

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский Нижегородский университет им. Н.И.
Лобачевского»
Кафедра высокопроизводительных вычислений и системного программирования
«Решение задачи о максимальном разрезе графа с
использованием квантовых вычислений»

Научный руководитель:
к.т.н., доцент Мееров И. Б.

Автор работы:
Лапин Дмитрий Алексеевич
студент группы 382003_3 4 курса ННГУ

D-Wave Quantum Annealer



IBM Quantum System





Целью данной работы является рассмотрение и изучение способов решения задачи MaxCut с использованием квантовых вычислений.



В ходе выполнения производственной практики требовалось решить следующие задачи:

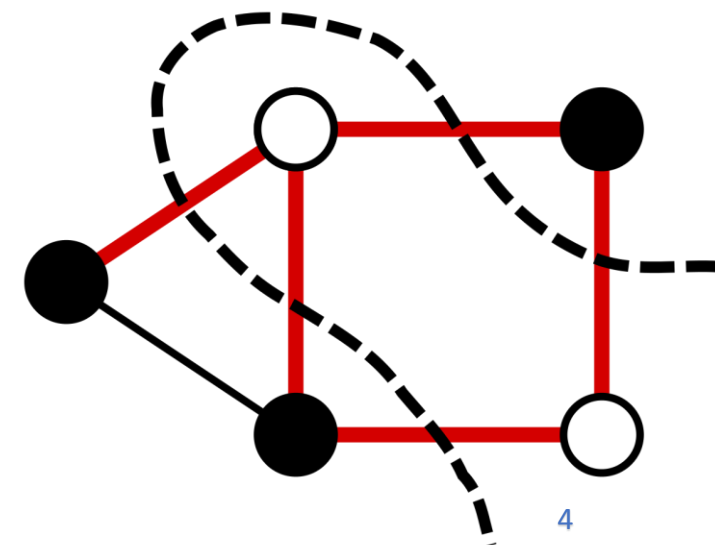
1. Реализовать метод квантового отжига решения задачи MaxCut с использованием D-Wave API.
2. Реализовать алгоритм QAOA решения задачи MaxCut с использованием библиотеки qiskit от IBM.
3. Реализовать свой симулятор идеального квантового компьютера
4. Реализовать алгоритм QAOA решения задачи MaxCut с использованием своего собственного симулятора.
5. Выполнение экспериментальных исследований качества получаемых решений с использованием квантового отжига на квантовом компьютере D-Wave, реализованного ранее симулятора идеального квантового компьютера, симулятора идеального квантового компьютера qiskit.
6. Выполнение экспериментальных исследований производительности разработанной реализации симулятора идеального квантового компьютера.

Задача о максимальном разрезе графа (MaxCut) — одна из самых известных задач комбинаторной оптимизации. Она заключается в поиске разбиения вершин графа на два множества с максимальным количеством ребер между ними. Задача является NP-сложной, и эффективного полиномиального алгоритма для ее решения не существует. $C(x)$ — функция стоимости задачи, которую нужно максимизировать. $x \in \{0, 1\}^n$ — битовая строка длины n , соответствует разбиению графа.

$$\textit{maximize } C(x), \quad x \in \{0, 1\}^n$$

$$\text{MaxCut: } C(x) = \sum_{(i,j) \in E} (x_i + x_j - 2x_i x_j)$$

$$\text{weight MaxCut: } C(x) = \sum_{(i,j) \in E} w_{ij} (x_i + x_j - 2x_i x_j)$$



- Алгоритм Гоеманса-Вильямсона – полиномиальный аппроксимационный алгоритм. Использует полуопределённое программирование и вероятностное округление. Коэффициент аппроксимации $\alpha \approx 0,878$. Реализация из библиотеки `qiskit.optimization`.
- Алгоритм имитации отжига – общий алгоритмический метод решения задачи глобальной оптимизации. Один из методов Монте-Карло. Основан на имитации физического процесса, который происходит при кристаллизации вещества. Реализация из библиотеки D-Wave SDK.
- Полный перебор – перебор всех возможных вариантов решения задачи, экспоненциальная сложность. Реализация из библиотеки D-Wave SDK.
- Квантовый отжиг – это технология оптимизации, использующая эффекты квантового туннелирования и квантовой суперпозиции для увеличения вычислительной производительности.
- Алгоритм QAOA – эвристический квантовый вариационный алгоритм, эмулирующий работу квантового отжига

Уравнение Шредингера: $H|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle$, H – гамильтониан системы

Энергия системы в состоянии $|\psi\rangle$: $E(|\psi\rangle) = \langle\psi|H|\psi\rangle$

Основным состоянием квантовой системы называется состояние $|\psi^*\rangle$ в котором достигается наименьшая энергия системы: $|\psi^*\rangle = \arg \min_{\psi \in \mathcal{H}} E(|\psi\rangle)$

Адиабатическая теорема:

Если процесс адиабатический, то медленное изменение условий позволяет системе подстроить свою конфигурацию, поэтому распределение вероятности меняется во время процесса. Если система в начале была в основном состоянии гамильтониана, она окажется в основном состоянии конечного гамильтониана.

$$H_{Ising} = -A(t) \cdot H_{Initial} + B(t) \cdot H_{Final}$$

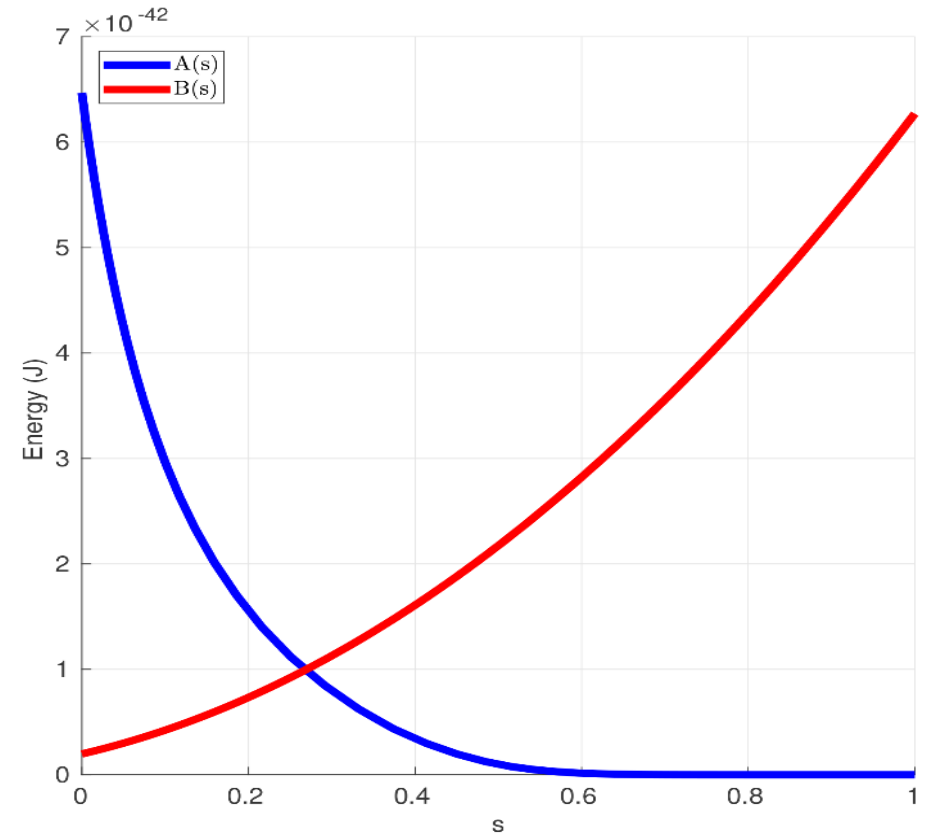
$$A(0) = B(T), \quad A(T) = B(0) = 0$$

$$t \in [0, T]$$

$A(t), B(t)$ – расписание отжига

Гамильтониан $H_{Initial}$ со временем полностью переходит в гамильтониан H_{Final} .

$H_{Initial}$ всегда один и тот же,
 H_{Final} кодирует решаемую проблему.



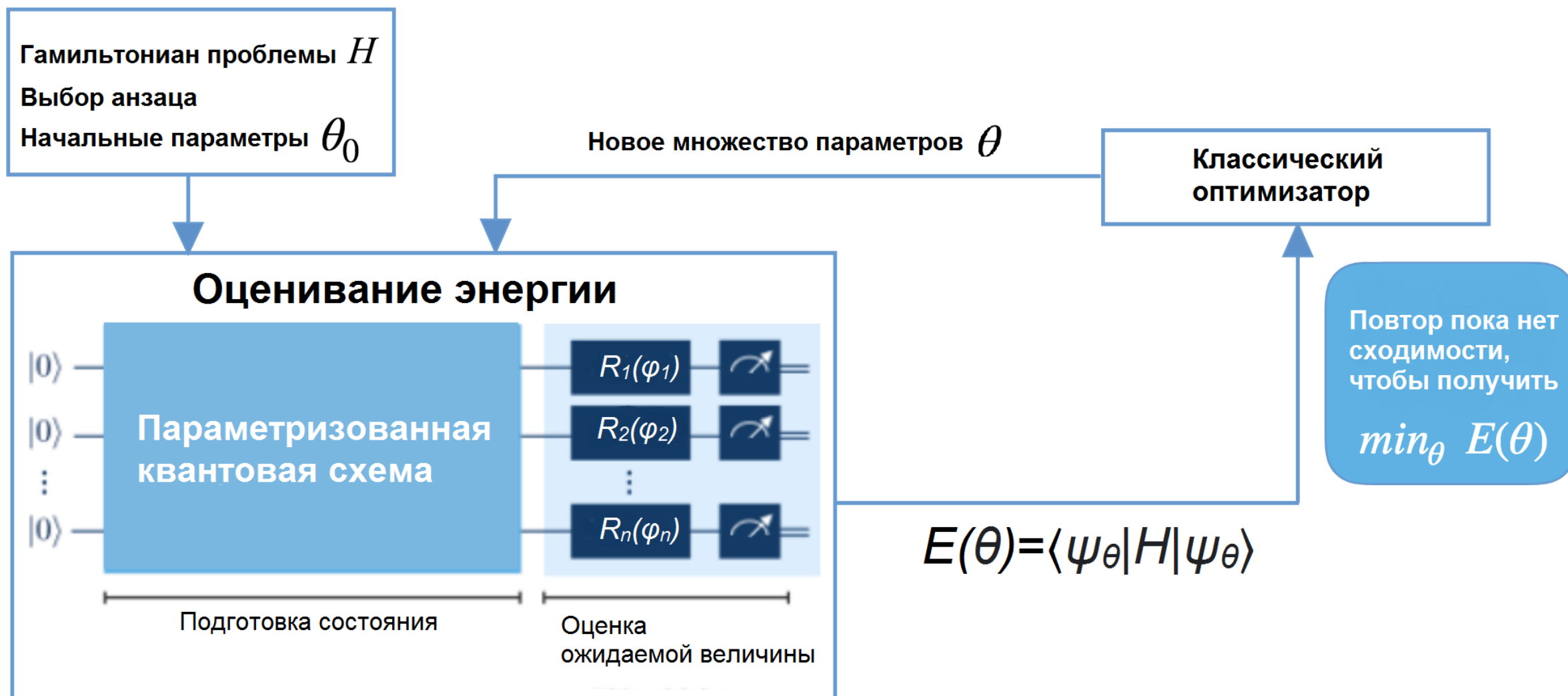
$$H_{Ising} = -A(t) \cdot H_{Initial} + B(t) \cdot H_{Final}$$

$$H_{Initial} = \sum_i X_i, \quad H_{Final} = \sum_i h_i Z_i + \sum_{i,j} J_{i,j} Z_i Z_j$$

$$C(x) = \sum_{(i,j) \in E} w_{ij} (x_i + x_j - 2x_i x_j) \xrightarrow{\{x_i \rightarrow (1-Z_i)/2\}} H = const + \sum_{(i,j) \in E} -\frac{1}{2} w_{ij} Z_i Z_j$$

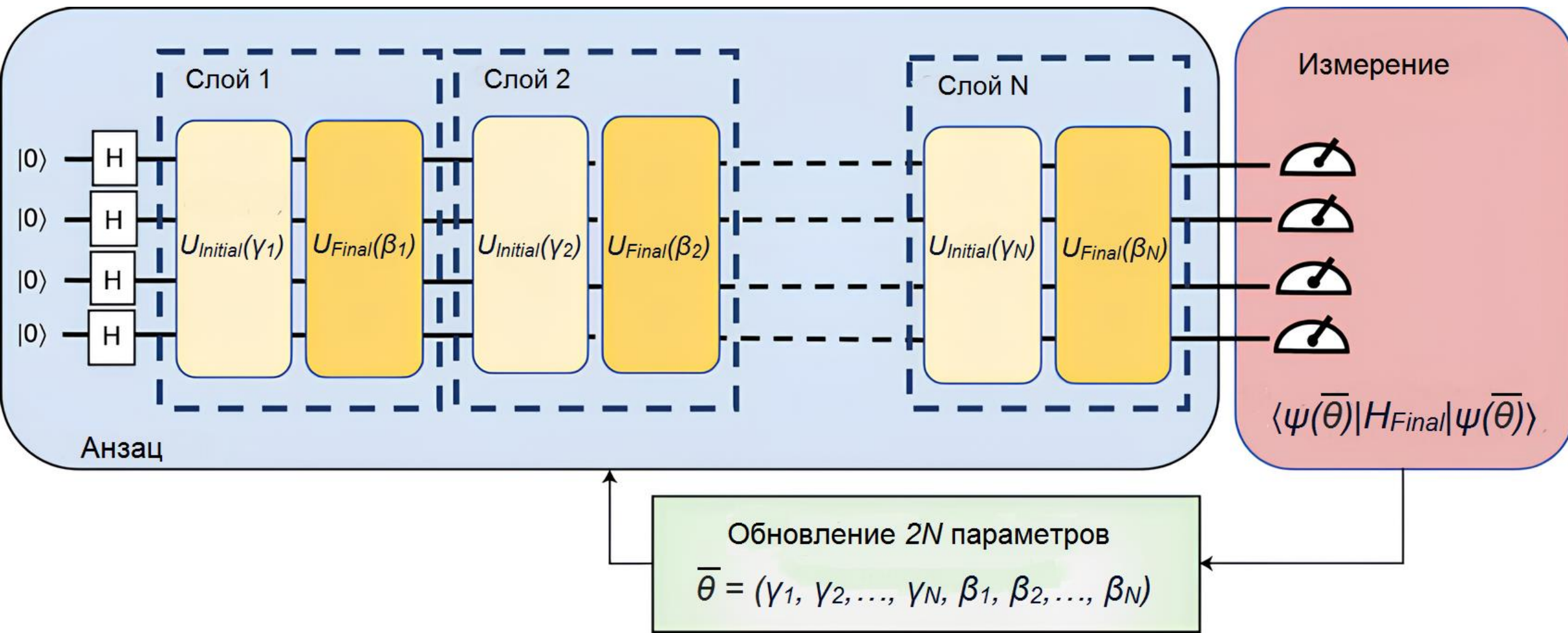
$$H_{Final} = \sum_{(i,j) \in E} w_{ij} Z_i Z_j$$

Variational Quantum Eigensolver



Quantum Approximation Optimization Algorithm

$$U_{Initial}(\gamma) = e^{-i\gamma H_{Initial}}, \quad U_{Final}(\beta) = e^{-i\beta H_{Final}}$$



Характеристики компьютера, на котором выполнялись эксперименты:

CPU: Intel(R) Pentium(R) G4560

RAM: 20,0 Gb

OS: Windows 7 Professional

Все эксперименты проводились в системе Linux Ubuntu 20.04.6 LTS, запущенной с помощью VirtualBox с 10 Gb выделенной оперативной памяти и 4-мя ядрами. В качестве интерпретатора Python взята версия Python 3.8.10

Выполняется решение задачи MaxCut следующими методами:

- полный перебор
- алгоритм Гоеманса-Вильямсона
- классический отжиг
- квантовый отжиг
- QAOA на qiskit

Методы применяются для графов размером 5, 10, 15, 20, 30, 45, 60, 90, 180, 360.

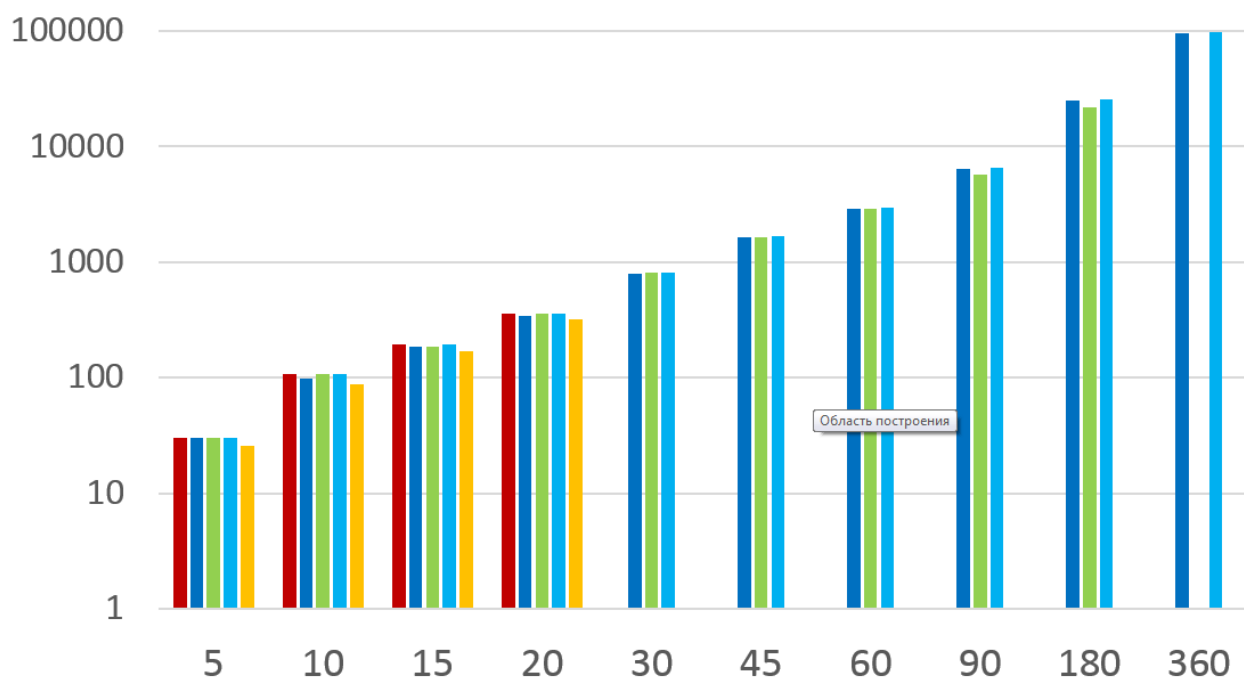
Измеряется время работы каждого алгоритма, сохраняется значение функции стоимости, которое соответствует полученному решению.

Графы для экспериментов сгенерированы при помощи модели Эрдеша-Реньи. Вероятность существования каждого ребра между любой парой вершин взята равной 0.5.

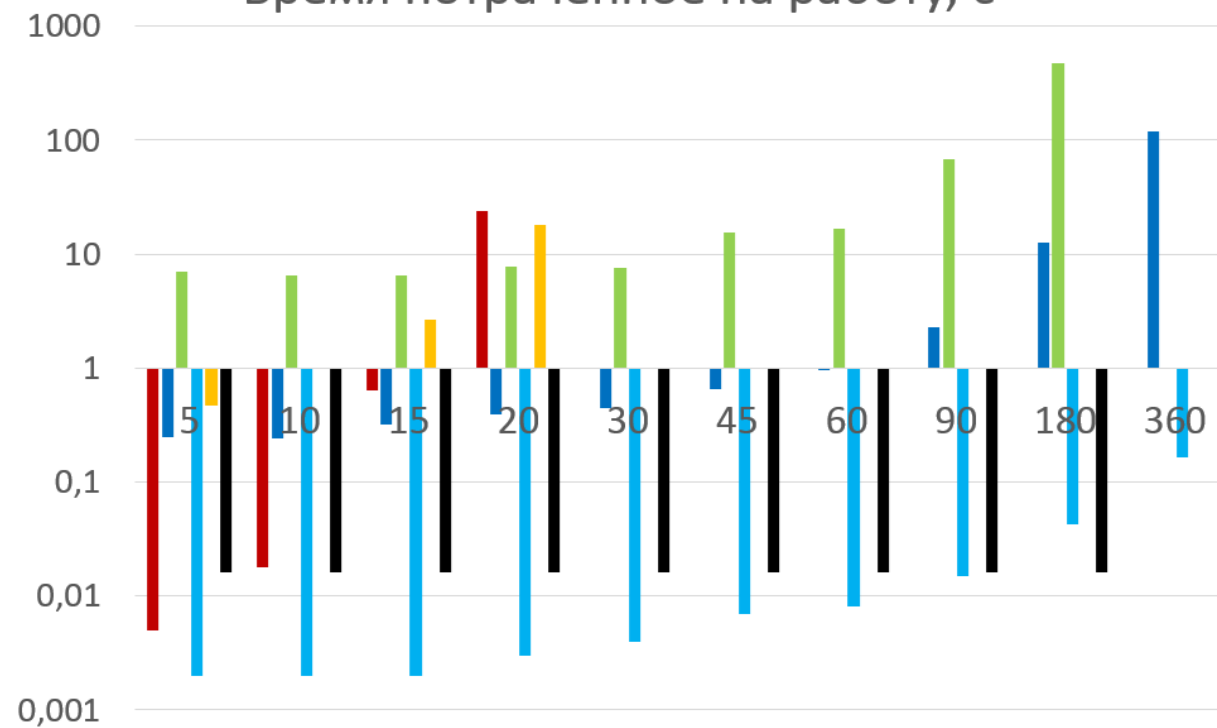
Результаты экспериментов

- Перебор
- Квантовый отжиг(всего)
- Симуляция QAOA
- Алгоритм Гоеманса-Вильямсона
- Классический отжиг
- Квантовый отжиг(кв. часть)

Стоимость найденного решения



Время потраченное на работу, с

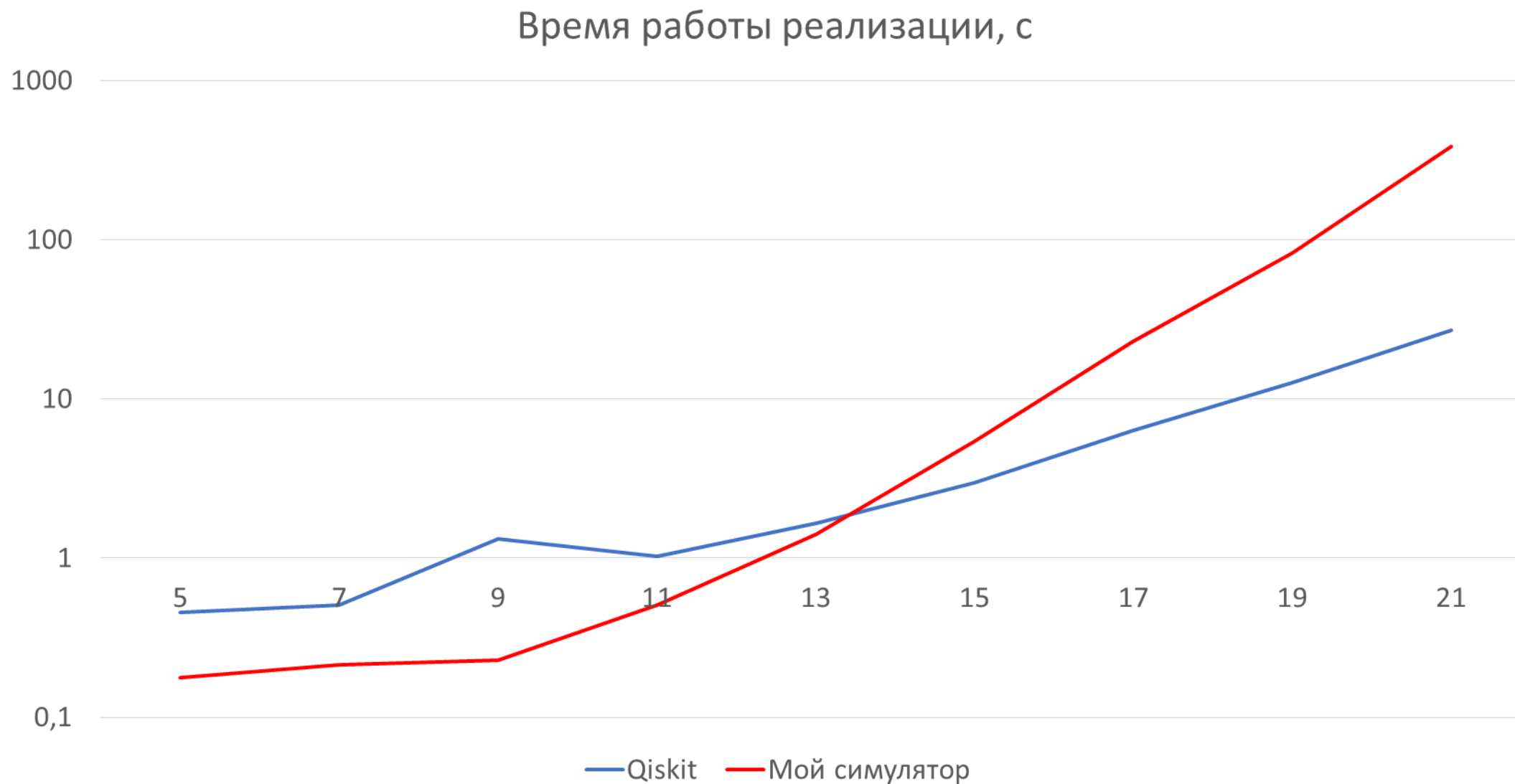


Цель: исследовать производительность работы моего симулятора на примере реализации алгоритма QAOA и сравнить с производительностью реализации QAOA на qiskit.

Описание:

Эксперименты проводятся над графами размером 5, 7, 9, 11, 13, 15, 17, 19, 21. Измеряется время работы каждой реализации. Графы для экспериментов сгенерированы при помощи модели Эрдеша-Реньи.

Вероятность существования каждого ребра между любой парой вершин взята равной 0.5. У алгоритма QAOA заданное количество слоев равно 1, в качестве классического оптимизатора взят алгоритм COBYLA. На каждой итерации алгоритма количество шотов у квантовой схемы равно 10000. В качестве начальных параметров анзаца взят вектор [1.0, 1.0].



Таким образом, были изучены различные механизмы работы и применения квантовых вычислений. В ходе практики был получен опыт работы с реальным квантовым компьютером D-Wave с использованием облачной платформы D-Wave Leap. Так же был получен опыт работы с различными квантовыми алгоритмами и их симуляции с использованием классического компьютера.

1. Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm. arXiv:1411.4028. 2014 Nov 14.
2. D-Wave: What is Quantum Annealing? URL: https://docs.dwavesys.com/docs/latest/c_gs_2.html (Дата обращения: 17.04.24)
3. Duke University: Maximum cut. URL: https://courses.cs.duke.edu/cps232/fall15/scribe_notes/lec17.pdf (Дата обращения: 16.06.2024)
4. Goemans M. X., Williamson D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming //Journal of the ACM (JACM). – 1995. – Т. 42. – №. 6. – С. 1115-1145.
5. University of Cambridge: Adiabatic Quantum Computing. URL: https://www.cl.cam.ac.uk/teaching/1920/QuantComp/Quantum_Computing_Lecture_15.pdf (Дата обращения: 17.06.24)
6. Stanford University: The Ising Model. URL: <https://stanford.edu/~jeffjar/statmech/intro4.html> (Дата обращения: 17.06.24)
7. Glover F., Kochenberger G., Du Y. Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models //4or. – 2019. – Т. 17. – №. 4. – С. 335-371.
8. Tilly J. et al. The variational quantum eigensolver: a review of methods and best practices //Physics Reports. – 2022. – Т. 986. – С. 1-128.
9. The Santa Cruz Institute for Particle Physics: The time evolution operator as a time-ordered exponential. URL: <http://scipp.ucsc.edu/~haber/ph215/TimeOrderedExp.pdf> (Дата обращения: 22.06.24)

Спасибо за внимание!

$H(t)|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle$ – уравнение Шредингера

$|\psi(T)\rangle = \mathcal{T} e^{-\frac{i}{\hbar} \int_0^T H(t) dt} |\psi(0)\rangle$, \mathcal{T} – оператор упорядочивания во времени

$\int_0^T H(t) dt \approx \sum_{n=0}^N H(n\Delta t) \Delta t$, где $\Delta t = \frac{T}{N}$ – дискретизация

$e^{(A+B)\Delta t} \approx e^{A\Delta t} e^{B\Delta t}$ – формула Троттер-Сузуки

$H_{Ising} = -A(t) \cdot H_{Initial} + B(t) \cdot H_{Cost}$ – гамильтониан Изинга

$$\mathcal{T} e^{-\frac{i}{\hbar} \int_0^T H(t) dt} \approx \prod_{n=0}^N e^{-i \frac{A(n\Delta t)\Delta t}{\hbar} H_{Initial}} e^{-i \frac{B(n\Delta t)\Delta t}{\hbar} H_{Cost}}$$

$$|\psi(T)\rangle \approx \prod_{n=0}^N e^{-i\frac{A(n\Delta t)\Delta t}{h}H_{Initial}} e^{-i\frac{B(n\Delta t)\Delta t}{h}H_{Cost}} |\psi(0)\rangle$$

Расписание отжига:

$$\frac{-A(n\Delta t)\Delta t}{h} \rightarrow \gamma_1, \gamma_2, \dots, \gamma_N$$

$$\frac{B(n\Delta t)\Delta t}{h} \rightarrow \beta_1, \beta_2, \dots, \beta_N$$

$$|\psi_{Final}\rangle \approx e^{-i\gamma_1 H_{Initial}} e^{-i\beta_1 H_{Cost}} \dots e^{-i\gamma_N H_{Initial}} e^{-i\beta_N H_{Cost}} |\psi(0)\rangle$$

$\arg \min_{\gamma_1, \gamma_2, \dots, \gamma_N, \beta_1, \beta_2, \dots, \beta_N} \langle \psi_{Final} | H_{Cost} | \psi_{Final} \rangle$ – задача оптимизации расписания отжига

$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = U(\vec{\gamma}, \vec{\beta})|\psi_0\rangle$$

Анзац полученной квантовой схемы:

$$U(\vec{\gamma}, \vec{\beta}) = U_{phase}(\gamma_1)U_{mixer}(\beta_1)U_{phase}(\gamma_2)U_{mixer}(\beta_2) \dots U_{phase}(\gamma_N)U_{mixer}(\beta_N)$$

$$U_{phase}(\gamma) = e^{-i\gamma H_{Initial}}$$

$$U_{mixer}(\beta) = e^{-i\beta H_{Cost}}$$

$$H_{Initial} = \sum_i X_i$$

Стартовое состояние системы:

$$|\psi_0\rangle = |+\rangle^n = \sum_{x \in \{0;1\}^n} \frac{1}{\sqrt{2^n}} |x\rangle$$

$$U_{mixer}(\beta) = e^{-i\beta H_{Cost}} = e^{-i\beta \sum_{(i,j) \in E} w_{ij} Z_i Z_j} = \prod_{(i,j) \in E} e^{-i\beta w_{ij} Z_i Z_j} = \prod_{(i,j) \in E} RZZ(i, j, \beta w_{ij})$$

$$U_{phase}(\gamma) = e^{-i\gamma \sum_i X_i} = e^{-i\gamma H_{Initial}} = \prod_i e^{-i\gamma X_i} = \prod_i RX(\gamma)$$

$|\langle x | \psi(\vec{\gamma}, \vec{\beta}) \rangle|^2$, $x \in \{0,1\}^n$ - полученное распределение вероятностей

$$e = \langle \psi(\vec{\gamma}, \vec{\beta}) | H_{Cost} | \psi(\vec{\gamma}, \vec{\beta}) \rangle = \sum_{x \in \{0,1\}^n} C(x) |\langle x | \psi(\vec{\gamma}, \vec{\beta}) \rangle|^2$$

QUBO – quadratic unconstrained binary optimization

$$f(\mathbf{x}) = \mathbf{x}^\top Q \mathbf{x} = \sum_{i,j=1}^n Q_{ij} x_i x_j, Q \in$$

$$\mathbb{R}^{n \times n}, \mathbf{x} \in \{0, 1\}^n$$

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \{0, 1\}^n} f(\mathbf{x})$$