Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАЗРАБОТКА СТРИМИНГОВОГО АЛГОРИТМА ДЛЯ ДЕКОМПОЗИЦИИ ГРАФОВЫХ МЕТРИК В МЕТРИКИ ДЕРЕВЬЕВ

Автор: Фафурин Олег Геннадьевич _____

Направление подготовки: 01.03.02 Прикладная математика и информатика

Квалификация: Бакалавр

Руководитель ВКР: Аксенов В.Е., PhD _____

Санкт-Петербург, 2021 г.

Обучающийся Фафурин Олег Геннадьевич
Группа М3439 Факультет ИТиП

Направленность (профиль), специализация
Математические модели и алгоритмы в разработке программного обеспечения

Консультанты:

a) Kapralov M., PhD, assistant professor _____

ВКР принята «____» _____ 20___ г.

Оригинальность ВКР _____%

ВКР выполнена с оценкой _____

Дата защиты «15» июня 2021 г.

Секретарь ГЭК Павлова О.Н. _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Изучение метрических пространств имеет большое прикладное значение — многие вычислительные задачи на графах и строках исследуют те или иные характеристики, связанные с расстояниями между вершинами. В современных реалиях для решения подобных задач порой требуются приближенные алгоритмы, способные приближать метрики с некоторой точностью. Такие задачи возникают в случае обработки больших объёмов данных, где входные данные слишком велики для классических алгоритмов — точные алгоритмы могут быть в таком случае неэффективны с точки зрения расходования ресурсов (времени, памяти). Стриминговые же алгоритмы работают с малым количеством памяти, чем позволяют еще больше оптимизировать её расход.

Целью работы является исследование задачи приближения произвольной метрики над конечным множеством с помощью вероятностного распределения метрик деревьев. Указанные деревья, приближающие исходную метрику, содержат исходное множество вершин графа — таким образом, можно установить инъективные отображения множества вершин исходного графа в множество вершин каждого из возможных приближающих деревьев. Такое исследование алгоритмов, работающих в классической модели вычислений, по большей части завершено — известна точная оценка на математическое ожидание относительного искажения веса кратчайшего пути. Иными словами, существует алгоритм, описанный в [6], приближающий произвольную метрику с заданной точностью, однако существует и семейство метрик, про которые в [4] доказано, что их нельзя приблизить лучше, чем с этой точностью. Стриминговая модель вычислений накладывает дополнительные ограничения на память, доступную для работы алгоритма. В этой модели пока неизвестна точная оценка на математическое ожидание искажения метрики. Сближение верхней и нижней границ такой оценки представляет научный интерес и является основной целью данной работы. Результаты исследования представляют также практическую значимость — более точное приближение метрик алгоритмом в стриминговой модели может позволить более эффективно работать графовым алгоритмам для решения многих задач, таких как «metric labeling problem», «minimum cost communication network problem», «metrical task system».

В первой главе настоящей работы изложен краткий обзор исследуемой области и прогресса в исследовании поставленной задачи, решаемой в классической модели вычислений, а также обозначена терминология и конвенции, характерные для описания изучаемой области в целом. Кроме того, в первой главе разъяснены цели работы с учётом введённых определений, а также изложен наивный алгоритм решения поставленной задачи, определён критерий качества ожидаемого решения.

Вторая глава содержит анализ алгоритма, улучшающего требуемые целевые характеристики по сравнению с исходной асимптотикой математического ожидания относительного изменения веса пути $O(\log^2 n)$ — он является результатом оптимизации наивного алгоритма. Также в этой главе описан анализ двух частей алгоритма, мотивирующий и поясняющий вносимые в алгоритм изменения. Кроме того, представлены примеры, позволяющие судить о точности верхней оценки исследуемой величины в разработанном алгоритме — математического ожидания относительного изменения длины кратчайшего пути между двумя вершинами, или, в терминах метрики, относительного изменения расстояния между точками. Наконец, в конце главы обсуждается реализация описанного алгоритма.

В третьей главе отмечены наблюдения о свойствах работы алгоритма из статьи [6], являющегося одним из двух основных звеньев описываемого нами алгоритма для решения задачи в стриминговой модели. Эти наблюдения произведены с использованием и на основе описанного во второй главе разбора анализа алогритма из статьи [6].

ГЛАВА 1. ПРИБЛИЖЕНИЕ КОНЕЧНЫХ МЕТРИК

В этой главе описана задача приближения конечных метрик, рассказано об основных работах, повлиявших на текущее представление научного сообщества о её решении, введены общепринятые обозначения и термины для описания задачи, поставлена цель работы. Также объяснено решение поставленной задачи, называемое нами наивным, улучшение которого представлено далее в работе.

1.1. Постановка задачи

Приближение конечных метрик более простыми — объект обширного изучения со стороны научного сообщества на протяжении нескольких последних десятилетий. Эта тема имеет непосредственную связь с эффективностью работы некоторых алгоритмов, в том числе алгоритмов на графах, оперирующих метрикой кратчайшего расстояния между вершинами. Понятия конечной метрики и произвольного конечного взвешенного графа эквивалентны в исследуемой нами области — чтобы получить из графа метрику, достаточно добавить отсутствующие ребра с весами, равными весам кратчайших путей в исходном графе. Чтобы получить из метрики эквивалентный ей граф, просто проведём рёбра между всевозможными парами точек данного пространства с весами, равными расстояниям между ними в данной метрике. В дальнейшем мы будем говорить о метрике и о графе как об одном и том же. В основном мы будем рассматривать невзвешенные графы и метрики кратчаших расстояний в них — в дальнейшем изложении по умолчанию рассматриваются именно такие графы, если явно не указано иное. Для исследования свойств рассматриваемых алгоритмов достаточно изучать компоненты связности графа по отдельности, поэтому считаем, что рассматриваемый граф связен.

Метрикой дерева называется конечная метрика на графе $G = (V_1,E_1)$, для которой существует такое дерево $T = (V_2, E_2), V_1 \subset V_2$, что $\forall u, v\ d_G(u,v) = d_T(u,v)$, где d_G и d_T — метрики кратчайших расстояний в графах G и T соответственно. Не все метрики являются метриками деревьев — несложно понять это, к примеру, про метрику цикла на 4 вершинах с рёбрами единичного веса. Однако идея *приблизить* произвольную метрику именно с помощью метрики дерева разумна: деревья имеют простую структуру, что позволяет многим алгоритмам, в том числе оперирующим с расстояними между вершинами, работать эффективнее. Принято фиксировать следующее усло-

вие на приближающую метрику: для любых двух точек исходного графа новое, приближающее расстояние должно быть не меньше исходного — в таком случае говорят, что новая, приближающая метрика *доминирует* над исходной. Мерой же оценки точности приближения является относительное изменение веса крайтчайшего пути — оно называется **искажением** *(англ. distortion)* ребра. Как следует из изложенного, искажение не меньше единицы, потому что все полученные приближающие расстояния между парами вершин не меньше исходных. **Искажением метрики** на графе будем называть наибольшее из искажений его рёбер. Оценка искажения метрики сверху, и представляет непосредственный интерес. В отдельных случаях может также представлять интерес, каково искажение рёбер, обладающих теми или иными свойствами.

1.2. Связанные работы

Одна из самых известных и простых конструкций построения графа (не обязательно дерева), приближающего метрику кратчайших расстояний данного графа с точностью $O(t)$, где параметр t не зависит от n, представлена в [2]. Она состоит в следующем: рассмотрим множество вершин исходного графа V, пустое множество рёбер на нём и будем последовательно добавлять в этот граф рёбра исходного графа. Если при попытке добавления ребра (u,v) оказывается, что $d_T(u,v) \leqslant t \cdot d(u,v)$, то ребро не добавляется. Этот подход можно мотивировать следующим образом: если между вершинами u и v к моменту вставки ребра (u,v) уже и так существует путь длины не более t, то ребро (u,v) можно не вставлять, потому что его отсутствие приведёт к искажению пути от u до v, а значит, и других путей, не более чем в t раз. Также в [2] показано, что в полученном приближающем графе останется $O(n^{1+\frac{1}{t}})$ рёбер.

Как показано в [8], некоторые метрики, например, цикл с рёбрами единичного веса, нельзя приблизить метрикой дерева с точностью лучше чем $\Omega(n)$ (здесь и далее n — количество вершин графа). В качестве способа устранить эту проблему в [8] было предложено рассматривать приближение метрики не одной метрикой дерева, а *вероятностным распределением* таковых. В работе [3] были формально определены вероятностные приближения метрик. В таком случае целевой характеристикой исследования становится не искажение веса пути графа, а математическое ожидание этого искажения. В дальнейшем в [4] показано, что для некоторых графов приближение возможно с точностью не лучше чем $\Omega(\log n)$. Наконец, в [6] представлен алгоритм (да-

лее **FRT**), позволяющий приблизить произвольную конечную метрику вероятностным распределением метрик деревьев с искажением не более $O(\log n)$, таким образом, была найдена точная оценка на асимптотику искажения вероятностного приближения метрик.

Стриминговая модель вычислений подразумевает, что входные данные представлены потоком, доступным на чтение, а количество памяти, доступной алгоритму в процессе работы, ограничено и недостаточно даже для хранения входных данных целиком. Эта модель ориентирована на ситуации, в которых данных, поступающих на обработку алгоритмом, настолько много, что выделение памяти и хранение их становится нежелательным. В случае задач, связанных с графами, принято считать, что количество памяти, доступной для алгоритма, ограничено сверху $O(n \log^{\alpha} n)$, где $\alpha > 0$ — некоторая константа.

1.3. Цели

Мы исследуем задачу приближения метрики графа вероятностным распределением метрик деревьев в стриминговой модели вычислений. Основной задачей оптимизации будет минимизировать искажение исходной метрики. При этом количество памяти, доступной алгоритму, не должно быть больше, чем $O(n \log^{\alpha} n), \alpha = const$.

1.4. Наивное решение

На тему этой задачи нет публикаций, позволяющих определить уровень прогресса до выполнения нашего исследования. Был известен «наивный» подход, предполагающий последовательное использование двух алгоритмов. Первый из них — алгоритм из [2] с параметром $t = O(\log n)$. Второй — алгоритм FRT из [6], позволяющий приблизить произвольную метрику с точностью $O(\log n)$. Отсюда получаем итоговую асимптотику верхней оценки на искажение математического ожидания веса кратчайшего пути $O(\log^2 n)$.

1.4.1. Удаление коротких циклов

Алгоритм удаления коротких циклов представлен в общем виде в [2]. Мы будем именовать эту стадию соответствующе. Также показано, что в любом графе без циклов длины не более $O(t)$ проведено не более $O(n^{1+\frac{1}{t}})$ рёбер.

Будем добавлять рёбра исходного графа в пустой граф последовательно. В случае, если в текущем графе G' добавляется ребро (u,v) и $d_{G'}(u,v) \leqslant t$,

Листинг 1 – Удаление коротких циклов

```
function Spanner(G = (V, E), t)
    G' := (V, ∅);
    for e = (u, v) ∈ E do
        Compute d_{G'}(u, v), length of the shortest path from u to v in G';
        if t < d_{G'}(u, v) then
            Add e to G';
        end if
    end for
    return G';
end function
```

ребро не вставляется в граф. Для поиска кратчайших расстояний в конструируемом в процессе работы алгоритма графе G' в случае невзвешенного графа можно использовать поиск в ширину, который потребует количества памяти, равного $O(m) = O(n^{1+\frac{1}{t}})$, что не превосходит необходимых $O(n \log^{\alpha} n)$ при соответствующем выборе параметра t. Возьмём $t = O(\log n)$. Тогда мы получим граф с $O(n^{1+\frac{1}{\log n}}) = O(n \cdot e) = O(n)$ рёбрами, а значит, и в процессе в графе G' число рёбер не превосходило требуемого ограничения по памяти. Полученное же искажение при $t = O(\log n)$ не будет превышать $O(t) = O(\log n)$.

Обоснование необходимости применения этой части состоит в том, что в произвольном графе число рёбер может быть велико, и мы не сможем применить алгоритм FRT напрямую к входному графу из-за ограничений, обусловленных стриминговой моделью вычислений — для FRT нужно искать расстояния между парами вершин, и для этого приходится, например, хранить рёбра графа. В случае, если их количество превосходит $O(n \log^{\alpha} n)$, становится невозможным хранить их при стриминговой модели вычислений.

1.4.2. Алгоритм FRT

Второй этап — применение алгоритма из [6], он выдаёт на выходе распределение метрик деревьев с дополнительным искажением $O(\log n)$. Мы подробно обсудим структуру этого алгоритма и проведем его анализ во второй главе.

1.4.3. Итоговый «наивный» алгоритм

Таким образом, итоговое математическое ожидание искажения кратчайшего пути между произвольными вершинами u и v исходного графа после

двух этапов не превосходит $O(\log n) \cdot O(\log n) = O(\log^2 n)$. Этот результат мы и будем считать исходным для данной работы.

Выводы по главе 1

Сформулирован «наивный» алгоритм, решающий требуемую задачу приближения произвольной графовой метрики вероятностным распределением метрик деревьев в стриминговой модели вычислений с искажением не более $O(\log^2 n)$. Мы будем стремиться уменьшить эту оценку в работе.

ГЛАВА 2. СТРИМИНГОВЫЙ АЛГОРИТМ ДЛЯ ПРИБЛИЖЕНИЯ МЕТРИКИ РАСПРЕДЕЛЕНИЕМ МЕТРИК ДЕРЕВЬЕВ

2.1. План действий

Описанный в первой главе исходный алгоритм решает поставленную задачу о приближении метрики с искажением $O(\log^2 n)$. Это искажение состоит из двух частей, соответствующих двум описанным фазам алгоритма — в каждой из фаз искажение метрики составляет $O(\log n)$. Мы проанализируем обе фазы с целью оптимизации этих оценок.

2.2. Анализ фазы «удаление коротких циклов»

Суть оценки алгоритма этой фазы состоит в том, что в Алгоритме 1 ребро исходного графа исключается в том и только в том случае, если его добавление в граф G' замыкает некоторый цикл длины не более $t+1$. Тогда для любого пути в исходном графе G рассмотрим его рёбра в графе G', который возвращён алгоритмом. Если какое-либо ребро этого пути было исключено в процессе работы алгоритма, то к этому моменту оно замыкало бы цикл длины не более $t+1$, значит, в G' существует путь длины не более t, соединяющий концы этого исключённого ребра. Таким образом, искажение каждого ребра на рассматриваемом пути не более t, а значит, и искажение расстояния между концами пути не превосходит t. Возникает вопрос, можно ли улучшить эту оценку, например, в смысле математического ожидания при случайном порядке добавления ребер? К сожалению, анализ, необходимый для ответа на этот вопрос, затрагивает области вероятностной комбинаторики на графах, представляется существенно сложным и проведён не был. Однако же, была осуществлена программная реализация этого алгоритма и сделаны наблюдения о характере зависимости искажения длины пути для случайных графов в модели Эрдёша-Реньи, случайных масштабно-инвариантных графов, сгенерированных в модели Чанг-Лу [1], а также для социального подграфа сети «Facebook», взятого из [9]. Эти наблюдения описаны в разделе 2.5.3.

Вернёмся к теоретическим оценкам. Заметим, что предположение о порядке параметра t, доставшееся нам от «наивного» решения, может быть скорректировано. Значение параметра $t = O(\log n)$ было выбрано, в частности, чтобы сбалансировать искажение метрики на двух этапах — сделать его логарифмическим в обеих фазах. Однако, было упущено наблюдение, что порядок количества рёбер, остающихся в графе после применения алгоритма, даже

чересчур мал и составляет всего $O(n)$ при позволенных $O(n \log^{\alpha} n)$. Заметим также, что параметр t влияет непосредственно лишь на оценку сверху количества рёбер, остающихся в графе после окончания этой фазы, а также на оценку сверху искажения метрики — других ограничений на работу алгоритма изменения параметра не накладывает.

Таким образом, мы можем вычислить порядок значения параметра t, исходя не из желаемой асимптотики искажения, а из ограничения на используемую алгоритмом память. Так, если $n^{1+\frac{1}{t}} \sim n \log^{\alpha} n$, то $n^{\frac{1}{t}} \sim \alpha \cdot \log n$, тогда $\frac{1}{t} \sim \frac{\log \log n}{\log n}$, то есть $t = O(\frac{\log n}{\log \log n})$. Заметим, что мы получили улучшение оценки искажения на первой фазе в $\log \log n$ раз.

2.3. Анализ алгоритма FRT

2.3.1. Термины и обозначения

Для проведения анализа алгоритма FRT понадобится ввести несколько обозначений. Большая их часть эквивалентна обозначениям, вводимым авторами [6] в статье об алгоритме.

В алгоритме выбирается случайная перестановка π вершин графа, а также случайное число β из распределения на промежутке $(1,2)$ с плотностью $p(x) = \frac{1}{x \ln 2}$, Δ — диаметр графа, без потери общности и для упрощения выкладок полагаем $\Delta = 2^{\delta}, \delta \in \mathbb{Z}$. Исходное расстояние между вершинами будем обозначать метрикой $d(\cdot, \cdot)$, а полученное в результате работы алгоритма приближающее расстояние — $d'(\cdot, \cdot)$. Кроме того, говоря об алгоритме FRT, мы будем называть ребром не именно ребро исходного графа, но произвольную пару вершин в нём, поскольку в терминах метрики все рёбра в графе проведены с некоторыми весами.

Семейство подмножеств S множества V называется *ламинарным*, если $\forall U,V \in S\colon U \cap V \neq \varnothing \implies U \subset V \vee V \subset U$.

Декомпозиция порядка r — разбиение множества вершин V на кластеры, каждый из которых имеет центр в некоторой вершине и радиус меньше, чем r. Диаметр каждого кластера, отсюда, меньше $2r$.

Иерархическая декомпозиция графа — последовательность декомпозиций D_i, такая что:

- $D_{\delta} = \{V\}$
- D_i — декомпозиция порядка 2^i, а также $\forall A \in D_i\ \exists B \in D_{i+1} \mid A \subset B$, то есть любой кластер из D_i содержится в некотором кластере из D_{i+1}.

2.3.2. Алгоритм FRT

Алгоритм строит иерархическую декомпозицию множества вершин графа V, которая образует ламинарное семейство подмножеств. Она может быть интерпретирована как дерево, подвешенное за корень, представляющий тривиальное разбиение на единственный кластер. Условимся нумеровать уровни дерева от 0 до δ в порядке от листьев к корню. Тогда каждая вершина дерева символизирует кластер на нулевом уровне, а i-тый уровень дерева есть некоторая декомпозиция порядка 2^i. Так, вершины нулевого уровня, то есть листья дерева, суть кластеры радиуса менее единицы, а значит, содержат лишь свои центры, ведь расстояние от вершины до любой другой не меньше единицы. В связи с этим не имеет смысла разбивать кластеры этого уровня на меньшие, и уровней ниже у дерева нет.

Опишем процесс построения иерархической декомпозиции графа, приведённый авторами [6]. После равновероятного выбора случайной перестановки π вершин графа и случайного числа β начинаем построение дерева с уровня δ, на котором будет находиться лишь корень дерева — единственный кластер радиуса с произвольным центром, включающий в себя все вершины. Приведём псевдокод алгоритма FRT.

Листинг 2 – Алгоритм FRT

```
function FRT(G = (V, E))
    π ← random permutation of order n = |V|
    β ← random number from (1,2) with density p(x) = 1/(x ln 2)
    D_δ ← V, i ← δ − 1
    while D_{i+1} has non-singleton clusters do
        β_i ← 2^{i−1} β;
        for l in [1 . . . n] do
            for cluster S in D_{i+1} do
                create a new cluster from all the unassigned vertices in S
                closer than β_i to π(l)
            end for
        end for
        i ← i − 1
    end while
end function
```

В полученном дереве положим вес ребра из вершины S на i-том уровне в каждую из её дочерних вершин $i-1$-ого уровня равным 2^i. Напомним, что при

этом радиус каждого кластера на i-том уровне не превосходит 2^i, а диаметр, соответственно, не больше 2^{i+1}.

Отметим, что на каждом фиксированном уровне дерева каждая из вершин дерева принадлежит какому-то из кластеров, причём только одному. Тогда листья дерева декомпозиции взаимно однозначно соответствуют вершинам исходного графа. Воспользуемся этим и положим итоговое приближающее расстояние между двумя произвольными вершинами u и v как расстояние в полученном дереве декомпозиции между кластерами-листьями, содержащими u и v соответственно. Понятно, что $d'(u,v) \geqslant d(u,v)$: рассмотрим наименьший уровень l, на котором u и v принадлежат одному кластеру S (этот кластер и есть их наименьший общий предок). Тогда $d'(u,v) = 2 \cdot \sum_{i=1}^{l} 2^i \geqslant 2^{l+1} \geqslant diameter(S) \geqslant d(u,v)$.

2.3.3. Схема доказательства алгоритма FRT

Схема доказательства у авторов [6] понадобится нам для проведения анализа алгоритма и вывода оценок на растяжение ребра снизу. Она анализирует изменение расстояния между произвольными вершинами u и v в результате работы алгоритма и выглядит следующим образом.

Будем говорить, что кластер S *покрывает* ребро (u,v) на данном уровне i, если на данном уровне S — первый кластер i-того уровня в процессе работы алгоритма, содержащий какую-либо из вершин u и v (возможно, обе). Фиксированное ребро на данном уровне покрывается одним и только одним кластером.

Также будем говорить, что кластер S *разрезает* ребро (u,v) на уровне i, если S покрывает (u,v) на этом уровне, причем S содержит ровно одну из вершин u и v, то есть не содержит их обе.

Как следует из построения дерева, расстояние между вершинами u и v определяется их наименьшим общим предком, то есть кластером наименьшего уровня, которому принадлежали одновременно u и v. На всех уровнях с меньшими номерами, а значит, и с меньшими радиусами, ребро (u,v) будет разрезано некоторым кластером.

Положим $d_w^T(u,v) = \sum_i \mathbf{1}(w \text{ cuts } (u,v)) \cdot 2^{i+2}$, где $\mathbf{1}$ — функция-индикатор события, а w — некая вершина и центр некоторых кластеров. Понятно, что $\sum_w d_w^T(u,v) \geqslant d'(u,v)$, потому что для каждого i-того уровня ниже уровня l наименьшего общего предка u и v в дереве FRT есть два ребра с $i+1$-

го уровня на i-й и $d'(u,v) = \sum_{i=0}^{l-1} 2^{i+2}$, при этом на каждом из этих уровней существует кластер w, разрезающий (u,v), а значит, вносящий вклад 2^{i+2} в сумму $\sum_w d_w^T(u,v)$.

Расположим все вершины в порядке нестрогого увеличения расстояния до ребра (u,v), рассмотрим s-тую вершину в этом списке. Не умаляя общности, $d(w_s,u) \leqslant d(w_s,v)$. Математическое ожидание $d_{w_s}^T(u,v)$ может теперь быть оценено сверху. Чтобы кластер с центром w_s разрезал ребро (u,v) на некотором уровне i, необходимые и достаточные условия таковы:

а) $d(w_s,u) \leqslant \beta_i < d(w_s,v)$

б) w_s покрывает ребро (u,v) на i-том уровне.

Важное свойство выбранного распределения для β состоит в том, что плотность распределения $\beta_i = 2^{i-1}\beta$ в пределах $(2^{i-1}, 2^i)$ равна $p(x) = \frac{dx}{x \ln 2}$. При фиксированном значении β_i условная вероятность события б) при условии события а) не превосходит $\frac{1}{s}$: первые s вершин заведомо могут покрыть ребро (u,v), и сделают это, если окажутся в перестановке π раньше, чем w_s. Вклад w_s в $d_{w_s}^T(u,v)$ на i-том уровне ограничен $2^{i+2} \leqslant 8\beta_i$. Тогда

$$\mathbf{E}\left[d_{w_s}^T(u,v)\right] \leqslant \int_{d(w_s,u)}^{d(w_s,v)} \frac{dx}{x \ln 2} \cdot 8x \cdot \frac{1}{s} = \frac{8}{s \ln 2}(d(w_s,v) - d(w_s,v)) \leqslant \frac{8d(u,v)}{s \ln 2} \tag{1}$$

Теперь из линейности математического ожидания может быть получено:

$$\mathbf{E}\left[d'(u,v)\right] \leqslant \sum_w \mathbf{E}\left[d_w^T(u,v)\right] \leqslant \sum_s \frac{8d(u,v)}{s \ln 2} = \frac{8d(u,v)}{\ln 2} \cdot \sum_{s=1}^{n} 1/s = 8 \log n \cdot d(u,v) \tag{2}$$

2.3.4. Существенные неравенства в доказательстве алгоритма FRT

Отметим, какие из неравенств, использованных авторами [6] в доказательстве алгоритма FRT, являются существенными, то есть влияют на возможную итоговую оценку асимптотики искажения $O(\log n)$, а какие являются асимптотическими равенствами, то есть равенствами с точностью до константы. Для последних мы сможем написать оценку не сверху, как выше, а снизу, и таким образом проанализировать, за счёт каких частей в доказательстве оцен-

ка асимптотики искажения ребра может быть улучшена, а за счёт каких не может.

Существенными для нас будут являться следующие неравенства:

а) Вероятность б) при условии а) не превосходит $\frac{1}{s}$.

На самом деле кроме первых s вершин, которые могут покрыть ребро (u,v) на данном уровне при фиксированном значении радиуса кластера β_i, могут быть вершины с индексами большими s, находящиеся на таком же расстоянии от (u,v), как и w_s.

б) $\frac{8}{s \ln 2}(d(w_s,v) - d(w_s,v)) \leqslant \frac{8d(u,v)}{s \ln 2}$

Неравенство треугольника в метрике выполнено по определению последней, но тот факт, насколько оно «близко» к равенству, сложно оценить в случае произвольных рассматриваемых графов.

в) $\sum_w \mathbf{E}\left[d_w^T(u,v)\right] \leqslant \sum_s \frac{8d(u,v)}{s \ln 2}$

В ситуации, когда предыдущие два пункта, являются асимптотическими равенствами для конкретного набора индексов S, оценка $\mathbf{E}\left[d_{w_s}^T(u,v)\right] \sim \frac{d(u,v)}{s}$ может оказаться недостаточной, чтобы дать оценку снизу $\Omega(\log n)$ на искажение расстояния. Например, если $S = [\frac{n}{2} \ldots n]$, то в последнем переходе оценка даст $\sum\limits_{s \in S} \frac{1}{s} \sim \ln n - \ln \frac{n}{2} = \ln 2 = const$, то есть мы не смогли оценить снизу искажение ребра больше, нежели константой. В связи с этим нужно обратить внимание на то, для каких наборов индексов равенства из двух предыдущих пунктов обращаются в асимптотические равенства.

Оценка величины вклада в $d_{w_s}^T(u,v)$ равная $8x$ в уравнении 1 является асимптотическим равенством, потому что этот вклад на i-том уровне равен $2^{i+2} \geqslant 4\beta_i$. Оценка плотности распределения величин радиусов кластеров β_i тоже, очевидно, является равенством с точностью до константы.

2.3.5. Итоги анализа алгоритма FRT

Как видно из предыдущего раздела, условия, при которых достигается асимптотическое равенство оценок растяжения ребра в алгоритме FRT, в совокупности в общем случае не позволяют определить в явном виде графы и пары их вершин, растяжение которых достигает верхней асимптотической оценки.

2.4. Точность оценок

Мы описали итоговый стриминговый алгоритм, оценив для каждого расстояния в графе его искажение сверху асимптотикой $O(\frac{\log^2 n}{\log\log n})$. Точна ли эта оценка? Покажем, что существуют семейства графов, на которых эта оценка достигается.

2.4.1. Регулярный граф

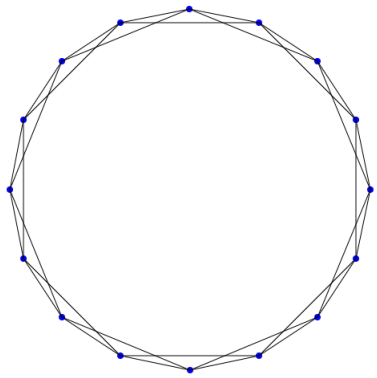

Рисунок 1 – Регулярный граф

Построим регулярный граф степени $\frac{2\log n}{\log logn}$: расположим все n вершин графа по кругу и соединим ребрами каждую из вершин с $k = \frac{\log n}{\log\log n}$ ближайшими к ней с обеих сторон. Применим к этому графу наш алгоритм: после удаления циклов длины не более $k+1$ могло случиться так, что все рёбра между вершинами, соседними в смысле расположения на окружности, проведены, а все остальные стёрты — например, если все первые рёбра оказались раньше в порядке попытки вставки в граф на стадии удаления коротких циклов. Таким образом, после первой части алгоритма остался цикл длины n, и вершины u и v на расстоянии k были в исходном графе соседними, то есть расстояние между ними увеличилось как раз в $k = \frac{\log n}{\log\log n}$ раз. Таким образом, на этой стадии для такого ребра (u,v) достигнута верхняя оценка искажения.

Оценим теперь искажение расстояния между вершинами u и v на расстоянии k в цикле длины n при применении алгоритма FRT. Возвращаясь к существенным неравенствам в анализе алгоритма FRT, отмеченным в предыдущей главе, отметим, что неравенство треугольника $|d(w_s,v) - d(w_s,u)| \leqslant d(u,v)$ обращается в равенство для всех вершин, за исключением промежутков между u и v, а также между вершинами, диаметрально противоположными u и v, то есть для набора индексов

$S = [2k+1 \ldots n-2k] = \left[\dfrac{2\log n}{\log\log n}+1 \ldots n-\dfrac{2\log n}{\log\log n}\right]$ оно заведомо выполнено. Также заметим, что на фиксированном расстоянии d от «ребра» (u,v) находится не более 4 вершин — таким образом, оценка снизу вероятности б) при условии а) для s-той вершины в порядке отдаления от (u,v) есть $\dfrac{1}{s+4}$. Таким образом, может быть получена итоговая оценка снизу на математическое ожидание нового расстояния $d'(u,v)$:

$$\mathbf{E}\left[d'(u,v)\right] \geqslant \sum_{w} \mathbf{E}\left[d_w^T(u,v)\right] \geqslant \sum_{s\in S} \frac{4d(u,v)}{(s+4)\ln 2} = \frac{4d(u,v)}{\ln 2} \cdot \sum_{s=\frac{2\log n}{\log\log n}}^{n-\frac{2\log n}{\log\log n}} \frac{1}{s+4} \geqslant$$

$$\geqslant \frac{4d(u,v)}{\ln 2} \cdot \sum_{s=\log n}^{\frac{n}{2}} \frac{1}{s} \sim (\log n - const - \log\log n)\cdot d(u,v) = \Omega(\log n)\cdot d(u,v) \tag{3}$$

Следовательно, мы получили, что итоговое искажение расстояния 1 между соседями u и v в описанном графе будет порядка оценки сверху $\frac{\log^2 n}{\log\log n}$, обозначенной нами выше.

Недостаток этого примера состоит в том, что в нём и без первой стадии удаления циклов малой длины всего рёбер порядка $O(n\log n)$, и алгоритм второй стадии можно применить к исходному графу.

2.4.2. Граф-звезда

Построим другой пример. Рассмотрим $0<\alpha<1$. Одна из вершин графа будет его центром, из которого выходят n^α цепочек длины $n^{1-\alpha}$ каждая. Назовём этот граф G'. Рассмотрим теперь все вершины на расстоянии не более $\dfrac{\log n}{\log\log n}$ от центра в G' и проведём все рёбра между ними, получим граф G. Заметим, что в нём порядка $\left(\dfrac{n^\alpha \log n}{\log\log n}\right)^2$ рёбер, а после удаления циклов длины не более $\dfrac{2\log n}{\log\log n}$ мог остаться граф G'. Рассмотрим в нём u — центр и v — вершину на расстоянии $\dfrac{\log n}{\log\log n}$ от него. Для каждого фиксированного расстояния d существует не больше n^α вершин, удалённых от (u,v) на расстояние d. Оценка снизу вероятности б) при условии а) для s-той вершины в порядке отдаления от (u,v) получится $\dfrac{1}{s+n^\alpha}$, а неравенство треугольника выполнено

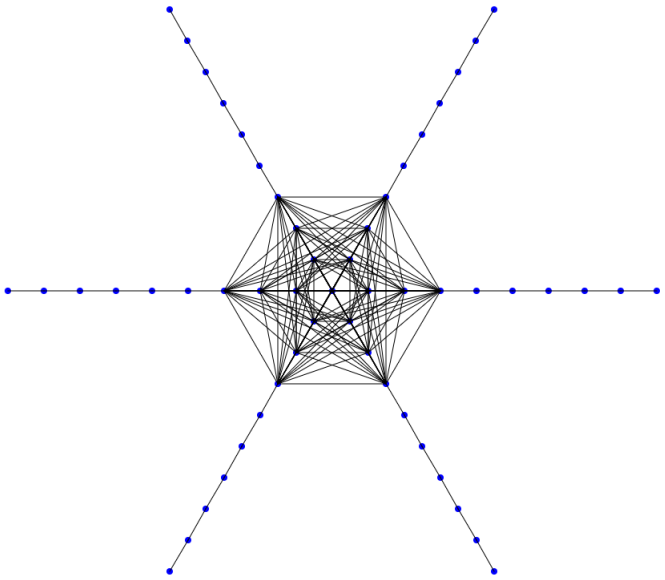

Рисунок 2 – Граф-звезда

вообще для всех s, потому что G' — дерево, то есть множество S индексов, по которым суммируем оценки $\mathbf{E}\left[d_{w_s}(u,v)\right]$ есть $S = [1 \dots n]$. Тогда

$$\mathbf{E}\left[d'(u,v)\right] \geqslant \sum_{s \in S} \frac{4d(u,v)}{(s+n^\alpha)\ln 2} = \frac{4d(u,v)}{\ln 2} \cdot \sum_{s=1}^{n} \frac{1}{s+n^\alpha} = \frac{4d(u,v)}{\ln 2} \cdot \sum_{s=n^\alpha}^{n} 1/s \sim$$

$$\sim (\log n - \log^\alpha n) \cdot d(u,v) = (1-\alpha)\log n \cdot d(u,v) = \Omega(\log n) \cdot d(u,v) \quad (4)$$

Таким образом, в этом графе искажение расстояния $d(u,v) = 1$ также достигло оценки сверху, показанной нами выше, и составило $\Omega(\frac{\log^2 n}{\log\log n})$.

2.5. Реализация

Описанный в текущей главе алгоритм был реализован на языке Kotlin. Расскажем подробнее о реализации обеих стадий алгоритма.

2.5.1. Реализация удаления коротких циклов

Реализация первой стадии по большей части тривиальна и соответствует псевдокоду 1. В момент попытки добавления каждого ребра запускается поиск в ширину с целью обнаружения короткого цикла, замыкаемого этим ребром. На каждой итерации добавления ребра алгоритм занимает время $O(n+m_i)$, где m_i — количество рёбер в графе в данный момент. Мы знаем, что в итоговом графе ребер окажется всего $O(n \log n)$, значит $\forall i \; m_i = O(n \log n)$. Поэтому

итоговая асимптотика равна $O(\sum_{i=1}^{m} O(n + m_i)) = O(mn \log n)$. Память, требуемая на этой стадии, ограничена $O(n \log n)$ как максимальным числом рёбер итогового графа, ведь поиск в ширину требует лишь $O(n)$ дополнительной памяти.

2.5.2. Реализация алгоритма FRT

Идея для упрощения реализации алгортма FRT была заимствована у авторов статьи [5] о параллельной реализации алгоритма. Совмещение концепций параллельного программирования и стриминговой модели вычислений представляет собой принципиально новую задачу, требующую определения новой модели для вычислений, а потому выходящую за рамки настоящего исследования. Исходя из этого, было принято решение реализовать последовательный алгоритм.

Идея, позволяющая не хранить в явном виде ламинарное семейство подмножеств вершин графа, образующее дерево иерархической декомпозиции, состоит в том, чтобы хранить в каждой вершине дерева лишь информацию о центре кластера. Такое дерево декомпозиции можно построить описанным ниже способом:

Для каждой вершины v будем искать последовательность $\chi_i, i \in [0 \dots \delta]$, центров кластеров на i-ом уровне дерева декомпозиции. Напомним, что уровни нумеруются в порядке от листьев к корню, и кластеры i-ого уровня имеют радиус $2^{i-1}\beta$. Таким образом, согласно алгоритму FRT, центром кластера для вершины v на i-ом уровне окажется первая вершина c в смысле перестановки π, такая что $d(v, c) < 2^{i-1}\beta$. Каждую вершину, занумерованную от 0 до $n - 1$, можно обрабатывать по отдельности. Для построения описанной последовательности центров будем хранить перестановку π^{-1}, позволяющую узнать место, на котором стоит вершина под номером i в перестановке π. Отсортируем вершины в порядке удаления от v, будем называть этот массив a. Заведём также вспомогательный массив для хранения минимума величины $\pi^{-1}(a_i)$ на префиксе. Теперь будем для каждого уровня i и его кластера радиуса $r = 2^{i-1}\beta$ двоичным поиском искать в отсортированном массиве a последнюю вершину u, удалённую от v на расстояние не более r, а затем искать минимум π^{-1} на префиксе до u включительно. Несложно видеть, что мы нашли индекс искомого центра кластера в перестановке π, и чтобы получить номер вершины, осталось лишь снова применить перестановку. Теперь вставим полученные после-

довательности кластеров в префиксное дерево и получим искомое дерево декомпозиции. Веса, приписываемые рёбрам, можно учитывать на этапе поиска расстояний. Очевидно, что для поиска расстояния между произвольной парой вершин u и v достаточно знать лишь глубину дерева и глубину наименьшего общего предка $lca(u, v)$.

Объясним асимптотику затрат времени для предложенной реализации. Для каждой из n вершин мы осуществляем сортировку расстояний от неё до остальных, это занимает $O(n \log n)$ времени на добавление одной вершины. Дальнейшие расходы ограничиваются $O(n)$ для построения вспомогательных массивов и $O(\log n)$ на их использование для поиска каждого из $\delta = \log D < \log N$ центров кластеров, таким образом, итоговое время построения дерева декомпозиции составляет $O(n^2 \log n)$. Поиск наименьшего общего предка в полученном дереве декомпозиции, а значит, и поиск произвольного приближающего расстояния между произвольными вершинами-листьями дерева может быть произведён за $O(1)$ на запрос и $O(n)$ предподсчёта алгоритмом Фараха-Колтона и Бендера, однако для простоты реализации и в силу небольших порядков величин n, доступных для локального тестирования из-за асимптотики $O(n^2 \log n)$ по времени, для поиска наименьшего общего предка был реализован метод двоичных подъёмов, работающий за $O(\log n)$ на запрос и $O(n \log n)$ предподсчёта.

Отметим, что в статье [6] исходно предложен алгоритм, работающий, по словам авторов, за $O(n^3)$ и упомянуто, что реализация может быть оптимизирована до $O(n^2)$. Можно, однако, видеть, что при учёте деталей реализации хранения собственно кластеров дерева упомянутые оценки возрастают соответственно до $O(n^3 \log n)$ и $O(n^2 \log n)$. Таким образом, быстрейшая известная нам последовательная реализация, служащая основой для дальнейшего многопоточного алгоритма в [5], работает за $O(n^2 \log n)$.

2.5.3. Наблюдения

Были проведены тестовые запуски реализованных алгоритмов для изучения характеристик алгоритмов на некоторых семействах графов. Наибольший интерес представляет изучение работы алгоритмов на графах, близких к реальным. Многие реальные графы, например, граф узлов сети Интернет, биологические цепи питания, сети нейронных связей, согласно исследованиям имеют показательное распределение степеней [7] с основаниями

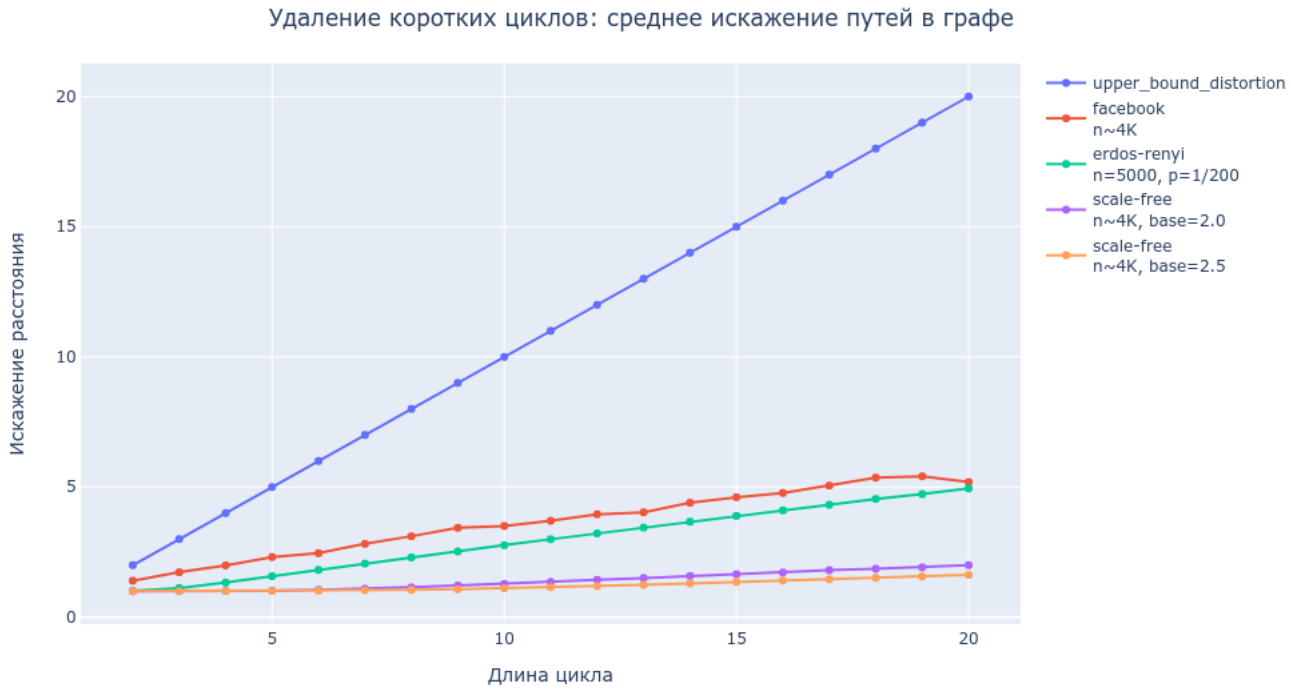

Рисунок 3 – Среднее искажение расстояний при удалении коротких циклов

$2 < base < 3$, поэтому были рассмотрены т.н. масштабно-инвариантные *(англ. scale-free)* графы, имеющие распределение степеней вершин близкое к показательному[7]. С помощью модели Чанг-Лу [1] были сгенерированы случайные масштабно-инвариантные графы, имеющие распределение степеней, близкое к показательному. Также были сгенерированы случайные графы Эрдёша-Реньи, рассмотрен подграф социальной сети «Facebook» [9].

На рисунке 3 показано, как в среднем изменяется расстояние между всевозможными парами вершин в графе для различных графов на этапе удаления коротких циклов. Видно, что верхняя оценка на это искажение довольно груба и не достигается на моделях реальных графов. Строгий анализ причин этого явления не был осуществлен в рамках данной работы, однако представляет определённый научный и прикладной интерес.

Искажение длинных путей на этапе удаления коротких циклов может быть меньше ещё и потому, что различные пути между фиксированной парой вершин могут быть подвержены различному изменению в процессе работы алгоритма, и после его завершения будет выбран кратчайший искажённый путь. Таким образом, существование нескольких путей между парой вершин может уменьшать итоговое искажение длины пути. В связи с этим целесообразно рас-

смотреть искажение соседних вершин в графе. На рисунке 4 показано среднее искажение ребра при изменении параметра — верхней границы длины удаляемых из графа циклов. Отсюда видно, что масштабно-инвариантные модели не идеально приближают характеристики реальных графов: искажение на подграфе социальной сети «Facebook» оказалось выше.

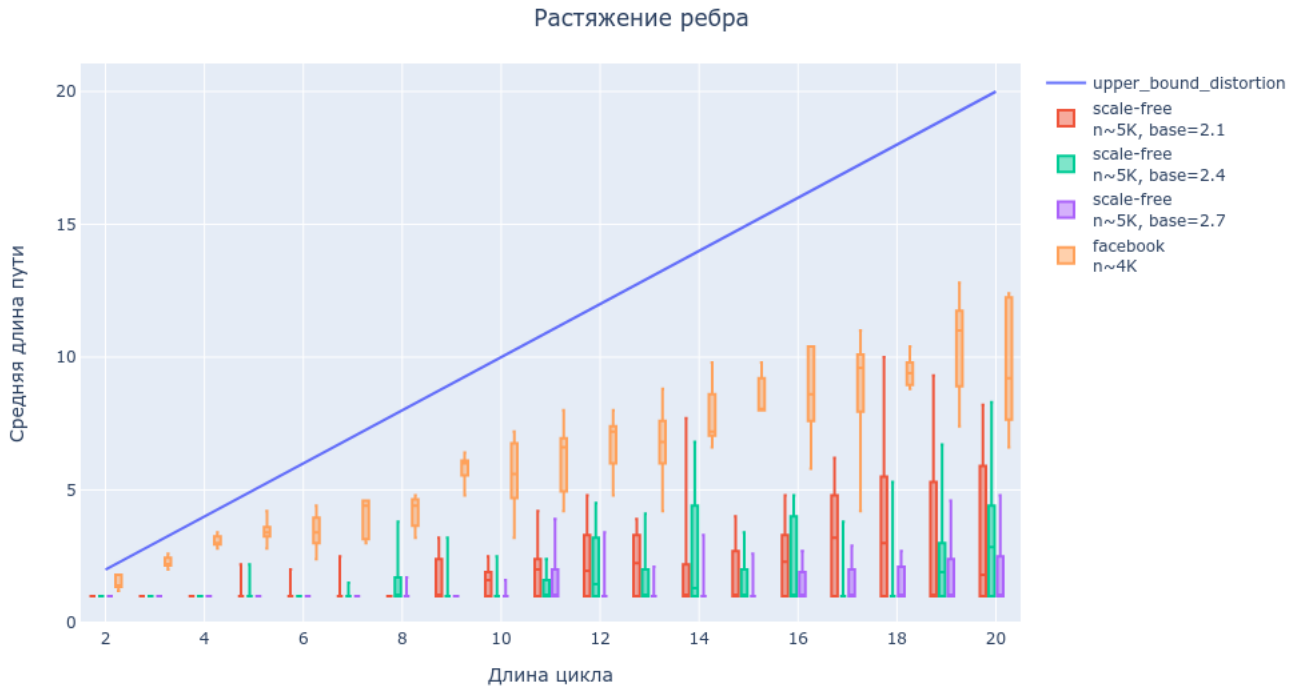

Рисунок 4 – Среднее искажение рёбер при удалении коротких циклов

Также были произведены запуски алгоритма FRT на моделях масштабно-инвариантых графов. На рисунке 5 показано среднее искажение расстояния 1 между соседними вершинами для различных оснований показательного распределения. Из графика видно, что среднее искажение рёбер действительно меньше оценки сверху на математическое ожидание такового искажения в большинстве измерений. Однако, можно заметить, что порядок величин искажений, полученных эмпирически, близок к порядку оценки сверху математического ожидания искажения $O(\log n)$, верной для любых графов. В связи с этим становится интересной задача аналитического доказательства асимптотического равенства этих величин для масштабно-инвариантных графов.

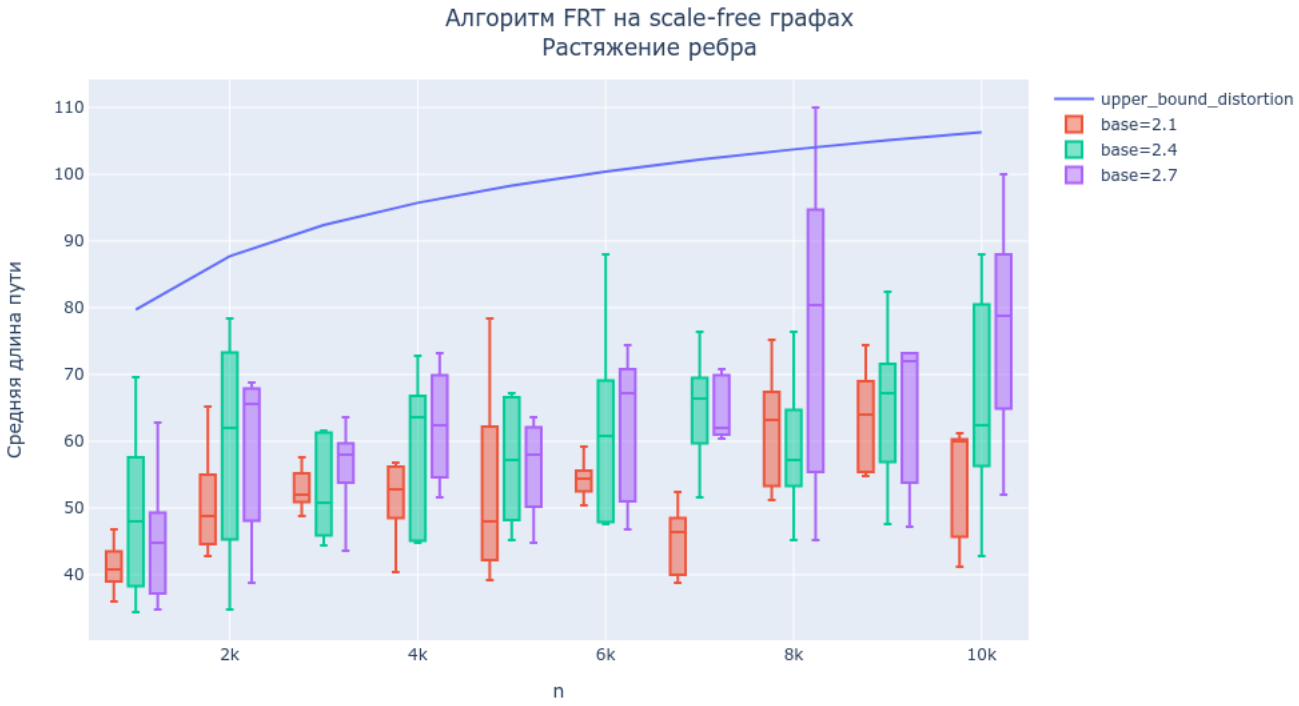

Рисунок 5 – Алгоритм FRT. Среднее искажение рёбер

Выводы по главе 2

Мы рассмотрели и исследовали две фазы исходного алгоритма на предмет возможной оптимизации оценки целевой асимптотики растяжения произвольного расстояния. В то время, как в фазе удаления коротких циклов улучшение было достигнуто в общем случае, свойства алгоритма FRT не позволяют нам уточнить оценку растяжения в общем случае, для произвольных графов.

Кроме того, мы привели примеры показывающие точность полученных оценок: существуют графы и пары их вершин, для которых матожидание искажения расстояния достигает означенной верхней границы асимптотики.

ГЛАВА 3. СВОЙСТВА АЛГОРИТМА FRT

В этой главе изложены наблюдения, касающиеся работы алгоритма FRT в случае его применения на некоторых классах графов и их рёбер. Знание о том, как растягиваются некоторые рёбра таких графов может быть полезно как с академической, так и с практической точки зрения, если имеется представление о том, что граф, к которому применяется алгоритм FRT, обладает теми или иными свойствами.

Важное примечание к этой главе в связи с рассуждениями о величинах, носящих асимптотический характер, заключается в следующем: мы рассматриваем *классы* графов, обладающие некоторыми свойствами, и делаем заключения о характеристиках результата работы алгоритма на этом графе при числе вершин графа, устремлённом к бесконечности. Говоря, например, что длина некоторого пути не превосходит логарифма от числа вершин, мы имеем ввиду, что фиксируем некоторые константы c_1, c_2 и изучаем последовательность представителей рассматриваемого класса графов с числами вершин n, стремящимся к бесконечности. В каждом из них мы изучаем пути длины $\{d(u,v) \mid c_1 \log n \leqslant d(u,v) \leqslant c_2 \log n\}$.

3.1. Абсолютное значение приближающего расстояния

Непосредственно из описанного алгорима FRT следует, что все пути в конкретном графе на выходе алгоритма имеют длину не более $2 \cdot \sum\limits_{i=1}^{\delta} 2^i \leqslant 4 \cdot 2^{\delta} = O(2^{\delta}) = O(\Delta)$, то есть по порядку величины не превосходят диаметра исходного графа. Это значит, например, что расстояния длины порядка диаметра в исходном графе в любом случае будут иметь искажение, не превосходящее константы. Более строго, если $d(u,v) \geqslant \dfrac{\Delta}{c}, c \geqslant 1$, то из $d'(u,v) \leqslant 4\Delta$ следует $\dfrac{d'(u,v)}{d(u,v)} \leqslant 4c$. Такое наблюдение мотивирует больше анализировать и обращать внимание на растяжение именно коротких расстояний в графе, которые теоретически могут быть искажены, согласно известной оценке сверху, в $O(\log n)$ раз.

Ещё одно примечательное наблюдение, следующее из сказанного, состоит в том, что если диаметр исходного графа не больше, чем логарифм числа его вершин, то утверждение об оценке сверху искажения как $\log n$ становится бессодержательным: приближающее расстояние в любом случае окажется

порядка не более Δ, а значит, искажение даже расстояния длины константа — например, искажение расстояния 1 между соседними вершинами — не будет превосходить $O(\Delta) \leqslant O(\log n)$. Примеры таких графов, хотя и могут быть построены, но являются весьма искусственными и представляют в основном академический интерес. Мы рассмотрим этот случай подробнее в одном из следующих разделов.

3.2. Экспансия графа и её влияние на алгоритм

Экспансией мы называем меру увеличения количества вершин в графе в окрестности радиуса d от данной. Эта величина имеет непосредственное влияние на оценку вероятности разреза ребра (u, v) со стороны кластера с центром w_s. Однако, мы покажем, что в большинстве графов это не влияет на итоговую асимптотику искажения длины пути.

Приведём мотивирующие рассуждения о влиянии экспансии на интересующие нас оценки. Пусть число вершин в окрестности радиуса d выбранной вершины v задаётся функцией $f(d)$, тогда количество вершин на расстоянии ровно d от v имеет порядок $f'(d)$, а более строго, представляет собой разность $f(d) - f(d-1)$. Тогда s-тая вершина в порядке удаления от v находится на расстоянии порядка $f^{-1}(s)$, и, значит, вместе с ней на данном расстоянии находятся порядка $f'(f^{-1}(s))$ вершин, то есть уже знакомая нам оценка снизу вероятности разрезания некоторого ребра будет иметь порядок $\dfrac{1}{s + f'(f^{-1}(s))}$. Подстановка конкретной функции f в зависимости от типа графа — несложный, хотя и несколько умозрительный способ понять порядок этой оценки. Далее мы покажем, как применяются подобные рассуждения в случае конкретных типов графов.

3.2.1. Полиномиальные решётки

Конечной k-мерной решёткой порядка N называют следующий граф с числом вершин $n = N^k$: вершинами будут всевозможные последовательности длины k элементов множества $[1 \dots n]$. Соседними вершинами будут те, у которых все координаты, кроме одной, совпадают, а значения оставшейся координаты являются соседними. Рассмотрим k-мерную решётку порядка N.

Будем рассматривать искажение ребра между двумя соседними вершинами после применения алгоритма FRT. Для этого оценим, каково число вершин в окрестности радиуса d ребра (u,v). Несложно понять, что оно

имеет порядок $\Theta(d^k)$, а значит, его граница порядка $\Theta(d^{k-1})$. Тогда вероятность разрезания ребра (u,v) со стороны кластера с центром w_s не меньше $\dfrac{1}{s+\Theta(s^{1-\frac{1}{k}})} > \dfrac{1}{2s}$, а это означает, что полиномиальная экспансия сама по себе не влияет на нашу оценку искажения. Осталось понять, для каких s обращается в асимптотическое равенство неравенство треугольника $|d(w_s,v) - d(w_s,u)| \leqslant d(u,v)$. Раз мы рассматриваем случай, когда u и v соседи, то $d(w_s,v) \neq d(w_s,u)$, а значит, $|d(w_s,v) - d(w_s,u)| \geqslant 1 = d(u,v)$. Значит, и оценка снизу на матожидание приближающего расстояния выполнена:

$$\mathbf{E}\left[d'(u,v)\right] \geqslant \sum_{s\in S} \frac{4}{2s\ln 2} = \frac{4}{\ln 2}\cdot\sum_{s=1}^{n}\frac{1}{2s} \sim \frac{\log n}{2} = \Omega(\log n) \tag{5}$$

3.2.2. Деревья

Идея приближать расстояния в деревьях распределениями метрик деревьев, безусловно, звучит несколько надуманно, однако рассмотрение таких графов представляет собой академический интерес в контексте изучения свойств алгоритма. Деревья интересуют нас как минимум потому, что некоторые их типы представляют кардинально другой тип экспансии — экспоненциальный. Покажем, однако, что и это не повлияет на выведенные оценки.

Рассмотрим сначала деревья с ограниченной степенью вершин: пусть каждая степень не больше $a = const$. Тогда пусть на расстоянии d от фиксированной вершины v находятся k вершин. Тогда на расстоянии $d-1$ находятся как минимум $\frac{k}{a}$ вершин, так как из каждой из них выходит не более a рёбер в вершины на расстоянии d. По аналогии, на расстоянии $b < d$ находятся как минимум $\frac{k}{a^{d-b}}$ вершин, и всего в окрестности радиуса $d-1$ находится порядка хотя бы $\sum\limits_{i=1}^{\log d}\dfrac{k}{a^i} \sim \dfrac{k}{a-1}$ вершин, а значит, если s-тая вершина находится на расстоянии d от ребра (u,v), то $s > \dfrac{k}{a-1} \implies k < s\cdot(a-1)$, то есть оценка снизу вероятности разреза ребра кластером с центром w_s есть $\dfrac{1}{s+k} \leqslant \dfrac{1}{a\cdot s}$, и, так как неравенство треугольника выполнено всегда, а равенство расстояний до двух соседних вершин дерева невозможно, оценка на растяжение ребра

между двумя соседними вершинами дерева приобретает знакомый вид:

$$\mathbf{E}\left[d'(u,v)\right] \geqslant \sum_{s\in S} \frac{4}{as\ln 2} = \frac{4}{\ln 2}\cdot\sum_{s=1}^{n}\frac{1}{as} \sim \frac{\log n}{a} = \Omega(\log n) \tag{6}$$

В целях изменения порядка вероятности разреза ребра кластером с центром w_s можно рассмотреть деревья с более чем экспоненциальной экспансией — например, смоделировать граф, в котором от ребра (u,v) на расстоянии d будет 2^{2^d} вершин. Оценка вероятности действительно изменится, однако в таких графах диаметр асимптотически меньше логарифма — в приведённом примере диаметр порядка $\log\log n$, потому что от ребра между соседями u и v расстояние до самой далёкой вершины порядка не больше $\log\log n$. Как мы помним, в алгоритме FRT есть тривиальная оценка сверху приближающего расстояния — оценка порядка диаметра. В случае, когда диаметр имеет порядок меньше, чем $\log n$, анализ алгоритма FRT, дающего обычно нетривиальную оценку $\log n$, становится мало осмысленным. Таким образом, подобные графы представляют мало интереса для наших наблюдений. Кроме того, рассмотрение графов с такой быстрой экспансией во многом непрактично: они почти не встречаются в практических задачах. Графы с диаметром порядка $\log n$, напротив, возникают в различных научных областях: таковыми, например, являются социальные графы, имеющие показательное распределение степеней [7].

Расскажем о ещё одном свойстве работы алгоритма FRT, хорошо видном в случае запуска на дереве: Рассмотрим для простоты изложения идеально сбалансированное бинарное дерево глубины $depth = \log\frac{n}{2}$, подвешенное за корень. Тогда приближённое расстояние между любыми двумя листьями u и v зависит от вершин в поддереве их наименьшего общего предка $lca(u,v)$: действительно, расстояния ото всех остальных вершин x до u и v равны, так как кратчайшие пути от x до v и от x до u в таком случае должны будут проходить через $lca(u,v)$, а значит, иметь общую часть. Несовпадающие же части этих кратчайший путей равны по длине, так как эти длины равны разностям уровней $depth(lca(u,v)) - depth(u) = depth(lca(u,v)) - depth(v)$, так как u и v находятся на одинаковой глубине, как и все листья. Тогда оценка 1 вырождается в 0 для всех таких w_s. Для бинарного дерева, значит, ненулевой вклад

в $\mathbf{E}\left[d'(u,v)\right]$ будут вносить разве лишь s от 1 до, собственно, $O(2^{d(u,v)})$, то есть

$$\mathbf{E}\left[d'(u,v)\right] \leqslant \sum_{s} \frac{8d(u,v)}{s \ln 2} = \frac{8d(u,v)}{\ln 2} \cdot \sum_{s=1}^{2^{d(u,v)}} 1/s = O((d(u,v))^2) \tag{7}$$

С одной стороны, это означает, что оценка сверху на искажение $\dfrac{d'(u,v)}{d(u,v)}$ составит $O(d(u,v))$. Однако же, пар листьев на расстоянии асимптотически меньше, чем $\log n$, в дереве мало. Это объясняется тем, что для каждого листа сбалансированного бинарного дерева существует ровно 2^{k-1} листьев на расстоянии $2k$ от него, где $k \in [1 \ldots depth]$. Можно показать, что в среднем искажение расстояния между всеми парами листьев в таком дереве оценивается сверху константой, но это происходит из-за того, что большинство расстояний между листьями велики и имеют величину порядка $\Theta(\log n) = \Theta(\Delta)$, а значит, для них искажение можно оценить сверху константой из соображений, предъявленных нами выше, касающихся тривиальной оценки абсолютной величины приближаюшего расстояния величиной порядка диаметра. Иными словами, алгоритм FRT всегда возвращает расстояние не больше порядка диаметра, а среди наших исходных расстояний между листьями многие близки к этому диаметру и уже не могут быть сильно увеличены в процессе работы алгоритма.

ЗАКЛЮЧЕНИЕ

В работе была рассмотрена задача приближения метрики кратчайших расстояний графа вероятностным распределением метрик деревьев. Наивный алгоритм с асимптотикой искажения длины пути между произвольными вершинами $O(\log^2 n)$ был улучшен нами до асимптотики $O(\frac{\log^2 n}{\log\log n})$. Показано, что эта оценка точна: приведены семейства графов, на которых описанный нами алгоритм выдаёт результат с искажением длины пути $\Theta(\frac{\log^2 n}{\log\log n})$. Выполнена реализация полученного алгоритма, произведены запуски на случайных графах в модели Эрдёша-Реньи, случайных масштабно-инвариантных графах, являющихся известной моделью приближения реальных сетей, а также на социальном подграфе сети «Facebook». Кроме того, отмечены интересные закономерности, касающиеся среднего искажения длин путей при удалении из графа коротких циклов, что является составной частью нашего алгоримта. Проанализирована вторая составляющая нашего алгоритма — алгоритм FRT [6], аналитически показана точность его оценки сверху на искажение ребра $O(\log n)$ для полиномиальных решёток, а также для деревьев с ограниченной степенью вершины, рассказано о связи экспансии графа с характеристиками алгоритма FRT.

Интересным направлением для дальнейшей работы может являться аналитическое исследование зависимости искажения длин путей при удалении коротких циклов, а также изучение характеристик алгоритма FRT на моделях реальных масштабно-инвариантных социальных графов, имеющих показательное распределение степеней вершин.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Aiello W.*, *Chung F.*, *Lu L.* A random graph model for power law graphs // Experimental Mathematics. — 2001. — 10 (1). — P. 53–66.

2 *Althöfer I.* [et al.]. On sparse spanners of weighted graphs // Discrete and Computational Geometry. — 1993. — No. 9. — P. 81–100.

3 *Bartal Y.* Probabilistic approximations of metric spaces and its algorithmic applications // in: IEEE Symposium on Foundations of Computer Science. — 1996. — P. 184–193.

4 *Bartal Y.* On approximating arbitrary metrics by tree metrics // in: STOC. — 1998.

5 *Blelloch G. E.*, *Gupta A.*, *Tangwongsan K.* Parallel Probabilistic Tree Embeddings, k-Median, and Buy-at-Bulk Network Design // SPAA '12: Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures. — 2012. — P. 205–213.

6 *Fakcharoenphol J.*, *Rao S.*, *Talwar K.* A tight bound on approximating arbitrary metrics by tree metrics // Journal of Computer and System Sciences. — 2004. — No. 69. — P. 485–497.

7 *Fasino D.*, *Tonetto A.*, *Tudisco F.* Generating large scale-free networks with the Chung–Lu random graph model. — 2019.

8 *Karp R.* A 2k-competitive algorithm for the circle // Manuscript. — 1989.

9 *McAuley J.*, *Leskovec J.* Learning to Discover Social Circles in Ego Networks [Электронный ресурс]. — 2012. — URL: `http://snap.stanford.edu/data/ego-Facebook.html`.