_{i,j}\langle v_i, v_j\rangle$$

$$\forall k \sum_{1\leq i,j\leq n} a_{k,i,j}\langle v_i, v_j\rangle \leq b_k$$

$$\text{где } \langle x, y\rangle = \text{ скалярное произведение}$$

с точностью $\pm\varepsilon$ за время $\mathrm{poly}\,(n, \log(1/\varepsilon))$.

Теорема 3.1. *Существует рандомизированный полиномиальный алгоритм, находящий разрез размера $\frac{2}{3}|E|$ на трехдольных графах.*

Доказательство. Чтобы для данного трехдольного графа G найти разрез построим задачу полуопределенного программирования. Каждой вершине графа будет соответствовать вектор v_i.

$$\min \lambda$$
$$\forall i \in V(G)\ \langle v_i, v_i\rangle = 1$$
$$\forall ij \in E(G)\ \langle v_i, v_j\rangle \leq \lambda$$

Поскольку все вектора единичные, скалярное произведение равняется косинусу угла между векторами. Эти условия минимизируют максимальный косинус, тем самым максимизируя минимальный угол между векторами вершин, соединенных ребром.

Поскольку граф трехдольный, существует решение с $\lambda = -\frac{1}{2}$: в нем выбраны три равноудаленных вектора в одной плоскости и вершинам каждой доли сопоставлен один из них.

Значит в оптимальном решении $\lambda_{OPT} \leq -\frac{1}{2}$, а углы между векторами соседних вершин $\geq \frac{2\pi}{3}$. Найдем это оптимальное решение с достаточно большой точностью.

Равновероятно выберем гиперплоскость проходящую через 0 и поделим вершины на две доли соответственно разделению их векторов гиперплоскостью. Для каждого ребра $ij \in E(G)$ вероятность того, что оно войдет в разрез равняется $p_{ij} = \frac{\angle v_i v_j}{\pi} \geq \frac{2}{3}$, значит матожидание размера разреза $E_{cut} = \sum_{ij\in E(G)} p_{ij} \geq \frac{2}{3}|E(G)|$

Размер разреза принимает целые значения не превышающие $|E(G)|$. Значит любой разрез, меньший $\frac{2}{3}|E(G)|$, отличается от оценки хотя бы на $\frac{1}{3}$, а превосходящий оценку разрез отличается на $\leq \frac{|E(G)|}{3}$. Тогда вероятность встреть разрез $\geq \frac{2}{3}|E|$ оказывается $\geq \frac{1/3}{|E(G)|/3} = \frac{1}{|E(G)|}$ и за $\mathcal{O}(|E(G)|)$ повторений она превращается в $\geq \frac{1}{2}$, давая полиномиальный рандомизированный алгоритм. $\square$

Эта задача полуопределенного программирования была впервые использована как часть алгоритма раскраски трехдольных графов в $\mathcal{O}(n^{0.631})$ цветов [15].

Также мы доказываем невозможность нахождения разреза $\left(\frac{2}{3}+\varepsilon\right)|E|$ на трехдольных графах за полиномиальное время в предположении $\mathrm{P} \neq \mathrm{NP}$.

Лемма 3.2. *В предположении $P \neq NP$ не существует полиномиального алгоритма, который для любого трехдольного графа G находит разрез размера $\geq \frac{5}{6}|E|$ (при его наличии).*

Доказательство. Рассмотрим задачу NAE-3-SAT+. Это версия SAT, в которой каждый клоз состоит из трех переменных без отрицания и клоз считается выполненным только если в нем есть и истинная переменная и ложная. Эта задача NP-трудна, а потому не решается за полиномиальное время в наших предположениях [16].

Сведем её к поиску разреза. Для этого по формуле построим граф следующим образом:

- для каждой переменной x создадим вершину a_x
- для клоза S на переменных x, y, z создадим три вершины $c_{S,x}, c_{S,y}, c_{S,z}$ и добавим ребра между ними. Дополнительно, добавим ребра между вершинами переменных, участвующими в клозе, и соответствующими им вершинами клоза $a_x c_{S,x}, a_y c_{S,y}, a_z c_{S,z}$

Пример такого построения показан на рис. 4.

Найдем максимальный разрез в таком графе, если доли вершин a_x зафиксированы. Заметим, что для клоза, переменные которого лежат в одной доле, можно взять в разрез только 4 ребра. Зато если среди переменных клоза одна лежит в одной доле и две других в другой, можно построить разрез из 5 ребер, как на рис. 3.

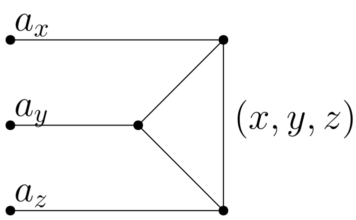

Рис. 3: гаджет для клоза

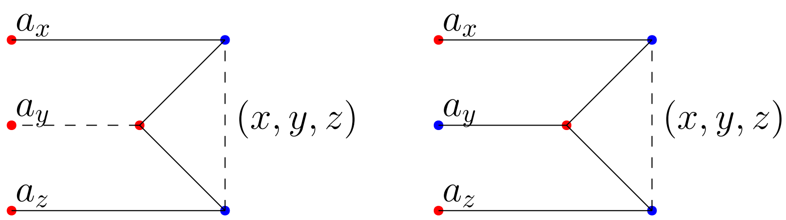

Рис. 4: Максимальный разрез гаджета

Заметим, что если сопоставить означению переменных разбиение соответствующих им вершин на две доли, получится что удовлетворяющее все клозы означивание переходит в удовлетворяющее по 5 ребер из каждого гаджета разбиение и наоборот. То есть разрез на $\frac{5}{6}|E|$ ребер существует тогда и только тогда, когда формула NAE-3-SAT выполнима.

Заметим, что построенный граф является трехдольным если формула была выполнимой. В этом случае можно опять же взять разбиение вершин a_x на две соответствующие выполняющему означению доли и достроить разбиение каждого гаджета как на рис. 5.

Значит чтобы понять, выполнима ли формула, достаточно запустить алгоритм поиска разреза на соответствующем ей графе. Если разрез размера $\frac{5}{6}|E|$ нашелся, то формула выполнима, иначе не выполнима.

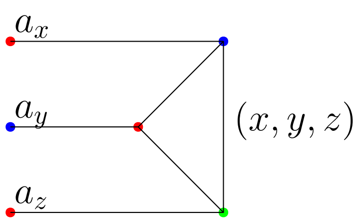

Рис. 5: Разбиение гаджета на доли

Здесь стоит отметить, что в случае невыполнимой формулы, мы запускаем алгоритм, предназначенный для трехдольных графов, не на трехдольном графе. В таком случае его результат может быть произвольным, но в любом случае он не найдет разрез размера $\frac{5}{6}|E|$ в графе, где такого разреза не существует. $\square$

Следствие 3.2.1. *В предположении $P \neq NP$ для любого $0 < \varepsilon < \frac{1}{6}$ не существует полиномиального алгоритма, который для любого трехдольного графа G находит разрез размера $\geq \left(\frac{2}{3} + \varepsilon\right)|E|$ (при его наличии).*

Доказательство. Докажем, что такой алгоритм позволит нам научиться определять наличие разреза $\frac{5}{6}|E|$ в трехдольных графах, что невозможно по предыдущей теореме.

Пусть в трехдольном графе G всего n вершин и m ребер. Надо проверить, есть ли в нем разрез размера $\geq \frac{5m}{6}$.

Добавим в граф k треугольников на новых вершинах. В получившимся графе надо проверить существование разреза размера $\geq \frac{5m}{6} + 2k$ т.к. в каждом треугольнике можно добиться разреза двух ребер.

Ребер же в получившемся графе $m+3k$. То есть отношение искомого разреза к количеству ребер стало $\frac{5m/6+2k}{m+3k}$. При увеличении k это число стремится к $\frac{2}{3}$. Рассмотрим максимальное k, при котором это отношение не меньше $\frac{2}{3} + \varepsilon$. Заметим, что при фиксированном ε такое $k = \mathcal{O}(m)$. Тогда т.к. было выбрано максимальное k, отношение $k+1$ уже меньше границы.

$$\frac{5m/6 + 2(k+1)}{m + 3(k+1)} < \frac{2}{3} + \varepsilon \tag{1}$$

Докажем что в графе с k добавленными треугольниками любой не максимальный разрез будет уже меньше $\frac{2}{3} + \varepsilon$ от ребер. Разрез не максимальный, значит его размер $\leq \frac{5m}{6} + 2k - 1$. Мы хотим доказать следующее:

$$\frac{5m/6 + 2k - 1}{m + 3k} < \frac{2}{3} + \varepsilon$$

Для этого покажем, что это отношение не превосходит отношения в формуле (1).

$$
\begin{aligned}
&\frac{5m/6+2k-1}{m+3k} < \frac{5m/6+2(k+1)}{m+3(k+1)}\\
&0 < \frac{5m/6+2k+2}{m+3k+3} - \frac{5m/6+2k-1}{m+3k}\\
&0 < \frac{(5m/6+2k+2)(m+3k)-(5m/6+2k-1)(m+3k+3)}{(m+3k+3)(m+3k)}\\
&0 < \frac{(5m/6+2k)(m+3k)+2(m+3k)-(5m/6+2k)(m+3k+3)+(m+3k+3)}{(m+3k+3)(m+3k)}\\
&0 < \frac{(5m/6+2k)(m+3k)+2(m+3k)-(5m/6+2k)(m+3k)-3(5m/6+2k)+(m+3k+3)}{(m+3k+3)(m+3k)}\\
&0 < \frac{2(m+3k)-3(5m/6+2k)+(m+3k+3)}{(m+3k+3)(m+3k)}\\
&0 < \frac{m(2-5/2+1)+k(6-6+3)+3}{(m+3k+3)(m+3k)}\\
&0 < \frac{m/2+3k+3}{(m+3k+3)(m+3k)}
\end{aligned}
$$

То есть в G есть разрез $\frac{5m}{6}+2k$ тогда и только тогда, когда в G' есть разрез $\left(\frac{2}{3}+\varepsilon\right)|E(G')|$, что может проверить алгоритм данный в условии.

□

3.3. Результаты главы

В этой главе приводится доказательство того, что невозможно получить алгоритм поиска разреза над оценкой $|V|-1$ для трехдольных графов с временем работы значительно лучше алгоритма для общего случая. Важным следствием из этого факта является отсутствие субэкспоненциальных алгоритмов в предположении гипотезы экспоненциального времени.

Также предлагается альтернативная нижняя оценка на разрез для трехдольных графов $\frac{2}{3}|E|$. Мы приводим полиномиальный алгоритм, достигающей этой оценки, а также доказываем что превзойти оценку на $\varepsilon|E|$ оказывается NP-трудно, а значит невозможно за полиномиальное время в предположении $\mathrm{P} \neq \mathrm{NP}$. Однако мы не приводим ни FPT алгоритма для параметризации над этой оценкой, ни доказательства его отсутствия.

Заключение

В работе были представлены алгоритмы с асимптотикой $2^{\mathcal{O}(\sqrt{k})}\text{poly}$ для задачи поиска максимального разреза, параметризованной над оценкой $|V|-1$ для кодвудольных и хордальных графов. Для последних приведенный алгоритм является асимптотически оптимальным даже в случае дальнейшего ограничения семейства графов до расщепляемых графов.

Со стороны препроцессинга для общего случая было предоставлено ядро линейного размера и доказано что в предположении гипотезы экспоненциального времени не существует сублинейных ядер. Однако в случае хордальных и кодвудольных графов вопрос об оптимальном ядре остается открытым.

Для трехдольных графов показано, что оценка $|V|-1$ не позволяет получить результаты асимптотически лучше чем в общем случае, а потому предложена другая оценка $\frac{2}{3}|E|$. Мы показываем, что данная оценка достижима за полиномиальное время, а также что превзойти ее на $\varepsilon|E|$ оказывается NP-трудно. Однако принадлежность задачи классу FPT остается открытым вопросом.

Список литературы

[1] Madathil Jayakrishnan, Saurabh Saket, Zehavi Meirav. Max-Cut Above Spanning Tree is Fixed-Parameter Tractable // Computer Science – Theory and Applications / Ed. by Fedor V. Fomin, Vladimir V. Podolskii. — Cham : Springer International Publishing, 2018. — P. 244–256.

[2] Barahona Francisco, Reinelt G. An Application of combinatorial optimization to statistical physics and circuit layout design // Operat. Research. — 1988. — 01. — Vol. 36. — P. 493 – 513.

[3] Agrawal Rakesh, Rajagopalan Sridhar, Srikant Ramakrishnan, Xu Yirong. Mining Newsgroups Using Networks Arising From Social Behavior. — 2003. — 07.

[4] Impagliazzo R., Paturi R. Complexity of k-SAT // Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317). — 1999. — P. 237–240.

[5] Impagliazzo Russell, Paturi Ramamohan, Zane Francis. Which Problems Have Strongly Exponential Complexity? // Journal of Computer and System Sciences. — 2001. — Vol. 63, no. 4. — P. 512–530. — URL: https://www.sciencedirect.com/science/article/pii/S002200000191774X.

[6] Bodlaender Hans L., Jansen Klaus. On the complexity of the maximum cut problem // STACS 94 / Ed. by Patrice Enjalbert, Ernst W. Mayr, Klaus W. Wagner. — Berlin, Heidelberg : Springer Berlin Heidelberg, 1994. — P. 769–780.

[7] Karp Richard. Reducibility Among Combinatorial Problems // Complexity of Computer Computations. — Vol. 40. — 1972. — 01. — P. 85–103.

[8] Garey M. R., Johnson D. S., Stockmeyer L. Some Simplified NP-Complete Problems // Proceedings of the Sixth Annual ACM Symposium on Theory of Computing. — STOC ’74. — New York, NY, USA : Association for

Computing Machinery, 1974. — P. 47–63. — URL: https://doi.org/10.1145/800119.803884.

[9] Mahajan Meena, Raman Venkatesh. Parameterizing above Guaranteed Values: MaxSat and MaxCut // Journal of Algorithms. — 1999. — Vol. 31, no. 2. — P. 335–354. — URL: https://www.sciencedirect.com/science/article/pii/S0196677498909968.

[10] Mahajan Meena, Raman Venkatesh, Sikdar Somnath. Parameterizing above or below guaranteed values // Journal of Computer and System Sciences. — 2009. — Vol. 75, no. 2. — P. 137–153. — URL: https://www.sciencedirect.com/science/article/pii/S0022000008000743.

[11] Crowston Robert, Jones Mark, Mnich Matthias. Max-Cut Parameterized above the Edwards-Erdős Bound // Automata, Languages, and Programming / Ed. by Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, Roger Wattenhofer. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. — P. 242–253.

[12] Crowston Robert, Gutin Gregory, Jones Mark, Muciaccia Gabriele. Maximum Balanced Subgraph Problem Parameterized above Lower Bound // Computing and Combinatorics / Ed. by Ding-Zhu Du, Guochuan Zhang. — Berlin, Heidelberg, 2013. — P. 434–445.

[13] Etscheid Michael, Mnich Matthias. Linear Kernels and Linear-Time Algorithms for Finding Large Cuts // Algorithmica. — 2018. — Vol. 80. — P. 2574–2615.

[14] Gavril Fănică. The intersection graphs of subtrees in trees are exactly the chordal graphs // Journal of Combinatorial Theory, Series B. — 1974. — Vol. 16, no. 1. — P. 47–56. — URL: https://www.sciencedirect.com/science/article/pii/009589567490094X.

[15] Karger David, Motwani Rajeev, Sudan Madhu. Approximate Graph Coloring by Semidefinite Programming // J. ACM. — 1998. — mar. — Vol. 45, no. 2. — P. 246–265. — URL: https://doi.org/10.1145/274787.274791.

[16] Schaefer Thomas J. The Complexity of Satisfiability Problems // Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. — STOC '78. — New York, NY, USA : Association for Computing Machinery, 1978. — P. 216–226. — URL: https://doi.org/10.1145/800133.804350.