

Санкт–Петербургский государственный университет

*Кузнецов Кирилл Александрович*

**Выпускная квалификационная работа**

*Квантовый алгоритм поиска кратчайшего пути на графе*

Уровень образования: бакалавриат

Направление 02.03.01 «Математика и компьютерные науки»

Основная образовательная программа СВ.5152.2019 «Математика,  
алгоритмы и анализ данных»

Научный руководитель:

Авдюшенко Александр Юрьевич, доцент, Факультет математики и компьютерных наук СПбГУ

Рецензент:

Исмагилов Нияз Салаватович, к.ф.-м.н., руководитель развития технологий квантовых вычислений, Общество с ограниченной ответственностью «Газпромнефть - Цифровые решения»

Санкт-Петербург

2023 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Глава 1. Обзорный раздел по предметной области</b> . . . . .	4
1.1. Квантовые подходы . . . . .	4
1.1.1 Hop-based подход . . . . .	4
1.1.2 Edge-based . . . . .	6
1.2. TTopr . . . . .	8
1.3. Постановка задачи . . . . .	9
<b>Глава 2. Проведенные эксперименты</b> . . . . .	10
2.1. Малый граф . . . . .	11
2.2. Граф $3 \times 3$ . . . . .	12
2.3. Граф $5 \times 5$ . . . . .	14
2.4. Граф $7 \times 7$ . . . . .	16
<b>Глава 3. Выводы</b> . . . . .	18
<b>Список литературы</b> . . . . .	19

## Введение

Проблема нахождения кратчайшего пути является хорошо изученной областью в теории графов. Самый простой для описания вариант этой задачи включает наличие одного источника и одного пункта назначения: необходимо найти путь с минимальной стоимостью между заданными начальной и конечной вершинами в графе с учетом стоимости переходов по ребрам. Эта задача является центральной для широкого круга практических приложений, таких как навигация, планирование маршрутов для автономных транспортных средств и анализ социальных сетей.

Квантовый отжиг — это метод оптимизации системы из  $N$  кубитов с использованием зависящего от времени гамильтониана путем постепенного уменьшения квантовых флуктуаций [1]. Гамильтониан представляет собой функцию энергии, которая начинается в полностью запутанном, случайном состоянии и постепенно превращается в гамильтониан, связанный с задачей оптимизации. Медленное развитие квантового энергетического состояния позволяет системе кубитов преодолевать энергетические барьеры и находить глобальный минимум многомерной энергетической функции. Процесс отжига начинается, когда все кубиты находятся в состоянии суперпозиции. Когда процесс завершен, кубиты находятся в основном состоянии искомого гамильтониана задачи и, следовательно, представляют собой глобальный оптимум [2]–[4]. Затем считывается и возвращается вектор решения значений кубитов. Функция энергии обычно выражается в виде модели Изинга или связанной с ней квадратичной неограниченной бинарной оптимизации (QUBO).

В данной работе исследуется QUBO постановка задачи и сравниваются методы квантового отжига и алгоритма Tensor train optimization (TTopt) [5] на данных постановках. TTopt представляет собой метод, используемый для эффективного представления многомерных тензоров (многомерных массивов) со значительно уменьшенным количеством параметров. Этот метод особенно полезен в областях машинного обучения, сжатия данных и численного моделирования, где тензоры с большим количеством измерений и элементов трудно хранить, обрабатывать и вычислять.

# Глава 1. Обзорный раздел по предметной области

В этой главе будут рассмотрены понятия, на которых основывается работа. В частности, QUBO подходы для решения проблемы поиска кратчайшего пути на графе, а также будет введен метод TТort.

Для начала рассмотрим основные определения:

**Определение.** Граф  $G = \{V, E, C\}$ . Граф состоит из множества вершин  $V = \{1, 2, \dots\}$ , множества ребер  $E = \{(1, 2), (2, 4), \dots\}$  и множества стоимостей переходов между вершинами  $C = \{c_{1,2}, c_{2,4}, \dots\} \subset \mathbb{R}_+$ . В данной работе рассматриваются ненаправленные графы.

**Определение.** Путем в  $G$  называется последовательность  $U = (v_0, \dots, v_n)$  вершин в  $V$ , где  $(v_i, v_{i+1})$  есть ребро из  $E$ . Стоимостью пути называется функция  $d(U) = \sum_{i=0}^{n-1} c_{v_i, v_{i+1}}$ . Кратчайший путь между вершинами  $s$  и  $t$  - путь, начинающийся в  $s$  и заканчивающийся в  $t$  с наименьшей стоимостью.

**Определение.** Задача QUBO с  $N$  переменными формулируется как вычисление для нахождения вектора решения  $x$ , который дает минимальную «энергию», где энергия описывается как однородный квадратичный полином.

$$H = \arg \min_x \{x^T Q x\}$$

, где  $x \in \{0, 1\}^N$ ,  $Q \in \mathbb{R}^{N \times N}$

## 1.1 Квантовые подходы

Сперва рассмотрим два подхода, сводящих задачу к QUBO [6] - бинарной квантовой оптимизации.

### 1.1.1 Hop-based подход

В данной постановке заранее вводится ограничение, что мы собираемся искать путь, который включает не более  $|P| \in \mathbb{N}_+$  вершин. Опишем пространство возможных решений как пространство пар (вершина, позиция). То есть определим результирующий путь решения как путь, имеющий вершину  $i$  в

позиции  $p$ , вершину  $j$  в позиции  $p + 1$  и так далее. Такая формулировка приводит к нескольким простым ограничениям. Во-первых, в каждой позиции должна быть вершина. Во-вторых, если вершина  $i$  находится в позиции  $p$ , а вершина  $j$  находится в позиции  $p + 1$ , то  $(i, j)$  должен находиться в множестве  $E$  и стоимость перехода равна стоимости ребра  $c_{i,j}$ . Если нет ребра  $(i, j)$ , то переходы между  $i$  и  $j$  должны быть оштрафованы.

Представим решение в виде набора двоичных переменных, каждая из которых обозначается  $x_{i,p}$  ( $i \in V, 1 \leq p \leq |P|$ ), где переменная  $x_{i,p}$  указывает, находится ли вершина  $i$  в позиции  $p$ . Иными словами, эти переменные представляют собой последовательность вершин пути, который мы хотим найти. Теперь мы можем перечислить ограничения, определенные ранее, и указать их вклад в полное уравнение энергии, которое будет описано как взвешенная сумма вкладов отдельных ограничений. ( $H = \sum_k \alpha_k H_k$ )

1. Каждая позиция должна быть занята ровно один раз на пути. Предположим, что путь проходит через  $|P| \leq |V|$  вершины. Тогда мы можем переформулировать первое ограничение как следующую квадратичную штрафную функцию:

$$H_P = \sum_{p=1}^{|P|} \left( 1 - \sum_{i=1}^{|V|} x_{i,p} \right)^2$$

Квадрат разности будет минимизироваться, когда внутренняя сумма равна единице, т.е. каждая позиция занята единственной вершиной. Весь штраф будет сведен к минимуму при 0, когда ограничение соблюдается во всех позициях.

2. Переход через ребро  $(i, j)$  влечет за собой штраф в размере  $c_{i,j}$ . Это можно легко закодировать в виде следующей квадратичной функции:

$$H_C = \sum_{p=1}^{|P|-1} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} c_{i,j} x_{i,p} x_{j,p+1}$$

Мы можем запретить переходы между вершинами  $i$  и  $j$ , которые не

соединяются ребром, установив  $c_{i,j}$  высокое значение.

3. Подразумеваются два дополнительных ограничения: путь начинается в исходной вершине и заканчивается в конечной вершине. Эти ограничения отражены в следующих штрафных терминах:

$$H_s = (1 - x_{s,1})^2 \quad H_t = (1 - x_{t,|P|})^2$$

Допустимые решения будут удовлетворять всем ограничениям; поэтому полное окончательное уравнение энергии будет (взвешенной) суммой вкладов энергии отдельных ограничений:

$$H = \alpha_P H_P + \alpha_C H_C + \alpha_s H_s + \alpha_t H_t$$

Мы хотим гарантировать, что любое нарушенное ограничение имеет более высокую стоимость, чем общая стоимость  $c_P$  кратчайшего пути. Поскольку мы не знаем  $c_P$  априори, разумной заменой является любое значение, превышающее  $c_E = \sum_{(i,j) \in E} c_{i,j}$  - сумма стоимостей всех ребер графа. Это дает коэффициенты  $\alpha_C = 1, \alpha_P = \alpha_V = \alpha_s = \alpha_t > c_E$ .

### 1.1.2 Edge-based

В данной постановке переменная  $x_{i,j}$  указывает, включено ли ребро  $(i, j)$  в путь. Будем обозначать искомым кратчайший путь через  $P$ . Рассмотрим следующую задачу с ограничениями:

$$\begin{aligned} & \text{minimize } \sum_{(i,j) \in E} c_{i,j} x_{i,j} \quad x_{i,j} \in \{0, 1\} \forall (i, j) \in E \\ & \text{and } \forall i \in V \sum_{j \in V} x_{i,j} = \begin{cases} 1, & \text{if } i \in \{s, t\} \\ 2, & \text{if } i \notin \{s, t\} \text{ and } i \in P \\ 0, & \text{if } i \notin \{s, t\} \text{ and } i \notin P \end{cases} \end{aligned}$$

В допустимом пути каждая из вершин  $s$  и  $t$  будет инцидентна одному ребру пути, каждая другая вершина пути будет инцидентна двум ребрам

пути, и каждая вершина не на пути не будет инцидента ребрам в пути. Эти ограничения можно переписать в виде квадратичных штрафных членов. Для этого необходимо включить  $|V|$  дополнительных переменных  $x_1, x_2, \dots$ , где  $x_i$  указывает, включена ли вершина  $i$  в путь.

1. Ограничения для стартовой и конечной вершины:

$$H_s = -x_s^2 + \left( x_s - \sum_j x_{s,j} \right)^2 \quad H_t = -x_t^2 + \left( x_t - \sum_i x_{i,t} \right)^2$$

Квадратный член в начале каждой функции накладывает ограничение на то, что вершины  $s$  и  $t$  должны быть включены в решение. Сумма в правой части гарантирует, что когда рассматриваемая вершина входит в решение, она инцидентна ровно одному ребру. Каждая функция минимизируется при  $-1$ , когда выполняются оба ограничения.

2. Штрафные функции для остальных вершин:

$$H_i = \left( 2x_i - \sum_j x_{i,j} \right)^2$$

Если  $i$  не входит в путь, то  $x_i = 0$ , и эта функция минимизируется в  $0$ , когда в путь не входят ребра, инцидентные  $i$ . Если  $i$  находится на пути, то функция минимизируется в  $0$ , когда включены ровно два ребра, инцидентные  $i$ .

3. Исходная целевая функция может быть переписана в квадратичном виде как

$$H_C = \sum_{(i,j) \in E} c_{i,j} x_{i,j}^2$$

Окончательное уравнение энергии:

$$H = \alpha_s H_s + \alpha_t H_t + \sum_{i \notin \{s,t\}} \alpha_i H_i + \alpha_C H_C$$

Если бы мы заранее знали стоимость  $c_P$  кратчайшего пути, то мы могли бы масштабировать  $H_C$  на  $\alpha_C = 1$ , а все штрафные члены — на любое  $\alpha > c_P$ . Это гарантировало бы, что энергия, соответствующая кратчайшему пути, будет  $-c_P < 0$ , в то время как любое невозможное решение будет иметь более высокую энергию независимо от его общей стоимости ребер. Конечно, поскольку мы не знаем длину кратчайшего пути заранее, нам нужно будет использовать соответствующую замену. Простейший выбор - это  $\alpha > c_E$ , как определено в предыдущем разделе, поскольку  $c_E \geq c_P$ , с равенством, когда граф сам является кратчайшим путем.

## 1.2 TTopt

Алгоритм TTopt [5] применяется для поиска экстремума многомерной функции  $\mathbf{J}$  размерности  $d$ , дискретизируя задачу и используя тензор с выбранными точками для решения проблемы. Для каждого непрерывного параметра  $\theta_k, k = \overline{1, d}$  целевой функции  $\mathbf{J}(\theta)$  введем сетку  $\left\{ \theta_k^{(n_k)} \right\}_{n_k=1}^{N_k}$ . В каждой точке  $(\theta_1^{(n_1)}, \dots, \theta_d^{(n_d)})$  этой сетки с индексом  $(n_1, \dots, n_d)$  целевая функция имеет значение  $\mathbf{J}(\theta_1^{(n_1)}, \dots, \theta_d^{(n_d)}) \equiv \mathcal{J}[n_1, \dots, n_d]$ . Таким образом, целевую функцию можно рассматривать как неявный  $d$ -мерный тензор  $\mathcal{J}$  с размерами компонент  $N_1, \dots, N_d$ . Нахождение максимума функции  $\mathbf{J}(\theta)$  сводится к нахождению максимального элемента тензора  $\mathcal{J}$  в дискретной постановке. Заметим, что число значений в тензоре  $\mathcal{J}$  равно:  $|\mathcal{J}| = N_1 \cdot N_2 \cdot \dots \cdot N_d \sim (\max_{1 \leq k \leq d} N_k)^d$ . Размер  $\mathcal{J}$  экспоненциально зависит от размерности  $d$ . Этот тензор не может быть оценен или сохранен для достаточно больших  $d$ . Алгоритм использует эффективные приближения для работы с многомерными массивами.

Основная идея TTopt состоит в том, чтобы выразить многомерный тензор в виде последовательности низкоразмерных тензоров, которые называются «ядрами». Ядра связаны через набор индексов, которые определяют так называемый TT ранг. Ранг является важным параметром, определяющим качество аппроксимации и вычислительную сложность этого представления. Низкий TT-ранг указывает на то, что тензор может быть представлен с небольшим количеством параметров, что приводит к эффективным вычислениям и хранению.

### 1.3 Постановка задачи

Цель данной работы - реализация QUBO подходов (hop- и edge-based) поиска кратчайшего пути на графе и применение квантового отжига и метода ТТорт для решения данной задачи, а также сравнение реализованных методов. В работе предполагается, что мы работаем с графом, который можно представить в виде “сетки”, в которой  $w$  столбцов и  $h$  рядов, разрешены кардинальные и диагональные движения. Пример графа приведен на рисунке 1.

4.37	6.36	9.73	11.98	8.43	5.15
6.94	8.76	10.51	15.28	12.83	8.06
9.27	9.61	12.16	18.62	12.00	11.76
12.74	13.64	16.73	20.13	16.51	14.70
14.41	18.23	20.05	21.41	20.21	16.77
16.41	18.56	21.88	24.01	21.91	18.91

Рис. 1: Пример поля.

Значения в ячейках - стоимость перемещения по данной ячейке. Стоимость перехода по ребру, соединяющим две ячейки есть полусумма значений в ячейках. Из каждой клеточки можно перейти в любую соседнюю. В подобном графе число вершин и число ребер имеют квадратичную асимптотику  $\mathcal{O}(w \cdot h)$  от размера графа.

Для hop-based подхода число переменных равно  $|P| \cdot |V|$ . Для исследуемых графов достаточно использовать  $P \sim w + h$ , получается кубическая зависимость числа переменных от размера графа. Для edge-based подхода число используемых переменных есть  $|E| + |V| = \mathcal{O}(w \cdot h)$ , в этом случае квадратическая зависимость от размера графа. Целевые функции в каждом из подходов используются в качестве функций для применения в ТТорт, тензоры соответственно имеют размерность  $N$ , каждая компонента хранит два значения - 0 и 1. Соответственно тензор, фактически, содержит  $2^N$  значений, где  $N$  - число переменных данной функции, описываемой матрицей  $Q$  (однако алгоритм ТТорт не хранит тензор целиком и вычисляет значения по мере необходимости).

## Глава 2. Проведенные эксперименты

Оба QUBO подхода были реализованы на языке Python. Для симуляции квантового отжига использовалась библиотека Qubover. Для применения TTopt использовалась библиотека, написанная авторами статьи [5]. Код с результатами в репозитории: [github.com/bicikar/anneal\\_ttop](https://github.com/bicikar/anneal_ttop). Далее представлены результаты и сравнение работы каждого из подходов на графах разного размера. Было так же решено использовать результат ответа TTopt для инициализации вершин при запуске квантового отжига. Поэтому в статистиках будут описаны следующие подходы:

- TTopt on edge - TTopt алгоритм, минимизирующий целевую функцию, заданную edge-based подходом.
- TTopt on hop - TTopt алгоритм, минимизирующий целевую функцию, заданную hop-based подходом.
- Edge classic annealing - классический квантовый отжиг для edge-based подхода, случайно инициализирует вершины в начале работы отжига.
- Edge init annealing - квантовый отжиг для edge-based подхода, использующий результат минимизации алгоритма TTopt для edge-based подхода
- Hop classic annealing - классический квантовый отжиг для hop-based подхода, случайно инициализирует вершины в начале работы отжига.
- Hop init annealing - квантовый отжиг для edge-based подхода, использующий результат минимизации алгоритма TTopt для Hop-based подхода

В программе было решено генерировать сетчатые графы следующим образом. Сперва задается точка  $x_i$  на графе с заданной начальной скоростью и задается  $\delta > 0$  - положительная величина, отвечающая за потерю скорости и увеличение стоимости переходов при удалении от заданной точки  $x_i$ . Случайно выбирается заданное число клеток с высокой стоимостью перехода. Ко всем клеткам применяется шум - случайная величина с нормальным распределением.

## 2.1 Малый граф

На первой итерации решено посмотреть, как работают подходы и алгоритмы на малом графе, который имеет вид сетчатого, только без одного ребра. Пример графа на рисунке 2

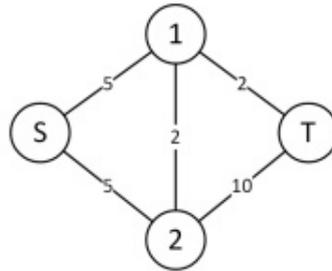


Рис. 2: Малый граф.

В текущей задаче имеем следующие параметры:

- Число переменных в edge-based постановке: 9
- Число переменных в hop-based постановке: 12
- Число обращений к функции у ТТорт:  $3 * 10^5$
- Число запусков каждого алгоритма: 100

Далее сводная таблица для малого графа. В ней представлены следующие параметры:

- Correct path - по выданному результату можно построить правильный путь (начинается в стартовой, завершается в конечной вершинах)
- True shortest - восстановленный путь является кратчайшим
- Admissible path - Является ли путь допустимым. В случае hop based подхода, путь является допустимым, если каждой позиции  $p$  в пути  $P$  соответствует ровно одна вершина графа. Для edge based подхода путь является допустимым, если в качестве ответа выдаются только ребра,

которые лежат на пути (по выданным ребрам можно восстановить путь, но не всегда все выданные ребра вообще лежат на восстановленном пути).

- Equal vertices - набор выданных вершин совпадает с набором вершин, которые лежат на кратчайшем пути (только для Hop Based)

В таблице представлена статистика относительно 100 запусков каждого из алгоритмов.

**Таблица 1:** Сводная таблица для малого графа

	Edge init annealing	Hop init annealing	Edge classic annealing	Hop classic annealing	TTopt on edge	TTopt on hop
Correct path	100%	100%	100%	100%	100%	100%
True shortest	100%	100%	100%	100%	100%	100%
Admissible path	100%	33%	100%	32%	100%	35%
Equal vertices	-	100%	-	100%	-	100%

По результатам, каждый из алгоритмов справился хорошо, хотя hop-based подходы зачастую выдавали не допустимые с точки зрения постановки ответы.

## 2.2 Граф $3 \times 3$

Далее используются сетчатые графы, представленные в параграфе 1.3. Представим результаты работы алгоритмов на графе  $3 \times 3$ . Тесты будут проводиться на разных графах и статистика собирается по результатам работы на всех графах. В текущей задаче имеем следующие параметры:

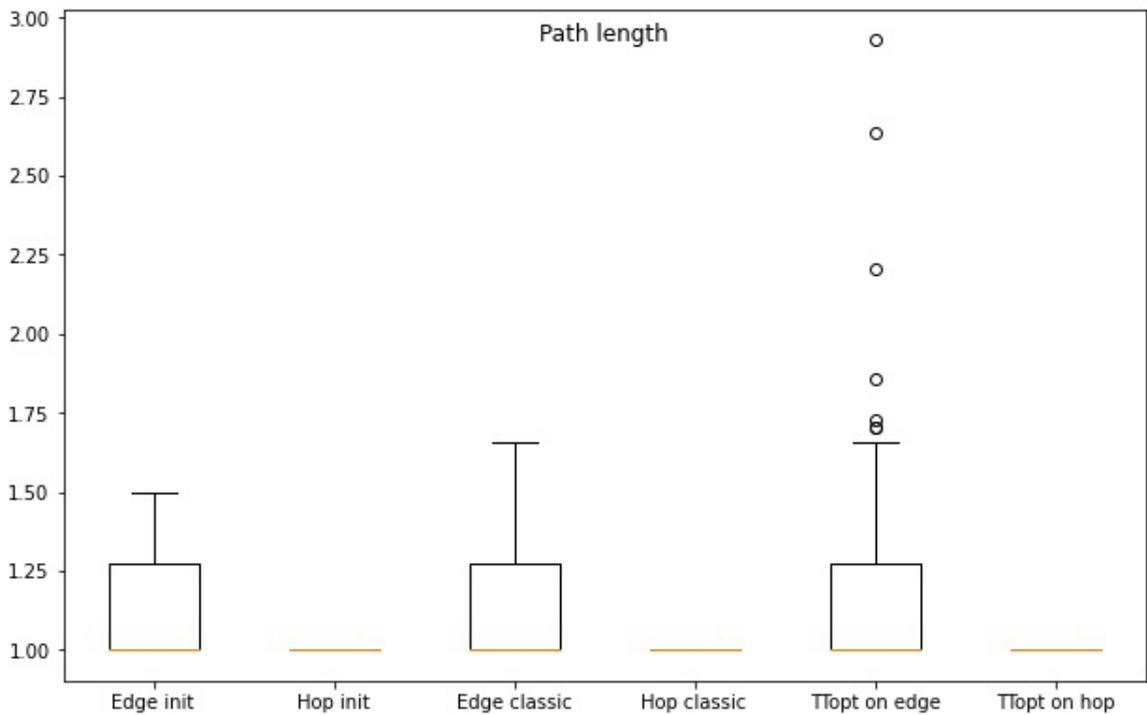
- Число переменных в edge-based постановке: 29
- Число переменных в hop-based постановке: 36
- Число обращений к функции  $y$  TTopt:  $6 * 10^5$
- Число графов: 10
- Число запусков каждого алгоритма на каждом графе: 50

Приведем аналогичную сводную таблицу:

**Таблица 2:** Сводная таблица для графа  $3 \times 3$

	Edge init annealing	Hop init annealing	Edge classic annealing	Hop classic annealing	TTopt on edge	TTopt on hop
Correct path	100.0%	55.4%	100.0%	63.0%	99.0%	73.6%
True shortest	59.4%	55.4%	66.0%	63.0%	67.0%	<b>73.6%</b>
Admissible path	100.0%	39.0%	100.0%	39.8%	99.0%	61.6%
Equal vertices	-	100.0%	-	100.0%	-	100.0%

В текущей задаче, edge-based подходы начинают выдавать не кратчайший путь в качестве ответа. В то же время, hop-based подходы не всегда выдают допустимый путь, но в любом случае, когда путь найден, он является кратчайшим. Алгоритм TTopt для обоих подходов чаще выдавал кратчайший путь, чем квантовый отжиг. На рисунке 3 представлен нормированный разброс найденных путей. В качестве нормализующего множителя берется длина кратчайшего пути для данной задачи, которая честно находится стандартным алгоритмом Дейкстры.



**Рис. 3:** Разброс длины найденных путей. Граф  $3 \times 3$

### 2.3 Граф $5 \times 5$

Представим результаты работы алгоритмов на сетчатом графе  $5 \times 5$ . Аналогично, здесь тесты проведены на разных графах. В текущей задаче имеем следующие параметры:

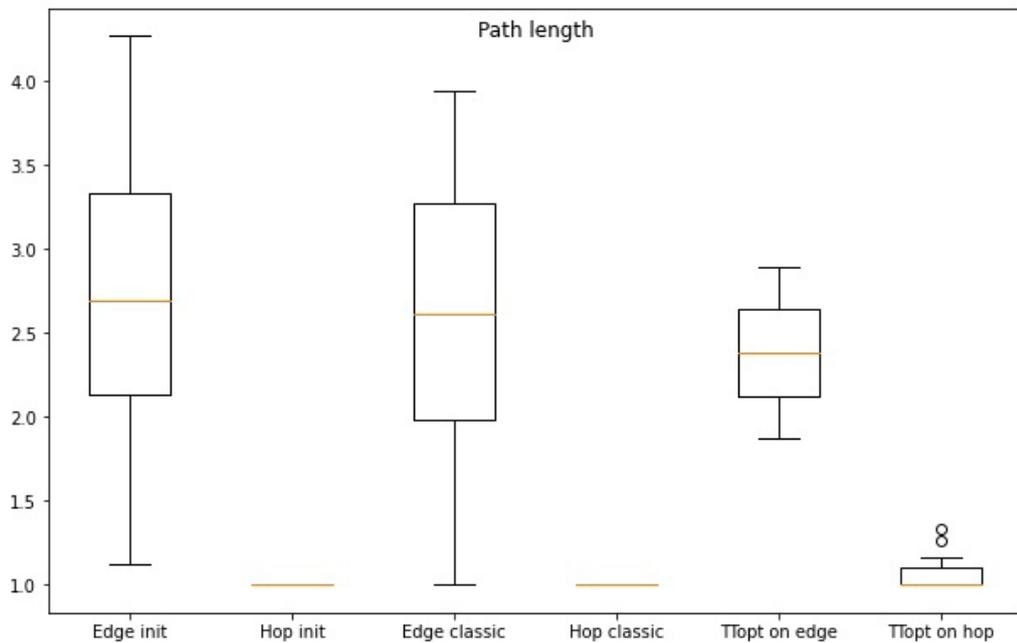
- Число переменных в edge-based постановке: 97
- Число переменных в hop-based постановке: 175
- Число обращений к функции у TTopt:  $2 * 10^6$
- Число графов: 10
- Число запусков каждого алгоритма на каждом графе: 50

Сводная таблица:

**Таблица 3:** Сводная таблица для графа  $5 \times 5$

	Edge init annealing	Hop init annealing	Edge classic annealing	Hop classic annealing	TTopt on edge	TTopt on hop
Correct path	100.0%	66.6%	100.0%	65.0%	3.2%	30.20%
True shortest	0%	<b>66.6%</b>	0.8%	65.0%	0.2%	19.2%
Admissible path	34.4%	45.6%	35.2%	45.6%	2.2%	29.6%
Equal vertices	-	100.0%	-	100.0%	-	19.8%

Заметно, что edge-based подходы начинают работать плохо, причем и при решении задачи в QUBO постановке, и при решении алгоритмом TTopt. TTopt начинает работать хуже из-за ограниченного числа обращений к функции. Квантовый hop-based подход решает задачу так же хорошо, как и на графе  $3 \times 3$ . Разброс длин найденных путей на рисунке 4.



**Рис. 4:** Разброс длины найденных путей. Граф  $5 \times 5$

## 2.4 Граф $7 \times 7$

Последний исследуемый сетчатый граф размера  $7 \times 7$ . Здесь не используется вычисление алгоритмом ТТорт. В предыдущей секции 2.3 отмечено, что ТТорт уже начинает работать плохо, потому что необходимо ограничивать число обращений к функции для разумного времени работы. В графе такого размера использовать ТТорт с теми же параметрами не имеет смысла. В текущей задаче имеем следующие параметры:

- Число переменных в edge-based постановке: 205
- Число переменных в hop-based постановке: 490
- Число графов: 10
- Число запусков каждого алгоритма на каждом графе: 50

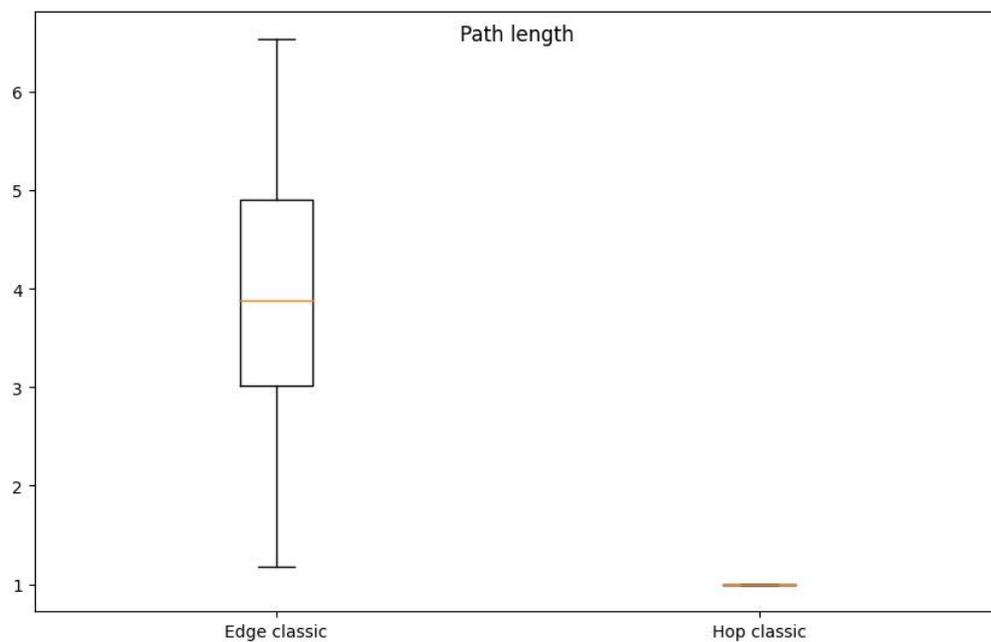
Сводная таблица:

**Таблица 4:** Сводная таблица для графа  $7 \times 7$

	Edge classic annealing	Hop classic annealing
Correct path	100.0%	59.2%
True shortest	0%	<b>59.2%</b>
Admissible path	7.4%	24.4%
Equal vertices	-	100.0%

В текущей задаче совершенно точно видно, что edge-based не только не выдает кратчайший путь, но и сильно упал в проценте допустимых решений. Hop-based почти с такой же вероятностью выдает кратчайший путь.

Более того, длины найденных путей у edge based подхода в разы выше, чем длина кратчайшего пути. Разброс длин найденных путей на рисунке 5.



**Рис. 5:** Разброс длины найденных путей. Граф  $7 \times 7$

## Глава 3. Выводы

По результатам работы, реализованы 2 QUBO подхода для решения задачи поиска кратчайшего пути на графе, проведены сравнения работы данных подходов с помощью квантового отжига и алгоритма TTopt

Если говорить об алгоритмах, TTopt алгоритм хорошо работает на малых графах (для edge-based и hop-based подходов), в том числе, TTopt для hop-based на графе  $3 \times 3$  отработал лучше, чем квантовый отжиг на hop-based подходе. На больших графах, TTopt-у необходимо большое число обращений к функции для качественного поиска экстремума, из-за чего метод квантового отжига для больших графов является более предпочтительным.

Если сравнивать подходы, edge-based тоже хорошо подходит только для решения задачи на малых графах. На графе  $5 \times 5$  все методы решения для edge-based подхода ни разу не нашли кратчайшего пути. На больших графах наилучший результат показывает hop-based подход. Единственный недостаток подхода - он использует сильно больше переменных (кубическая зависимость от размера графа), чем edge-based подход (квадратическая зависимость). Более того, hop-based подход с симуляцией квантового отжига не только не работает хуже с увеличением размера графа, он почти с одной и той же вероятностью возвращает кратчайший путь.

Идея инициализировать квантовый отжиг результатом TTopt к особым результатам не привела, квантовый отжиг с инициализацией работает не лучше, чем квантовый отжиг с рандомной инициализацией переменных. Вероятнее всего, это связано с имплементацией симуляции квантового отжига на обычном компьютере.

Отдельное внимание стоит обратить на метрику Equal vertices в сводных табличках. Во всех случаях, hop based подход на квантовом отжиге всегда выбирает те вершины, которые лежат на кратчайшем пути. Условие правильности пути в постановке hop-based сильно упирается в допустимость, однако, в исследуемых графах восстановить кратчайший путь можно однозначно по выданным вершинам. Таким образом, для графов, с которыми мы работаем, можно всегда восстановить кратчайший путь, используя hop-based подход.

## Список литературы

- [1] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” *Phys. Rev. E*, vol. 58, no. 5, 1998, Art. no. 5355, doi: 10.1103/PhysRevE.58.5355.
- [2] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, “Quantum annealing of a disordered magnet,” *Sci.*, vol. 284, no. 5415, pp. 779–781, 1999, doi: 10.1126/science.284.5415.779.
- [3] A. B. Finnila, M. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, “Quantum annealing: A new method for minimizing multidimensional functions,” *Chem. Phys. Lett.*, vol. 219, no. 5–6, pp. 343–348, 1994, doi: 10.1016/0009-2614(94)00117-0.
- [4] Z. Bian, F. Chudak, W. G. Macready, and G. Rose, “The Ising model: Teaching an old problem new tricks,” 2010.
- [5] K. Sozykin, A. Chertkov, R. Schutski, A.-H. Phan, A. Cichocki, I. Oseledets. TTOpt: A Maximum Volume Quantized Tensor Train-based Optimization and its Application to Reinforcement Learning. arXiv:2205.00293, 2022.
- [6] T. Krauss, J. McCollum. Solving the Network Shortest Path Problem on a Quantum Annealer. DOI: 10.1109/TQE.2020.3021921, 2020.