

Санкт-Петербургский государственный университет

**ДУРОВА Анастасия Вячеславовна**

**Выпускная квалификационная работа**

**Расчеты модели Изинга и электронной структуры атомов и молекул с  
помощью квантовых алгоритмов**

Уровень образования: *магистратура*

Направление *03.04.02 «Физика»*

Основная образовательная программа *ВМ.5511.2021 «Физика»*

Научный руководитель:

доцент,

кафедра квантовой механики,

к.ф.-м.н., без у/з,

Мальцев Илья Александрович

Рецензент:

научный сотрудник,

НИЦ «Курчатовский институт» – ПИЯФ,

к.ф.-м.н.,

Суясова Марина Вадимовна

Санкт-Петербург

2023 год

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Задача логистики</b>	<b>4</b>
2.1	Маршрутная формулировка . . . . .	4
2.2	Последовательная формулировка . . . . .	6
2.3	Перевод формулировок задачи логистики к задачам класса квадратичной бинарной оптимизации без ограничений . . . . .	7
2.4	Quantum Approximate Optimization Algorithm . . . . .	9
<b>3</b>	<b>Расчет энергии ионизации</b>	<b>10</b>
3.1	Основной формализм . . . . .	10
3.2	Variational Quantum Eigensolver . . . . .	12
3.3	Алгоритм расчета энергии ионизации . . . . .	13
<b>4</b>	<b>Параметризация состояния</b>	<b>15</b>
<b>5</b>	<b>Вычисление градиентов на квантовом компьютере</b>	<b>16</b>
<b>6</b>	<b>Результаты</b>	<b>18</b>
6.1	Задача логистики с временными окнами . . . . .	18
6.2	Задача логистики без временных окон . . . . .	19
6.3	Результаты вычисления энергии ионизации . . . . .	22
<b>7</b>	<b>Выводы</b>	<b>24</b>
7.1	Задача логистики буровых установок . . . . .	24
7.2	Расчет энергии ионизации . . . . .	24

# 1 Введение

На сегодняшний день квантовые вычисления являются быстро развивающейся областью исследований. Квантовый компьютер – это вычислительное устройство, которое использует эффекты квантовой механики (квантовая суперпозиция, квантовая запутанность) для реализации вычислений. В ряде случаев квантовые расчеты имеют экспоненциальное преимущество перед классическими [1].

Единицей информации квантового компьютера является квантовый бит (кубит), который описывается вектором в гильбертовом пространстве и может быть представлен в виде суперпозиции  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , где  $\{|0\rangle, |1\rangle\}$  – это некоторый базис и  $|\alpha|^2 + |\beta|^2 = 1$ . Состояние кубита преобразуется с помощью квантовых гейтов, которые одновременно изменяют всю суперпозицию, обеспечивая ограниченный измерениями параллелизм. Алгоритмы на квантовых компьютерах реализуются с помощью квантовых цепей, которые являются последовательностью квантовых гейтов. Кубит находится в суперпозиции на протяжении всего процесса вычислений, однако после измерения переходит в состояние  $|0\rangle$  с вероятностью  $|\alpha|^2$  или состояние  $|1\rangle$  с вероятностью  $|\beta|^2$ . Система из  $n$  кубитов может быть представлена в виде суперпозиции  $2^n$  базисных векторов.

Манин [2] и Фейнман [3] впервые предложили использовать квантовый компьютер для расчета квантовых систем. Основная мысль заключалась в том, чтобы построить управляемую квантовую систему и использовать ее для расчета многочастичных квантовых систем. Возможности для расчета таких систем на классических компьютерах ограничены из-за экспоненциального роста сложности волновой функции с увеличением числа частиц. При этом такие расчеты необходимы для теоретического описания электронной структуры атомов и молекул. Знание электронной структуры в свою очередь необходимо для понимания свойств атомов и химических соединений.

В ходе развития квантовых вычислений были найдены новые применения возможностей квантового компьютера. Так, в 1997 году Питер Шор предложил алгоритм факторизации чисел на простые множители [4]. Также со временем все большее внимание стали привлекать задачи комбинаторной оптимизации [5], решение которых может быть эффективно улучшено с помощью квантового компьютера [6]. Целью комбинаторных задач является поиск оптимального решения в рамках фиксированного набора ограничений. Одним из их приложений является задача логистики, которая представляет собой минимизацию расстояния при перемещении по заданной карте. В данной задаче существует множество подходов к решению (формулировок), влияющих на число переменных, вид ограничений и точность получаемых результатов.

На сегодняшний день, несмотря на значительный прогресс в разработке квантовых вычислительных устройств их возможности ограничены. Это происходит из-за того, что они подвержены шумам вследствие декогеренции и неточности выполнения гейтов. Одним из наиболее перспективных подходов к решению задач на таких устройствах являются вариационные квантовые алгоритмы (VQA), использующие классический компьютер наряду с квантовым. Считается, что VQA устойчивы к шумам различного происхождения за счет своей гибридной природы. Однако они являются эвристическими, в силу чего не ясно, смогут ли VQA превзойти результаты решения задач на классическом компьютере.

Среди вариационных подходов можно выделить два: Variational Quantum Eigensolver (VQE) [7] и Quantum Approximate Optimization Algorithm (QAOA) [8]. Выбор алгоритма определяется задачей, а их главное отличие заключается в подготовке начального состояния (анзаца). Гибридный алгоритм VQE используется для нахождения собственных значе-

ний гамильтониана задачи. Результаты реализации Variational Quantum Eigensolver зависят от сложности квантовых цепей и классической оптимизационной процедуры. То есть, выбор оптимизатора и вариационного анзаца, используемого для параметризации начального состояния, влияет на точность результатов алгоритма VQE. Quantum Approximate Optimization Algorithm полностью наследует структуру Variational Quantum Eigensolver. Однако в отличие от VQE, анзац которого мы можем менять, параметризация состояния в QAOA фиксирована и содержится в структуре алгоритма.

В расчетах квантовой химии зачастую рассматриваются задачи не вычисления энергии основного состояния, а расчет энергии ионизации молекулярной системы. Последовательная оценка энергий  $N$  и  $N-1$  частиц с помощью VQE и нахождение их разности может привести к накоплению ошибок квантового компьютера. Чтобы повысить точность расчетов, предлагается вычислять энергию ионизации сразу с помощью специально разработанного алгоритма. Он так же, как и предыдущие, является гибридным и включает в себя классическую оптимизационную процедуру, однако возвращает разность двух средних значений гамильтониана.

В данной работе задача логистики буровых установок описана в разделе 2. Теоретическим методам, используемым для расчета энергии ионизации посвящен раздел 3.3. Описание вариационных квантовых алгоритмов VQE и QAOA содержится в разделах 3.2 и 2.4 соответственно. Пункт 4 включает в себя подробное описание анзаца, используемого при реализации VQE. Метод расчета градиентов, необходимых для реализации оптимизационных процедур, представлен в разделе 5. Параграф 6 содержит результаты расчетов.

## 2 Задача логистики

Для начала введем некоторые определения, значимые для унификации дальнейшего повествования:

- Агент – человек, автомобиль или любое другое средство, которое непосредственно производит перемещение по точкам интереса (вершинам). В нашем случае агентами являются буровые установки.
- Узлы – точки интереса, по которым производится перемещение агентов. В задаче логистики буровых установок перемещение осуществляется по кустам бурения.

Здесь и далее будем называть узлы кустами, а агентов буровыми установками. В задачах логистики выделяется два типа кустов – активные и ”депо”. Куст ”депо” является местом отправления и назначения для всех буровых установок.

Чтобы сформулировать задачу логистики буровых установок, необходимы следующие данные:

- Матрица расстояний  $d$  – матрица, в которой содержатся расстояния между всеми кустами в задаче. Элемент матрицы  $d_{ij}$  соответствует расстоянию между кустом  $i$  и кустом  $j$ .
- Матрица времен  $t$  – матрица, в которой содержатся времена переезда между всеми кустами в задаче.
- Матрица цен  $c$  – матрица, в которой содержится цена переезда между всеми кустами в задаче. В подавляющем большинстве задач полагается  $c_{ij} = d_{ij}$  или  $c_{ij} = t_{ij}$ , но в общем случае данные предположения не имеют значения. В наших расчетах будем полагать  $c_{ij} = d_{ij}$ .

Заметим, что требований по симметричности этих матриц нет, т.е.  $d_{ij}$  может быть не равно  $d_{ji}$ . Такое может случиться, например, в случае наличия дорог с односторонним движением на пути.

Задача логистики буровых установок принадлежит классу задач бинарной оптимизации [5] и заключается в минимизации стоимости, затрачиваемой на переезды буровых установок между кустами при бурении. Для конкретных приложений необходимо разработать точные подходы, математику и метаэвристику с целью получения решений допустимой точности при относительно небольших вычислительных затратах. Поэтому задача логистики имеет различные математические формулировки, отличающиеся размерностью и ограничениями на переезд от одного куста к другому. В данной работе представлены две возможные формулировки для задачи логистики буровых установок.

### 2.1 Маршрутная формулировка

Маршрутная формулировка представляет собой перебор всех возможных путей. В данных терминах путь – это последовательность узлов  $(i_1, i_2, \dots, i_P)$  с заданными ограничениями: путь начинается и заканчивается в депо, при этом каждый его элемент должен принадлежать множеству допустимых путей  $(i_p, i_{p+1}) \in \mathcal{A} \forall i_p, i_{p+1} \in \mathcal{N}$ , где  $\mathcal{N}$  – это множество кустов. Часто в данной задаче вводятся временные окна, накладывающие дополнительные ограничения: время прибытия на куст  $i_p$  должно быть больше времени открытия куста  $a_p$ , но меньше времени закрытия куста  $b_p \forall p = 1, \dots, P$ . Если положить время

прибытия на первый куст  $T_{i_1}$  равным 0, то эффективное время прибытия на куст  $i_{p+1}$   $T_{i_{p+1}} = \max\{a_{i_{p+1}}, T_{i_p} + t_{i_p, i_{p+1}}\}$ , где  $t_{i_p, i_{p+1}}$  – время перемещения между кустами  $i_p$  и  $i_{p+1}$ ,  $T_{i_p}$  – эффективное время прибытия на куст  $i_p$ . Тогда  $T_{i_p} \leq b_{i_p} \forall i_p \in \mathcal{N}$ .

При рассмотрении задачи логистики буровых установок вводятся бинарные переменные  $x_r = 1$ , если вышка движется по маршруту  $r$ , в противном случае  $x_r = 0$ . Такие переменные образуют набор  $\mathbf{x} = \{x_r\}_{r \in \mathcal{R}}$ , где  $\mathcal{R}$  – множество всех путей, удовлетворяющих условиям задачи. Пусть  $V$  – это число буровых установок,  $c_r$  – это суммарная стоимость перемещения вышки между всеми кустами, включенными в маршрут  $r \in \mathcal{R}$  или длина маршрута  $r \in \mathcal{R}$ , а величина  $\delta_{i,r} = 1$ , если путь  $r$  посещен вышкой  $i \in \mathcal{N}$ , в противном случае  $\delta_{i,r} = 0$ , тогда задача имеет вид

$$\min_{\mathbf{x}} \sum_{r \in \mathcal{R}} c_r x_r, \quad (1)$$

$$\sum_{r \in \mathcal{R}} \delta_{i,r} x_r = 1, \quad \forall i \in \mathcal{N}, \quad (2)$$

$$\sum_{r \in \mathcal{R}} x_r = V, \quad (3)$$

$$x_r \in \{0, 1\}, \quad \forall r \in \mathcal{R}. \quad (4)$$

Ограничение на то, что каждый куст должен быть посещен один раз задается соотношением (2), а число рассматриваемых вышек фиксируется при помощи равенства (3).

Количество переменных маршрутной формулировки соответствует числу маршрутов, составляемых предварительно с учетом особенностей задачи. В общем случае, их количество может быть очень большим. Например, если перемещение из каждого куста в любой другой куст возможно, то количество маршрутов  $|\mathcal{R}|$  равно  $|\mathcal{N}|!$ , где  $|\mathcal{N}|$  – это количество кустов. Так, в задаче коммивояжера (Travelling Salesperson Problem (TSP)), где число буровых установок равно единице, содержится  $|\mathcal{N}|!$  маршрутов. Задачи о логистике нескольких буровых установок (Vehicle Routing Problem (VRP)) должны также учитывать все возможные маршруты разной длины. Для маршрута длины  $P$  есть  $P!$  различных путей, которые могут существовать в задаче VRP. Таким образом, общее число маршрутов в формулировке VRP

$$M_{\text{route}} = |\mathcal{R}| = \sum_{p=1}^N p! = \mathcal{O}(N!). \quad (5)$$

Такое количество переменных слишком велико как для универсального квантового компьютера, так и для солвера, созданного под решение определенного класса задач. В связи с этим, маршрутная формулировка в исходном виде почти не масштабируема на реальные задачи. Ограничения задачи могут значительно снизить количество допустимых путей. Например, наличие временных окон в идеальном случае приводит к виду, где на каждом шаге при генерации маршрута, любая буровая установка может поехать в  $K$  кустов, осуществляя при этом  $N/K$  шагов. Тогда, количество переменных будет выражаться формулой

$$M_{\text{route}} = |\mathcal{R}| = K \times \left(K^{\frac{N}{K}}\right) = \mathcal{O}\left(K^{\frac{N+K}{K}}\right). \quad (6)$$

Данное количество бинарных переменных все еще достаточно велико, но уже существенно меньше, чем в формулировке VRP без временных окон. Таким образом, количество переменных в маршрутной формулировке с временными окнами можно оценить согласно следующей формуле

$$\mathcal{O}\left(K^{\frac{N+K}{K}}\right) \leq M_{\text{route}} \leq \mathcal{O}(N!). \quad (7)$$

## 2.2 Последовательная формулировка

Формулировка на основе последовательностей не является линейной в отличие от маршрутной. В последовательной формулировке переменная  $x_{v,p,i} = 1$ , если буровая установка  $v \in \mathcal{V}$  посещает куст  $i \in \mathcal{N}$  с позицией  $p \in \{1, \dots, P\}$  в последовательности кустов в маршруте, в противном случае  $x_{v,p,i} = 0$ . Задача оптимизации заключается в поиске наименьшего суммарного расстояния на переезд из куста  $i$  с позицией  $p$  на куст  $j$  с позицией  $p + 1$  для каждой из вышек  $v$  и имеет вид

$$\min_{\mathbf{x}} \sum_v \sum_{p < P} \sum_{(i,j) \in \mathcal{A}} c_{i,j} x_{v,p,i} x_{v,p+1,j}, \quad (8)$$

$$\sum_{v \in \mathcal{V}} \sum_{p=1}^P x_{v,p,i} = 1, \quad \forall i \in \mathcal{N}, \quad (9)$$

$$\sum_{i \in \mathcal{N} \cup \{d\}} x_{v,p,i} = 1, \quad \forall v \in \mathcal{V}, \quad p \in \{1, \dots, P\}, \quad (10)$$

$$x_{v,p,i} x_{v,p+1,j} = 1, \quad \forall v \in \mathcal{V}, \quad p \in \{1, \dots, P-1\}, \quad (i, j) \notin \mathcal{A}, \quad (11)$$

$$x_{v,p,d} x_{v,p+1,j} = 0, \quad \forall v \in \mathcal{V}, \quad p \in \{2, \dots, P-1\}, \quad j \neq d, \quad (12)$$

$$x_{v,1,d} = 1, \quad \forall v \in \mathcal{V}, \quad (13)$$

$$x_{v,P,d} = 1, \quad \forall v \in \mathcal{V}, \quad (14)$$

$$x_{v,1,i} = 0, \quad \forall v \in \mathcal{V}, \quad i \in \mathcal{N}, \quad (15)$$

$$x_{v,P,i} = 0, \quad \forall v \in \mathcal{V}, \quad i \in \mathcal{N}, \quad (16)$$

$$x_{v,2,j} = 0, \quad \forall v \in \mathcal{V}, \quad (d, j) \notin \mathcal{A}, \quad (17)$$

$$x_{v,P-1,j} = 0, \quad \forall v \in \mathcal{V}, \quad (d, j) \notin \mathcal{A}, \quad (18)$$

$$x_{v,p,i} \in \{0, 1\}, \quad \forall (v, p, i). \quad (19)$$

Соответствующие ограничения задачи обеспечивают, что каждый куст за исключением депо посещается один раз (9), для каждой буровой установки куст, занимающий позицию  $p$  встречается один раз (10), посещаются только разрешенные условиями задачи кусты (11), все пути начинаются и заканчиваются в депо (12), (13), (14). Ограничения (15)-(19) являются следствием предыдущих.

В этой формулировке можно учесть временные окна, удаляя дуги из графа, которые не удовлетворяют им. В частности, если верхняя граница временного окна куста  $i$  в сумме с временем перемещения между вышками  $i, j$  больше, чем верхняя граница временного окна куста  $j$ , то эта дуга удаляется из множества допустимых дуг  $\mathcal{A}$ .

Последовательная формулировка оперирует переменными  $x_{vpi}$ . Для определения маршрута одной буровой установки необходимо  $P \times N$  переменных, где  $P$  – число позиций последовательности кустов в маршруте. Очевидно, что  $P \leq N$ . С другой стороны, в наилучшем

случае длина маршрута для каждого из агентов составляет  $N/V$ . Таким образом, оценить количество переменных в последовательной формулировке можно следующим образом:

$$N^2 \leq M_{\text{sequence}} \leq V \times N^2 \quad (20)$$

## 2.3 Перевод формулировок задачи логистики к задачам класса квадратичной бинарной оптимизации без ограничений

Задачи квадратичной бинарной оптимизации формулируются следующим образом

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^\top M \mathbf{x} + c, \\ & \mathbf{x} \in \{0, 1\}^n, \end{aligned} \quad (21)$$

где  $M$  – квадратная вещественная матрица,  $c$  – вещественная константа. Рассмотрим преобразование формулировок логистической задачи 2.1, 2.2 к виду (21).

Маршрутная формулировка является линейной по вектору бинарных переменных  $\mathbf{x} \in \{0, 1\}^n$  и может быть представлена в виде

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x}, \\ & A \mathbf{x} = \mathbf{b}, \end{aligned} \quad (22)$$

где  $A$ ,  $\mathbf{b}$  – матрица и вектор с вещественными коэффициентами, задающие ограничения задачи,  $\mathbf{c}$  – вектор весов (расстояний). В задачах квадратичной бинарной оптимизации без ограничений невозможно задавать условия для значений вектора переменных  $\mathbf{x}$ . Они могут быть внесены в целевую функцию в виде слагаемого, которое равно нулю при выполнении условий и дает большой вклад в противном случае. То есть, (22) можно переписать как

$$\min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x} + \rho \cdot \|A \mathbf{x} - \mathbf{b}\|^2, \quad (23)$$

где  $\rho$  – достаточно большое положительное число, называемое штрафным параметром. Принято выбирать  $\rho \gg \|\mathbf{c}\|$  для наиболее строгого учета ограничений.

Чтобы построить матрицу  $M$  задачи (21) необходимо преобразовать исходную формулировку к квадратичному виду. Если учесть, что  $x_i \in \{0, 1\}$ ,  $x_i = x_i^2$ , то можно осуществить замену

$$\begin{aligned} \mathbf{c}^\top \mathbf{x} &\rightarrow \mathbf{x}^\top \mathbf{Diag}(\mathbf{c}) \mathbf{x}, \\ \|A \mathbf{x} - \mathbf{b}\|^2 &= \mathbf{x}^\top A^\top A \mathbf{x} - 2\mathbf{b}^\top A \mathbf{x} + \mathbf{b}^\top \mathbf{b} \rightarrow \mathbf{x}^\top A^\top A \mathbf{x} - \mathbf{x}^\top \mathbf{Diag}(2\mathbf{b}^\top A) \mathbf{x}. \end{aligned} \quad (24)$$

Формулировка (23) становится квадратичной, а матрица  $M = A^\top A - \mathbf{Diag}(2\mathbf{b}^\top A)$ .

В случае последовательной формулировки задача оптимизации является изначально квадратичной и может быть записана в виде

$$\min_{\mathbf{x}} \quad \sum_v \sum_{p < P} \sum_{(i,j) \notin A} c_{ij} x_{v,p,i} x_{v,p+1,j} + \rho w(\mathbf{x}), \quad (25)$$

где  $w(\mathbf{x})$  – функция, фиксирующая ограничения.

Учет ограничений вида  $A \mathbf{x} = \mathbf{b}$  происходит по аналогии с предыдущим случаем. Однако в этой формулировке появляются ограничения вида  $x_{v,p,i} x_{v,p+1,j} = 0$ . Для включения

этих ограничений в формулировку задачи квадратичной бинарной оптимизации без ограничений QUBO [5] учтем, что  $x_{v,p,i}x_{v,p+1,j}$  может быть равно либо 0, либо 1. Таким образом, добавляя в формулу слагаемое вида (учитывается ограничение  $Ax < b$ )

$$\min_{\mathbf{x}} \rho \sum x_{v,p,i}x_{v,p+1,j}, \quad (26)$$

мы по умолчанию учитываем такие ограничения. Функция ограничений может быть определена следующим образом:

$$w : \mathbf{x} \rightarrow \|A\mathbf{x} - \mathbf{b}\|^2 + \sum_v \sum_{p < P} \sum_{(i,j) \notin \mathcal{A}} x_{v,p,i}x_{v,p+1,j} + \sum_v \sum_{p=2}^{P-1} \sum_{j:j \neq d} x_{v,p,d}x_{v,p+1,j}. \quad (27)$$

В отдельных случаях возникает необходимость введения ограничений вида  $A\mathbf{x} \leq \mathbf{b}$  ( $A\mathbf{x} \geq \mathbf{b}$ ). Их можно учесть с помощью введения дополнительного вектора переменных  $\mathbf{s}$ . При добавлении к минимизируемой функции (21) слагаемого

$$\rho \|A\mathbf{x} - \mathbf{b} + \mathbf{s}\|^2 \quad (28)$$

осуществляется учет ограничений в виде неравенства. Параметр  $\mathbf{s}$  контролирует выполнение неравенства, искусственно превращая его в равенство. Оптимизация в этом случае осуществляется по двум наборам переменных  $\mathbf{x}$  и  $\mathbf{s}$ . Как и раньше  $\mathbf{x} \in \{0, 1\}^n$ , а элементам  $\mathbf{s}$  разрешено меняться непрерывно на заданном диапазоне. При этом на каждом шаге оптимизации по  $\mathbf{x}$  запускается оптимизация по  $\mathbf{s}$ , что значительно усложняет процедуру поиска решения.

Чтобы перейти к алгоритму QAOA, нужно связать каждую переменную с соответствующим кубитом квантового регистра. Если переменная принимает значение 0, то это соответствует состоянию  $|0\rangle$ , аналогично значение переменной 1 соответствует кубиту, находящемуся в состоянии  $|1\rangle$ .

В описанных выше задачах оптимизации квадратичная функция от  $n$  двоичных переменных должна быть минимизирована по всем возможным  $2^n$  значениям. Целевая функция задачи оптимизации функция может быть представлена следующим образом:

$$H(\mathbf{x}) = \alpha + \sum_{i=0}^{2^N-1} \alpha_i x_i + \sum_{k,j} \alpha_{k,j}^{2^N-1} x_k x_j, \quad (29)$$

где  $\mathbf{x} \in \{0, 1\}^n$ ,  $\alpha, \alpha_i, \alpha_{kj}$  – вещественные коэффициенты. В каждой из описанных ранее задач такой функции соответствует гамильтониан задачи  $H_P$ . Построение гамильтониана осуществляется с помощью матрицы Паули  $\sigma^z$ . Необходимо сделать переход от бинарных  $x$  к операторам, для этого сделаем замену  $\sigma^z \rightarrow \frac{I - \sigma^z}{2}$ . Матрицы  $\frac{I - \sigma_z}{2}$  отвечают собственные векторы  $|0\rangle, |1\rangle$  с собственными значениями 0 и 1. Сделаем переход

$$x_i \rightarrow \frac{I - \sigma_i^z}{2}, \quad (30)$$

где

$$\sigma_i^z = \underbrace{I \otimes I \otimes \dots \otimes I}_{i-1} \otimes \sigma^z \otimes \underbrace{I \otimes \dots \otimes I}_{N-i} \quad (31)$$

представляет собой гейт Паули, действующий на кубит с индексом  $i$ . Тогда в общем виде гамильтониан задачи можно представить следующим образом:

$$H_P = \beta + \sum_{i=0}^{N-1} \beta_i \sigma_i^z + \sum_{k,j=0}^{N-1} \beta_{kj} \sigma_k^z \sigma_j^z, \quad (32)$$

где  $\beta$ ,  $\beta_i$ ,  $\beta_{kj}$  – вещественные коэффициенты. Такая форма гамильтониана соответствует модели Изинга [6].

## 2.4 Quantum Approximate Optimization Algorithm

Quantum Approximate Optimization Algorithm (QAOA) – это квантовый алгоритм, предназначенный для приближенного решения задач комбинаторной оптимизации. Схематическое изображение алгоритма приведено на рисунке 1.

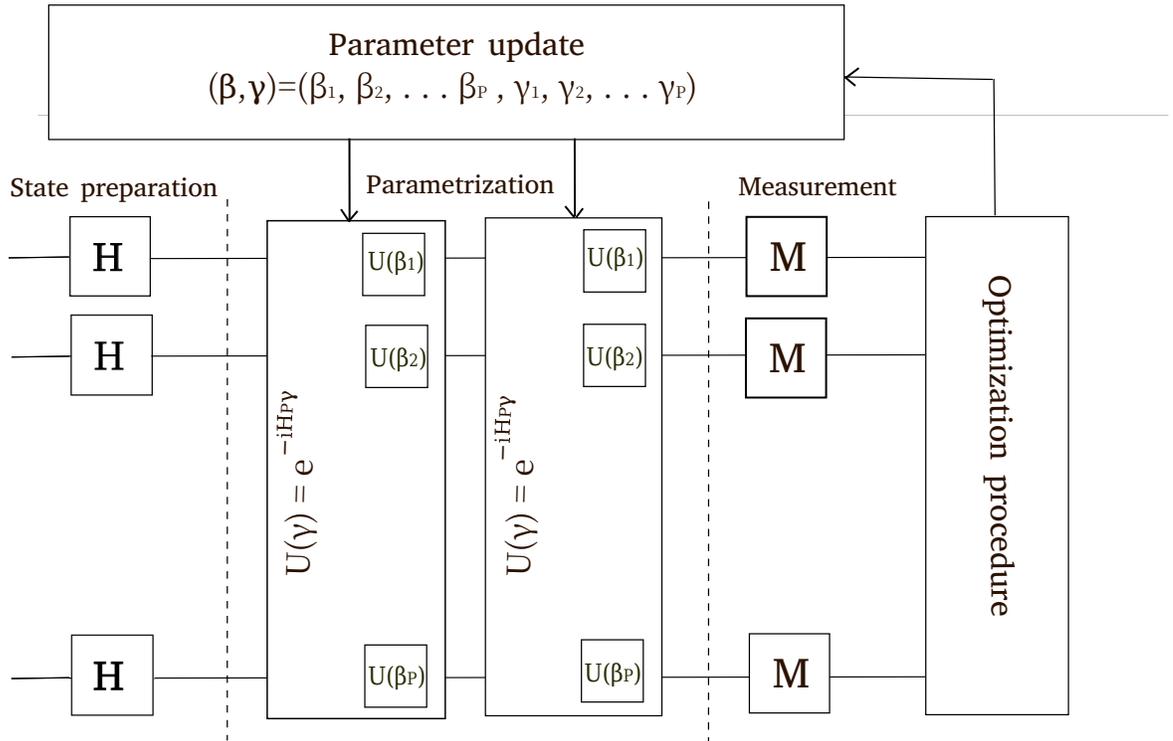


Рис. 1: Схема Quantum Approximate Optimization Algorithm

Первым шагом QAOA является подготовка начального состояния  $|\psi\rangle$ , представляющего собой суперпозицию всех возможных состояний заданной размерности  $n$ . Это осуществляется с помощью действия гейта Адамара  $H$  на каждый кубит квантового регистра

$$|\psi\rangle = H^{\otimes n} |0\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{j \in \{0,1\}^n} |j\rangle. \quad (33)$$

В QAOA оператор  $U(\beta, \gamma)$ , зависящий от параметров  $(\beta, \gamma) = (\beta_1, \beta_2, \dots, \beta_P, \gamma_1, \gamma_2, \dots, \gamma_P)$ , применяется к начальному состоянию  $|\psi_0\rangle$  для получения параметризованного состояния  $|\psi(\beta, \gamma)\rangle$ .  $U(\beta, \gamma)$  состоит из чередующихся  $U(\beta) = e^{-iH_M\beta}$  и  $U(\gamma) = e^{-iH_P\gamma}$ , которые  $H_M$  – это смешивающий гамильтониан, а  $H_P$  – это гамильтониан задачи. Смешивающий гамильтониан определяется следующим образом:

$$H_M = \sum_{i=1}^n \sigma_i^x, \quad (34)$$

где  $\sigma_i^x$  – оператор Паули, действующий на кубит с индексом  $i$ . Таким образом, параметризованное состояние имеет вид

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \overbrace{U(\boldsymbol{\gamma})U(\boldsymbol{\beta}) \dots U(\boldsymbol{\beta})}^{\text{P times}} |\psi\rangle, \quad (35)$$

где целочисленный параметр  $P$  определяет количество слоев параметризации и позволяет контролировать точность результата: чем больше  $P$ , тем выше точность расчетов. Однако поскольку современные квантовые компьютеры очень шумные, увеличение  $P$  может привести к ухудшению результатов, поскольку ошибка квантового компьютера растет с увеличением глубины квантовой цепи.

Состояние  $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$  используется для подготовки целевой функции

$$f(\boldsymbol{\beta}, \boldsymbol{\gamma}) := \frac{\langle \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) | H_P | \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle}{\langle \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) | \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle}. \quad (36)$$

С помощью измерения на квантовом компьютере можно сделать оценку величины  $f(\boldsymbol{\beta}, \boldsymbol{\gamma})$ . Этот процесс повторяется на каждом шаге классической оптимизации, которая позволяет определить новые значения параметров. Целью алгоритма QAOA является поиск оптимальных параметров  $(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}})$ , при этом  $|\psi(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}})\rangle$  является решением задачи.

## 3 Расчет энергии ионизации

### 3.1 Основной формализм

Гамильтониан в терминах вторичного квантования имеет форму

$$\hat{H} = \sum_{p,q} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s, \quad (37)$$

где оператор рождения  $\hat{a}_p^\dagger$  и оператор уничтожения  $\hat{a}_p$  отвечают рождению или уничтожению электрона, описываемого релятивистской одноэлектронной орбиталью  $\psi_p$ . В данном случае орбиталь характеризуется главным квантовым числом  $n$ , четностью  $l$ , полным угловым моментом  $j$  и его проекцией  $m$ . Одноэлектронные интегралы определяются следующим образом

$$h_{pq} = \int d\mathbf{r} \psi_p^\dagger(\mathbf{r}) \hat{h}(\mathbf{r}) \psi_q(\mathbf{r}), \quad (38)$$

где

$$\hat{h} = -\frac{1}{2} \hat{\mathbf{p}}^2 - V_C(\mathbf{r}) \quad (39)$$

является гамильтонианом,  $\hat{\mathbf{p}}$  – оператор импульса,  $V_C(\mathbf{r})$  – потенциал Кулона. Двухэлектронные интегралы можно выразить как

$$h_{pqrs} = \int d\mathbf{r}_1 d\mathbf{r}_2 \psi_p^\dagger(\mathbf{r}_1) \psi_q^\dagger(\mathbf{r}_2) \frac{1}{r_{12}} \psi_s(\mathbf{r}_1) \psi_r(\mathbf{r}_2). \quad (40)$$

Здесь используются обозначения  $\hat{\mathbf{r}}_{12} = \frac{\mathbf{r}_{12}}{r_{12}}$ ,  $\mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2$  и  $r_{12} = |\mathbf{r}_{12}|$ . Одноэлектронные и двухэлектронные интегралы вычисляются на классическом компьютере и являются входными параметрами для получения решения на квантовом компьютере.

Волновая функция основного состояния многоэлектронной системы описывается гамильтонианом (37) и может быть выражена в виде суперпозиции детерминантов Слейтера  $|\Phi_n\rangle$

$$|\psi\rangle = \sum_n c_n |\Phi_n\rangle. \quad (41)$$

В терминах формализма вторичного квантования детерминанты Слейтера имеют вид

$$|\Phi_n\rangle = |f_0 \dots f_{N_{\text{orbs}}-1}\rangle = \prod_{i=0}^{N_{\text{orbs}}-1} \left(\hat{a}_i^\dagger\right)^{f_i} |vac\rangle, \quad (42)$$

где  $N_{\text{orbs}}$  – это число одноэлектронных орбиталей,  $f_i \in \{0, 1\}$  – число заполнения орбитали с индексом  $i$ . Волновые функции, соответствующие основному состоянию, и детерминанты Слейтера являются собственными состояниями операторов числа частиц, полного углового момента и четности.

Для симуляции системы, описываемой гамильтонианом (37), на квантовом компьютере необходимо сопоставить состояниям в базисе чисел заполнения состояния кубитов

$$|f_0 \dots f_{N_{\text{orbs}}-1}\rangle \rightarrow |\mathbf{q}\rangle \equiv |q_{N_{\text{orbs}}-1}\rangle \otimes \dots \otimes |q_0\rangle = |q_{N_{\text{orbs}}-1}, \dots, q_0\rangle. \quad (43)$$

Самый простой случай подобного преобразования – кодировка Йордана-Вигнера [11], где каждый кубит  $|q_i\rangle$  хранит число заполнения  $f_i$ . Операторы рождения и уничтожения, удовлетворяющие фермионным антикоммутиационным соотношениям, в кодировке Йордана-Вигнера определяются следующими соотношениями

$$\hat{a}_i^\dagger = \hat{Q}_i^+ \otimes \sigma_i^z \otimes \dots \otimes \sigma_0^z, \quad (44)$$

$$\hat{a}_i = \hat{Q}_i^- \otimes \sigma_i^z \otimes \dots \otimes \sigma_0^z, \quad (45)$$

где

$$\hat{Q}_i^\pm = \frac{1}{2} (\sigma^x \pm i\sigma^y). \quad (46)$$

Также можно рассмотреть кодировку четности [12], где каждый кубит  $|q_i\rangle$  хранит суммарную четность кубитов до него  $p_i$ , то есть

$$p_i = \sum_{s=0}^i f_s. \quad (47)$$

Операторы рождения и уничтожения в кодировке четности имеют вид

$$\hat{a}_i = \sigma_{N_{\text{orbs}}-1}^x \otimes \dots \otimes \sigma_{i+1}^x \otimes \mathcal{P}_i^-, \quad (48)$$

$$\hat{a}_i^\dagger = \sigma_{N_{\text{orbs}}-1}^x \otimes \cdots \otimes \sigma_{i+1}^x \otimes \mathcal{P}_i^+, \quad (49)$$

где

$$\mathcal{P}_i^\pm = \frac{1}{2} (\sigma_i^x \otimes \sigma_{i-1}^z \mp i\sigma_i^y), \quad (50)$$

при этом используются матрицы Паули

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (51)$$

Подставляя выражения для операторов рождения и уничтожения (44), (45) в кодировке Йордана-Вигнера или (48), (49) в кодировке четности в уравнение (37), получим

$$\hat{H} = \sum_n \gamma_n P_n, \quad (52)$$

где  $P_n \in \{I, \sigma^x, \sigma^y, \sigma^z\}^{\otimes N_{\text{orbs}}}$  – тензорное произведение операторов Паули или строки Паули (PS).

## 3.2 Variational Quantum Eigensolver

Алгоритм Variational Quantum Eigensolver (VQE) является гибридным квантово-классическим алгоритмом. Схема VQE представлена на рисунке 2. Его задача сводится к минимизации среднего значения  $E_\theta$  на параметризованных волновых функциях  $|\psi(\theta)\rangle$ :

$$E_\theta = \frac{\langle \psi(\theta) | H | \psi(\theta) \rangle}{\langle \psi(\theta) | \psi(\theta) \rangle}. \quad (53)$$

Параметризованные функции определяются следующим образом:

$$|\psi(\theta)\rangle = U(\theta) |\psi\rangle, \quad (54)$$

где  $U(\theta)$  – это параметризирующая квантовая цепь (анзац), которая действует на ссылочное состояние  $|\psi\rangle$ . Выбор анзаца крайне важен для успешного выполнения алгоритма VQE. Волновая функция  $|\psi(\theta)\rangle$  должна хорошо аппроксимировать функцию основного состояния. При этом число параметров  $U(\theta)$  должно расти полиномиально с ростом размеров базиса. Более того, параметризирующий оператор должен быть представим через квантовые гейты.

Квантовый компьютер в алгоритме VQE используется для подготовки состояния (54), а так же для измерения среднего значения (53). Параметры  $\theta$  обновляются в ходе классической оптимизационной процедуры. Процедура повторяется, пока не будет достигнута сходимость алгоритма.

Для оценки величины (53) на квантовом компьютере необходимо кубитное представление гамильтониана  $H$ . В кодировке Йордана-Вигнера и в кодировке четности данное представление имеет вид (52). Учитывая формулу (54), среднее значение (53) может быть представлено в виде

$$E_\theta = \sum_{i=1}^{|P|} w_i \langle \psi | U^\dagger(\theta) P_i U(\theta) | \psi \rangle. \quad (55)$$

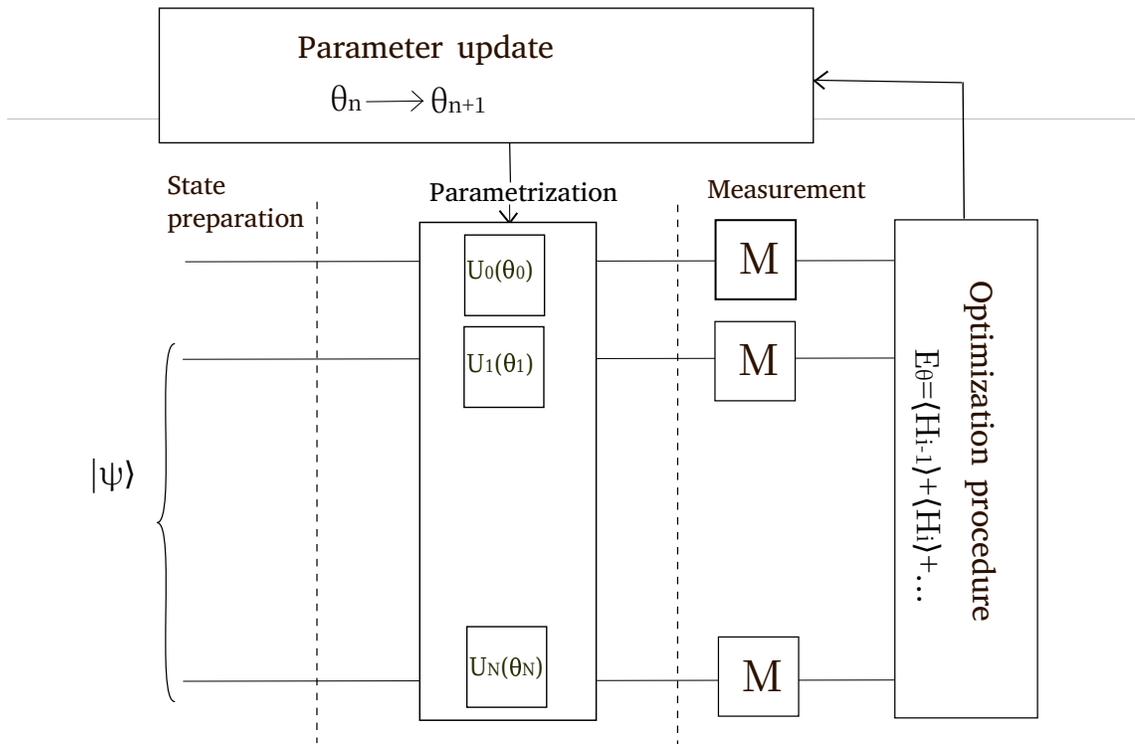


Рис. 2: Схема алгоритма Variational Quantum Eigensolver

Таким образом, в VQE оценка среднего значения гамильтониана сводится к оценке суммы средних значений матриц Паули, домноженных на вещественные константы.

### 3.3 Алгоритм расчета энергии ионизации

Для прямого вычисления разностей энергий на квантовом компьютере была разработана следующая процедура. Схема вычисления энергии ионизации на квантовом компьютере изображена на рисунке 3.

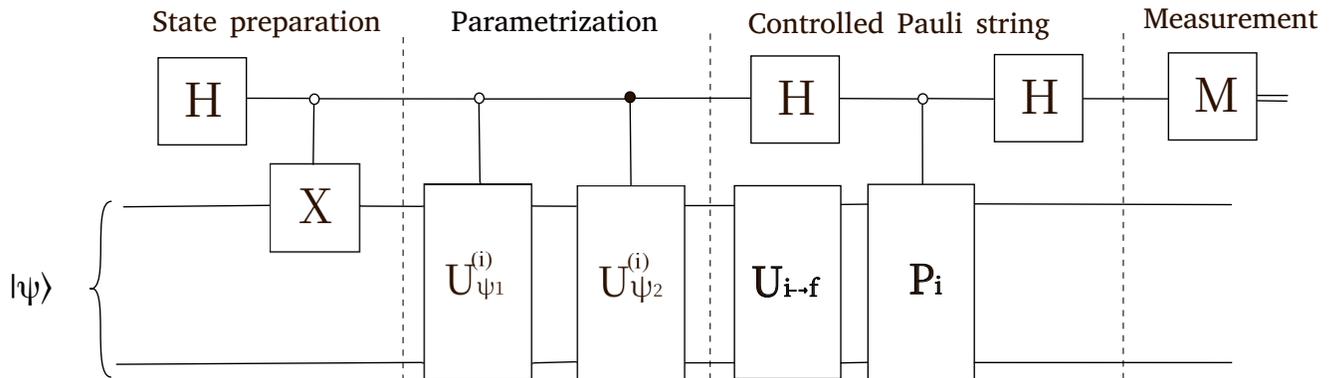


Рис. 3: Схема алгоритма расчета энергии ионизации

В алгоритме используется вычислительный регистр длиной  $N_q$ , где  $N_q$  – число кубитов, необходимое для кодирования состояния системы  $N$  частиц  $|\psi_N\rangle \in \{0, 1\}^{N_q}$ , и один вспомогательный кубит  $|q_0\rangle$ . Чтобы вычислить энергию ионизации, необходимо создать

состояние, находящееся в суперпозиции состояний частицы и иона. Для этого кубит  $|q_0\rangle$  из состояния  $|0\rangle$  переводится в состояние Белла  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  с помощью применения к нему гейта Адамара  $H$ . Операция CNOT, контролируемая  $|q_0\rangle$ , переводит систему в состояние

$$|\psi_{N,N-1}\rangle = \frac{1}{\sqrt{2}}(|1\rangle|\psi_N\rangle + |0\rangle|\psi_{N-1}\rangle), \quad (56)$$

где  $|\psi_N\rangle, |\psi_{N-1}\rangle$  – ссылочные состояния  $N$  и  $N-1$  частиц. В качестве ссылочных используются состояния, являющиеся решениями уравнений Хартри-Фока.

Далее необходимо перейти от ссылочных состояний к состояниям, которые соответствуют собственным функциям многоэлектронного гамильтониана. Аналогично алгоритму VQE это можно осуществить поиском минимума среднего значения гамильтониана  $H$  системы  $N$  и  $N-1$  частиц на параметризованных функциях. Зависимость от вектора параметров  $\theta$  присваивается с помощью квантовой цепи  $U(\theta)$ , действующей только на кубиты, кодирующие функцию состояния системы

$$|\psi_{N,N-1}(\theta)\rangle = \frac{1}{\sqrt{2}}(|1\rangle U(\theta)|\psi_N\rangle + |0\rangle U(\theta)|\psi_{N-1}\rangle). \quad (57)$$

В ходе классической оптимизационной процедуры осуществляется поиск минимума величины

$$E_\theta = \frac{\langle\psi_{N,N-1}(\theta)| H |\psi_{N,N-1}(\theta)\rangle}{\langle\psi_{N,N-1}(\theta)|\psi_{N,N-1}(\theta)\rangle}. \quad (58)$$

Этому минимуму соответствуют значения параметров  $\theta_{opt}$ , при этом  $|\psi_{N,N-1}(\theta_{opt})\rangle$  является суммой основных состояний частицы и иона.

Энергия ионизации имеет вид

$$E_{Ion} = \langle\psi_N(\theta_{opt})| H |\psi_N(\theta_{opt})\rangle - \langle\psi_{N-1}(\theta_{opt})| H |\psi_{N-1}(\theta_{opt})\rangle. \quad (59)$$

Чтобы вычислить (59), представим гамильтониан задачи  $H$  в терминах строк Паули  $P_i$ :

$$H = \sum_i \alpha_i P_i, \quad (60)$$

таким образом, выражение (59) примет вид

$$E_{Ion} = \sum_i \alpha_i (\langle\psi_N(\theta_{opt})| P_i |\psi_N(\theta_{opt})\rangle - \langle\psi_{N-1}(\theta_{opt})| P_i |\psi_{N-1}(\theta_{opt})\rangle). \quad (61)$$

То есть, для получения значения энергии ионизации можно сделать оценку отдельно для каждой строки Паули. Эта процедура может быть реализована на квантовом компьютере с помощью операции "Pauli-test" на рисунке 3. "Controlled Pauli string" состоит из двух гейтов Адамара  $H$ , действующих на вспомогательный кубит и строки Паули, контролируемой  $|q_0\rangle$ . После применения этой операции состояние системы соответствует

$$\begin{aligned} |\psi_{N,N-1}(\theta_{opt})\rangle \xrightarrow{CP_i} \frac{1}{2\sqrt{2}} (&|1\rangle (|\psi_N(\theta_{opt})\rangle + |\psi_{N-1}(\theta_{opt})\rangle - P_i |\psi_N(\theta_{opt})\rangle + P_i |\psi_{N-1}(\theta_{opt})\rangle) \\ &+ |0\rangle (|\psi_N(\theta_{opt})\rangle + |\psi_{N-1}(\theta_{opt})\rangle + P_i |\psi_N(\theta_{opt})\rangle - P_i |\psi_{N-1}(\theta_{opt})\rangle)). \end{aligned} \quad (62)$$

Введем обозначение

$$\Delta_i = \langle \psi_N(\boldsymbol{\theta}_{\text{opt}}) | P_i | \psi_N(\boldsymbol{\theta}_{\text{opt}}) \rangle - \langle \psi_{N-1}(\boldsymbol{\theta}_{\text{opt}}) | P_i | \psi_N(\boldsymbol{\theta}_{\text{opt}}) \rangle. \quad (63)$$

Для оценки значения энергии ионизации нужно выразить  $\Delta_i$  через результаты измерений квантовой цепи. Заметим, что вероятность измерить "0" на вспомогательном кубите составляет

$$\mathcal{P}_{0_i} = \frac{1}{2} - \frac{\Delta_i}{4}. \quad (64)$$

Таким образом, значение энергии ионизации имеет вид

$$E_{Ion} = \sum_i \alpha_i (2 - 4\mathcal{P}_{0_i}). \quad (65)$$

## 4 Параметризация состояния

Метод унитарных связанных кластеров (UCC) используется для параметризации "ссылочной" волновой функции. В отличие от классического метода связанных кластеров UCC сохраняет норму волновой функции. При этом не существует эффективной реализации UCC на классическом компьютере. Однако этот метод может быть эффективно реализован на квантовом компьютере. В методе унитарных связанных кластеров состояние может быть выражено следующим образом

$$|\psi_{\text{UCC}}\rangle = e^{\hat{T} - \hat{T}^\dagger} |\psi\rangle, \quad (66)$$

где  $\hat{T} = \sum_{\mu}^{\text{exc}} t_{\mu} \hat{\tau}_{\mu}$  – оператор возбуждения,  $|\psi\rangle$  – ссылочное состояние. В уравнении (66) суммирование происходит по всем возбуждениям с соответствующими кластерными амплитудами  $t_{\mu}$ , при этом

$$\hat{\tau}_{\mu} \equiv \hat{a}_{ij\dots}^{ab\dots} = \hat{a}_a^{\dagger} \hat{a}_b^{\dagger} \dots \hat{a}_j \hat{a}_i, \quad (67)$$

где индекс  $\mu$  включает в себя индексы  $i, j, \dots$  и  $a, b, \dots$ , нумерующие занятые и виртуальные орбитали ссылочного состояния. Будем считать, что кластерные амплитуды вещественные, тогда в соотношении (66) можно выразить степень экспоненты следующим образом

$$\hat{T} - \hat{T}^\dagger = \sum_{\mu}^{\text{exc}} t_{\mu} (\hat{\tau}_{\mu} - \hat{\tau}_{\mu}^\dagger). \quad (68)$$

Согласно общему формализму, разность  $\hat{T} - \hat{T}^\dagger$  может быть выражена через комбинацию строк Паули. На данный момент не существует эффективного метода представить этот анзац через квантовые гейты. Поэтому используем альтернативную формулировку с факторизованной экспонентой disentangled Unitary Coupled Cluster (dUCC) [9]

$$|\psi_{\text{dUCC}}\rangle = \prod_i e^{t_{\mu_i} (\tau_{\mu_i} - \tau_{\mu_i}^\dagger)} |\psi\rangle. \quad (69)$$

В dUCC анзаце все экспоненты расположены в особом порядке и встречаются ровно один раз. Заметим, что в соотношении (69) строки Паули, соответствующие разности  $\tau_\mu - \tau_\mu^\dagger$ , коммутируют между собой, таким образом, соответствующая экспонента может быть легко представлена в виде квантовой цепи.

Несмотря на то, что dUCC анзац можно легко представить с помощью квантовых гейтов, количество кластерных амплитуд растет экспоненциально с ростом размера базиса. Этот рост приводит к пропорциональному увеличению затрачиваемых вычислительных ресурсов как классических, так и квантовых, из-за чего проведение расчетов становится невозможным. В настоящей работе мы ограничимся рассмотрением однократных и двукратных возбуждений, такой анзац в дальнейшем будет обозначаться dUCC-SD. Этот анзац эквивалентен обычному анзацу UCC-SD, преобразованного с использованием первого порядка разложения Троттера

$$e^{\hat{T}-\hat{T}^\dagger} \approx \left( \prod_i e^{\frac{t_{\mu_i}}{\rho} (\tau_{\mu_i} - \tau_{\mu_i}^\dagger)} \right)^\rho, \quad (70)$$

где  $\rho = 1$ .

Приближенная факторизация не определена однозначно, при этом, порядок операторов влияет на точность, которая может быть достигнута при оценке среднего значения гамильтониана. В настоящей работе мы располагаем операторы однократного и двукратного возбуждений в порядке их появления в анзаце dUCC.

## 5 Вычисление градиентов на квантовом компьютере

Для реализации оптимизационных процедур таких, как Adam [21], Natural Gradients [22, 23] и Quantum Imaginary Time Evolution [24] необходима процедура вычисления градиентов на квантовом компьютере.

Рассмотрим функцию, зависящую от вектора параметров  $\boldsymbol{\theta}$

$$f(\boldsymbol{\theta}) := \langle H_i \rangle = \langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle. \quad (71)$$

Зависимость от параметра  $\theta_i \in \boldsymbol{\theta}$  квантовой цепи  $U(\boldsymbol{\theta})$  может быть представлена в общем виде

$$U_i(\theta_i) = \exp\left(-i\frac{\varphi(\theta_i)}{2}G\right), \quad (72)$$

где  $G$  – эрмитов оператор,  $\varphi(\theta_i)$  – обобщенная функция, для простоты можно положить  $\varphi(\theta_i) = \theta_i$ .

Перед описанием метода вычисления производных необходимо сделать преобразование генератора анзаца  $G$ . Это осуществляется с помощью унитарного оператора  $\mathcal{U}_G^\dagger$

$$\mathcal{U}_G^\dagger G \mathcal{U}_G = \sum_j \lambda_j \mathcal{P}_j =: D, \quad (73)$$

где  $\mathcal{P}_j = |j\rangle\langle j|$  – проектор на состояние  $|j\rangle$  оператора  $G$ . В таком случае параметризующая экспонента  $U_i(\theta_i)$  имеет вид

$$\exp(-i\theta_i G/2) = \mathcal{U}_G^\dagger \exp(-i\theta_i D/2) \mathcal{U}_G. \quad (74)$$

Переход (74) упрощает работу с матричными функциями, при чем любая функция  $g(\cdot)$  может быть представлена в виде суммы проекторов с соответствующими весами

$$g(D) = \sum_j g(\lambda_j) \mathcal{P}_j. \quad (75)$$

С учетом (75) можно выразить производную функции (71)

$$\begin{aligned} \frac{\partial f(\theta_i)}{\partial \theta_i} &= \partial_{\theta_i} \langle \psi | \left( \sum_{j=1}^d e^{i\theta_i \lambda_j} \mathcal{P}_j \right) Q \left( \sum_{j'=1}^d e^{-i\theta_i \lambda_{j'}} \mathcal{P}_{j'} \right) | \psi \rangle = \\ &= i \sum_{j,j'=1}^d (\lambda_j - \lambda_{j'}) e^{i\theta_i (\lambda_j - \lambda_{j'})} \langle \psi | \mathcal{P}_j Q \mathcal{P}_{j'} | \psi \rangle \end{aligned} \quad (76)$$

где видно, что производная функции представляет собой комбинацию неизвестных матричных элементов спроецированного целевого оператора с некоторыми комплексными элементами. Двойное суммирование в уравнении (76) может быть сведено к однократному. Для этого упорядочим собственные значения  $\Lambda$  оператора  $G$ , то есть будем рассматривать  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_d$ . Далее разделим вклады в уравнении (76) на три группы. Первая группа включает члены с индексами  $j > j'$ . Во второй группе наоборот рассматриваются  $j < j'$ , то есть для второй группы разность  $\lambda_j - \lambda_{j'}$  отличается знаком от такой же разности с индексами из первой группы, аналогично комплексные экспоненты  $\exp(i\theta_i (\lambda_j - \lambda_{j'}))$  отличаются сопряжением. Последняя группа членов с индексами  $j = j'$  не дает вклада в сумму (76). Стоит отметить, что вырожденные собственные значения для индексов  $j \neq j'$  также не дают вклад в производную. Исходя из выше сказанного, вместо набора собственных значений  $\Lambda$  можно рассматривать набор неотрицательных разностей  $\Gamma := \{\lambda_j - \lambda_{j'}\}_{j,j'=1}^d$ , где  $j > j'$ . Пусть  $\Delta_{s(j,j')} = |\lambda_j - \lambda_{j'}|$  для всех  $\lambda_j \neq \lambda_{j'}$ . Обозначим матричные элементы  $\langle \mathcal{P}_j Q \mathcal{P}_{j'} \rangle = O_s$ ,  $\langle \mathcal{P}_j Q \mathcal{P}_j \rangle = O_s^*$ . В терминах введенных обозначений производная выражается в виде

$$\frac{\partial f(\theta_i)}{\partial \theta_i} = \sum_s \Delta_s R_s, \quad (77)$$

где  $\{R_s\}_s$  – набор вещественных коэффициентов, который имеет вид

$$R_s = \frac{i}{2} e^{i\theta_i \Delta_s / 2} O_s - \frac{i}{2} e^{-i\theta_i \Delta_s / 2} O_s^* = -\sin\left(\frac{\theta_i}{2} \Delta_s\right) \operatorname{Re}\{O_s\} - \cos\left(\frac{\theta_i}{2} \Delta_s\right) \operatorname{Im}\{O_s\}. \quad (78)$$

Теперь задача вычисления производной сводится к поиску вектора  $\mathbf{R}$ . Это осуществляется с помощью измерения квантовой цепи со сдвигом аргумента на  $\pm \delta$  (конечный результат не зависит от выбора  $\delta$ )

$$\begin{aligned} f(\theta_i + \delta) - f(\theta_i - \delta) &= i \sum_{s=1}^S (e^{i\Delta_s \delta} - e^{-i\Delta_s \delta}) e^{i\theta_i \Delta_s} O_s + \text{H.c.} \\ &= 4 \sum_{s=1}^S \sin(\delta \Delta_s) R_s. \end{aligned} \quad (79)$$

С помощью соотношения (79) между вещественными коэффициентами и разностью измерений квантовой цепи на сдвинутых параметрах можно определить  $\mathbf{R}$ . Для некоторого

набора  $\{\delta_s\}_s$  введем обозначения  $F_s := f(\theta_i + \delta_s) - f(\theta_i - \delta_s)$ , тогда система уравнений на  $\mathbf{R}$  примет вид

$$\begin{aligned}
F_1 &= 4 \sum_{s=1}^S \sin(\delta_1 \Delta_s) R_s, \\
F_2 &= 4 \sum_{s=1}^S \sin(\delta_2 \Delta_s) R_s, \\
&\vdots \\
F_S &= 4 \sum_{s=1}^S \sin(\delta_S \Delta_s) R_s.
\end{aligned} \tag{80}$$

Введем вектор разностей измерений  $\mathbf{F} = (F_1, F_2, \dots, F_S)$ , система уравнений может быть записана компактно  $\mathbf{F} = \mathbb{M}\mathbf{R}$ , где  $\mathbb{M}$  – матрица синусов, полученная из известных фаз. Обратив матрицу  $\mathbb{M}$ , окончательно получим систему уравнений на вещественные коэффициенты

$$\mathbf{R} = \mathbb{M}^{-1}\mathbf{F}. \tag{81}$$

## 6 Результаты

### 6.1 Задача логистики с временными окнами

Целью задачи логистики ставилась минимизация величины (8) для последовательной формулировки с соответствующими ограничениями, также дополнительно вводилось ограничение на временные окна. При тестировании задачи логистики буровых установок с временными окнами была рассмотрена модель из 10 кустов и 2 буровых установок, что соответствует 286 бинарным переменными задачи QUBO. В таблице 1 содержатся результаты, полученные с помощью симулятора когерентной машины Изинга (КМИ) [17], алгоритма QAOA и классической задачи QUBO с оптимизаторами Or-tools [18], жадного алгоритма [19] и симуляции отжига [20]. В алгоритме QAOA использовался один слой параметризующих операторов и безградиентная оптимизационная процедура COBYLA. Реализация алгоритма симуляции отжига проводилась для маршрутной и последовательной формулировок с фиксированными входными параметрами: время отжига = 20000, число отжигов = 300, которые были подобраны в ходе тестирования. Время работы алгоритма Or-tools и симулятора когерентной машины Изинга составляет 20 и 12 секунд соответственно и является оптимальным для поиска решения задачи данной размерности.

Для представления эффективности алгоритмов была собрана статистика на 100 запусках для каждого алгоритма и представлена вероятность получения решения. Из полученных решений были выделены оптимальные и занесены в колонку ”Длина пути” таблицы 1.

По результатам тестирования видно, что наиболее часто допустимое решение находит жадный алгоритм и симулятор когерентной машины Изинга, все тестовые запуски были успешными, алгоритм симуляции отжига дал порядка двух верных решений, а запуски Or-tools ни разу не привели к удовлетворительному решению, поскольку результат ни разу не

Алгоритм	Формулировка	Длина пути	Вероятность получения решения
Or-tools	–	–	0.0
Жадный	–	9859	1.0
Симуляция отжига	Последовательная	9272	0.02
Симуляция когерентной машины Изинга	Последовательная	9071	1.0
QUBO	Последовательная	–	0.0

Таблица 1: Результаты тестирования алгоритмов на 100 запусках для 10 кустов и 2 буровых установок

смог удовлетворить ограничению на временные окна. Длина путей, полученная с помощью симулятора когерентной машины Изинга, является наименьшей, это свидетельствует, что данный метод является наиболее оптимальным при решении подобной задачи.

## 6.2 Задача логистики без временных окон

Тестирование задачи логистики без временных окон осуществлялось на нескольких наборах данных задач TSP и VRP, что позволило оценить сходимость задач разной размерности. Были рассмотрены маршрутная и последовательная формулировки. В ходе тестирования задачи были получены результаты с помощью алгоритма QAOA с оптимизационной процедурой COBYLA, для сравнения были посчитаны результаты с помощью алгоритма симуляции отжига, Or-tools и симулятора когерентной машины Изинга. Характеристики алгоритма симуляции отжига: время отжига = 20000, число отжигов = 300, были подобраны в результате тестирования. Время работы симулятора когерентной машины Изинга 20 секунд, время работы алгоритма Or-tools 10 секунд также были выбраны в ходе тестирования. В таблице 2 содержатся результаты для наименьшего расстояния, полученные в ходе для различных наборов данных задач TSP и VRP. Результаты, приведенные в таблице 3 соответствуют вероятностям получения допустимого решения.

Формулировка	Набор данных		QAOA	Симуляция отжига	Or-tools	КМИ
	Число кустов	Число вышек				
Последовательная	5	1	3441	3378	3294	3012
	7	1	3528	4300	3597	3545
	10	1	–	10482	9157	9152
	14	1	–	11489	11302	11302
	24	1	–	24138	23611	23611
	29	1	–	30391	30521	28029
	4	2	1596	1350	1183	1280
	6	2	1992	2328	1977	2252
	8	2	–	4410	4551	4317
	9	3	–	4592	4463	4701
	12	–	–	4359	4879	4231
	Маршрутная	5	1	2389	2699	2477
7		1	2863	3374	2963	–
10		1	4692	–	4990	–
14		1	–	–	–	–
24		1	–	–	–	–
29		1	–	–	–	–
4		2	2210	2170	3012	–
6		2	3425	3445	3545	–
8		2	6552	6649	9152	–
9		3	–	–	–	–
12		4	–	–	–	–

Таблица 2: Оптимальные длины маршрутов для буровых установок, полученные в ходе тестирования алгоритма QAOA с классическим оптимизатором COBYLA, QUBO задачи с алгоритмом симуляции отжига, Or-tools, симулятором когерентной машины Изинга

Формулировка	Набор данных		QAOA	Симуляция отжига	Or-tools	КМИ
	Число кустов	Число вышек				
Последовательная	5	1	1.0	1.0	1.0	1.0
	7	1	1.0	1.0	1.0	1.0
	10	1	0.0	1.0	1.0	1.0
	14	1	0.0	1.0	1.0	1.0
	24	1	0.0	1.0	1.0	1.0
	29	1	0.0	1.0	1.0	1.0
	4	2	1.0	1.0	1.0	1.0
	6	2	1.0	1.0	1.0	1.0
	8	2	0.0	1.0	1.0	1.0
	9	3	0.0	1.0	1.0	1.0
	12	0.0	0.0	1.0	1.0	1.0
	Маршрутная	5	1	1.0	0.03	1.0
7		1	1.0	0.03	1.0	–
10		1	1.0	0.0	1.0	–
14		1	0.0	0.0	0.0	–
24		1	0.0	0.0	0.0	–
29		1	0.0	0.0	0.0	–
4		2	1.0	0.05	1.0	–
6		2	1.0	0.01	1.0	–
8		2	1.0	0.01	1.0	–
9		3	0.0	0.0	0.0	–
12		4	0.0	0.0	0.0	–

Таблица 3: Вероятности получения результатов, удовлетворяющих ограничениям, для алгоритма QAOA с классическим оптимизатором COBYLA, QUBO задачи с алгоритмом симуляции отжига, Or-tools, симулятором когерентной машины Изинга

### 6.3 Результаты вычисления энергии ионизации

При реализации процедуры вычисления энергии ионизации был рассмотрен оптимизационный алгоритм Adam, для которого необходима оценка градиентов на квантовом компьютере, и безградиентный алгоритм COBYLA. Расчеты проводились в базе stobg для молекулы  $H_2$ . Пакеты pyscf и qiskit-nature использовались для оценки энергий HF, FCI, а так же для построения одноэлектронных и двухэлектронных интегралов. Результаты были получены в кодировке Йордана-Вигнера на 5 кубитах. Результаты, полученные в ходе реализации алгоритма расчета энергии ионизации, оценивались относительно разности энергий основного состояния молекулы водорода и иона, посчитанных с помощью методов Хартри-Фока (HF) и полного конфигурационного взаимодействия (FCI). Разность энергий FCI принимается за точное значение энергии ионизации. Также алгоритм расчета энергии ионизации был протестирован с простейшими ошибками (однокубитные, двухкубитные и ошибка измерения).

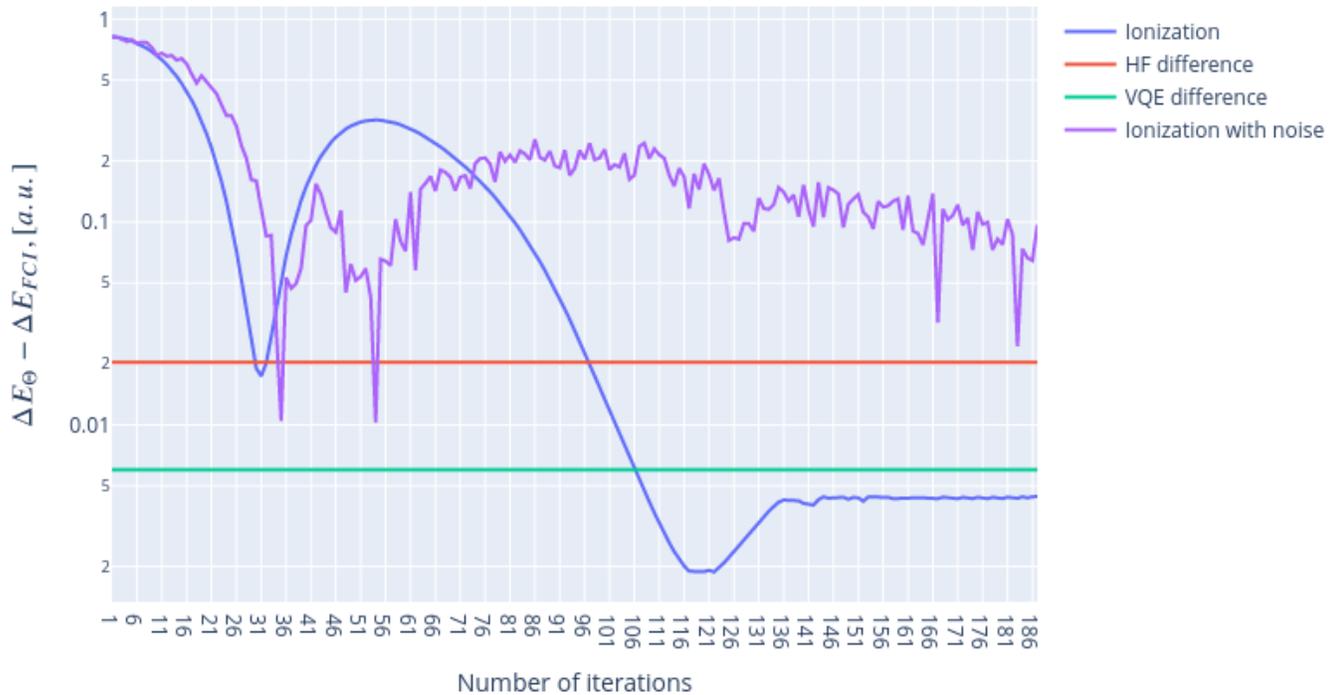


Рис. 4: Разность между энергией ионизации  $H_2$ , полученной с помощью алгоритма оптимизации Adam с dUCC-SD анзацем (на симуляторе идеального квантового компьютера и шумного квантового компьютера), и энергией полного конфигурационного взаимодействия (FCI), как функция от числа итераций. Энергия Хартри-Фока (HF) обозначена красной линией. Энергия разности VQE обозначена зеленой линией.

На рисунке 4 представлены результаты расчетов энергии ионизации с помощью оптимизационной процедуры Adam. Алгоритм расчета энергии ионизации, реализованный с

помощью симулятора идеального квантового компьютера, сходится к значению, которое лежит ниже, чем разность энергий Хартри-Фока и разность энергий, каждая из которых отдельно оценена с помощью процедуры VQE. Для сходимости алгоритму Adam необходимо порядка 200 итераций. Однако при добавлении шумов результаты перестают сходиться, то есть нельзя выделить плато из-за наличия колебаний, и лежат значительно выше разности энергий, посчитанных с помощью метода Хартри-Фока. Такое различие при относительно небольших шумах объясняется структурой оптимизационной процедуры Adam. Поскольку в алгоритме необходимо вычислять градиенты для оценки изменения параметров, то число измерений квантовой цепи для dUCC-SD анзаца на каждом шаге оптимизации увеличивается в 4 раза. Увеличение количества измерений ведет к росту ошибки.

Optimization procedure	VQE difference, a.u.	VQE difference with noise, a.u.	Ionization energy, a.u.	Ionization energy with noise, a.u.
COBYLA	$4.9 \times 10^{-3}$	$5.9 \times 10^{-3}$	$5.3 \times 10^{-3}$	$5.4 \times 10^{-3}$
Adam	$4.9 \times 10^{-3}$	–	$1.7 \times 10^{-3}$	–

Таблица 4: Энергия ионизации и разности значений VQE относительно разности энергий FCI для молекулы  $H_2$ , полученные с помощью алгоритмов оптимизации Adam и COBYLA с dUCC-SD анзацем на симуляторе идеального квантового компьютера и шумного квантового компьютера.

В таблице 4 результаты расчетов энергии ионизации с помощью оптимизационных процедур Adam и COBYLA. В отличие от алгоритма Adam алгоритму COBYLA необходимо около 500 итераций, для того чтобы сойтись. При добавлении шумов алгоритм сходится намного быстрее, примерно за 100 итераций, значение энергии ионизации ниже разности энергий, каждая из которых отдельно оценена с помощью процедуры VQE с теми же шумами. При этом энергия ионизации, полученная с помощью симулятора идеального квантового компьютера, хуже разности результатов энергий VQE. Разница, однако, незначительная. Из полученных результатов можно сделать вывод о корректности работы алгоритма расчета энергии ионизации.

## 7 Выводы

### 7.1 Задача логистики буровых установок

Результаты тестирования показали, что квантовые алгоритмы позволяют получать допустимые решения с удовлетворительной точностью. Эти решения обычно имеют значение целевых функций, порядка значения, полученного с помощью жадного алгоритма. Это может быть связано с низкой комбинаторной сложностью задач, либо, что более вероятно, с тем, что метод генерации данных идеально подходит для решения жадным алгоритмом. Проверка данной гипотезы подразумевает полный перебор всех решений и даже на задачах малой размерности. Ее реализация не представляется возможной за разумное время.

В отдельных случаях квантовые и квантово-вдохновленные алгоритмы показывали решение лучше классических. Например, в задаче логистики с временными окнами OR-Tools ни разу не нашел допустимое решение. В случае с OR-Tools было замечено, что он работает существенно лучше, когда ему предлагается какое-либо первоначальное решение (например, из жадного алгоритма). Также в задаче логистики без временных окон алгоритмы QAOA, симуляции отжига и симулятор когерентной машины Изинга зачастую давали решения более оптимальные, чем алгоритм Or-tools.

Стоит отметить, что решение существенно зависит используемого солвера. Например, облачные решения симуляции отжига для задач QUBO с вероятностью 100% выдавали допустимое решение, однако оно всегда было хуже жадного алгоритма. С другой стороны, реализация симуляции отжига QUBO из пакета qubover, которая использовалась во всех локальных тестах, выдает куда более хаотические результаты, и вероятность нахождения допустимого решения порядка 2%. Также, необходимо отметить, что такую метрику как "вероятность допустимого решения" довольно сложно определить для разных солверов, поскольку все квантово-вдохновленные солверы имеют разные параметры и критерии остановки. Например, в симуляции отжига в качестве критериев используются

- количество отжиггов,
- время одного отжига,

а в алгоритме Or-tools критериями являются

- стартовый алгоритм,
- время работы алгоритма,

что сложно сравнимо, например, с тем же жадным алгоритмом.

### 7.2 Расчет энергии ионизации

Алгоритм прямого вычисления энергии ионизации был рассмотрен на примере молекулы водорода. При расчете использовалось два вида оптимизационной процедуры Adam и COBYLA. Алгоритму Adam на симуляторе идеального квантового компьютера для сходимости потребовалось порядка 200 итераций, при этом алгоритм COBYLA сошелся за практически 500 итераций. Энергия ионизации, получаемая с помощью Adam находится ближе к разности энергий, полученных с помощью метода полного конфигурационного взаимодействия, нежели энергия, полученная с помощью COBYLA. Однако алгоритм расчета

энергии ионизации с оптимизационной процедурой Adam на симуляторе шумного квантового компьютера не имеет выраженной области сходимости, поэтому не представляется возможным оценить значение энергии ионизации.

В случае алгоритма COBYLA, который не требует оценки градиента на каждой итерации, можно получить результаты на симуляторе с шумами. При этом на шумном симуляторе результат энергии ионизации лучше, нежели разность энергий основного состояния посчитанных с помощью VQE с той же оптимизационной процедурой. Из полученных результатов можно сделать вывод о корректности работы алгоритма расчета энергии ионизации.

## Список литературы

- [1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2001).
- [2] И. Манин, *Вычислимое и невычислимое* (Сов. радио, Москва, 1980).
- [3] R. Feynman, *Int. J. Theor. Phys.* 21, 467 (1982).
- [4] P. Shor, *SIAM J. Comput.* 26, 5 (1997).
- [5] C. Gambella, and A. Simonetto, *IEEE Trans. Quantum Eng.* 1, 1-22 (2020).
- [6] S. Harwood, C. Gambella, D. Tenev, A. Siminetto, D. Bernal, and J. Greenberg, *IEEE Trans. Quantum Eng.* 2, 1-17 (2021).
- [7] A. Peruzzo, J. McClean, P. Shadbolt, M. Yung, X. Zhou, P. Love, A. Aspuru-Guzik, and J. L. O'Brien, *Nat Commun* 5, 4213 (2014).
- [8] E. Farhi, J. Goldstone, and S. Gutman, *arXiv: 1411.4028* (2014).
- [9] F. Evangelista, G. K.-L. Chan, and G. Scuseria, *J. Chem. Phys.* 151, 244112 (2019).
- [10] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Nature* 549, 242 (2017).
- [11] P. Jordan and E. Wigner, *Z. Phys.* 47, 631 (1928).
- [12] S. B. Bravyi and A. Y. Kitaev, *Ann. Phys.*, 298, 210 (2002).
- [13] O. Kyriienko and V. E. Elfving, *Phys. Rev. A* 104, 052417 (2021).
- [14] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, *Phys. Rev. A* 99, 032331 (2019).
- [15] K. Mitarai and K. Fujii, *Phys. Rev. Res.* 1, 013006 (2019).
- [16] I. Sokolov, P. Barkoutsos, L. Moeller, P. Suchsland, and G. Mazzola, *Chem. Phys.* 152, 124107 (2020).
- [17] Y. Haribara, S. Utsunomiya, and Y. Yamamoto, *Entropy* 18(4), 151 (2016).

- [18] L. Perron and V. Furnon, OR-Tools, <https://developers.google.com/optimization>
- [19] Т. Кормен, Ч. Лейзерсон, Р. Ривест и Штайн К. — Вильямс, Алгоритмы. Построение и анализ: [пер. с англ.], ISBN: 9785845908575 (2009).
- [20] V. Černý, J. Optim. Theory 45, 41–51 (1985).
- [21] D. Kingma and J. Ba, arXiv:1412.6980
- [22] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum 4, 269 (2020).
- [23] D. Wierichs, C. Gogolin, and M. Kastoryano, Phys. Rev. Res. 2, 043246 (2020).
- [24] S. McArdle, T. Jones, S. Endo, Y. Li, S. Benjamin, and X. Yuan, npj Quantum Inf. 5, 75 (2019).