

doi: 10.17586/2226-1494-2026-26-3-652-661

УДК 004.02

Поиск с квантовым ускорением кратчайшего пути на графе в несортированном пространстве

Паскаль Мишелевич Хури¹, Алина Дмитриевна Клишева², Петр Сергеевич Демченко³,
Ксения Витальевна Бебех⁴, Михаил Константинович Ходзицкий⁵,
Анна Викторовна Возианова⁶✉

^{1,2,4,6} Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

^{3,5,6} ООО «Терагерцовая Фотоника», Санкт-Петербург, 191167, Российская Федерация

⁴ ИП Бебех А.А., Санкт-Петербург, 198411, Российская Федерация

¹ paskal_khuri@mail.ru, <https://orcid.org/0009-0006-9817-2688>

² klishevaad@gmail.com, <https://orcid.org/0009-0008-4860-7019>

³ petr.s.demchenko@gmail.com, <https://orcid.org/0000-0003-4948-2023>

⁴ ksenya-gbd@yandex.ru, <https://orcid.org/0009-0001-7116-9028>

⁵ khodzitskiy@yandex.ru, <https://orcid.org/0000-0001-7261-8350>

⁶ anna-vozanova@yandex.ru✉, <https://orcid.org/0000-0002-4550-7169>

Аннотация

Введение. Алгоритмы и структуры данных важны в информационных технологиях (в частности, в терагерцовых беспроводных коммуникациях 6G), логистике, транспорте и планировании маршрутов в медицине. Одним из основополагающих алгоритмов, который находит применение во многих практических задачах, является алгоритм поиска кратчайшего пути на графе. Он используется для нахождения оптимального маршрута между двумя вершинами, минимизируя длину или количество промежуточных узлов. Эта задача относится к 21 NP-трудной проблеме, сформулированной Ричардом Карпом в 1972 году. Она по-прежнему представляет собой серьезный вызов для науки, поскольку не существует полиномиального алгоритма, способного обеспечить точное ее решение. **Метод.** Рассмотрены подходы к решению задачи поиска кратчайшего пути на графе, а также обсуждаются их преимущества и недостатки. Изучены алгоритмы, такие как Беллмана–Форда, Дейкстры и A*; проанализированы их эффективность и применимость в различных ситуациях. Обсуждаются вопросы оптимизации алгоритмов поиска кратчайшего пути, такие как использование эвристик, предварительная обработка графа и другие методы, которые могут улучшить производительность и точность алгоритмов. **Основные результаты.** Предложена оптимизация алгоритма Дейкстры с квантовым ускорением. Проведена оценка устойчивости квантового алгоритма и величины искажений в схеме с различным количеством кубитов. Показано, что с возрастанием интенсивности искажений вероятность обнаружения искомого элемента уменьшается, а время, затрачиваемое на его поиск, увеличивается. При применении алгоритма Гровера для оптимизации пути в квантовой сети эти помехи становятся причиной увеличения времени поиска оптимального пути и повышения вероятности выбора неверного маршрута. Показано, что увеличение количества кубитов и, соответственно, количества итераций делает алгоритм более устойчивым. Продemonстрировано, что устойчивость алгоритма Гровера для схем с большим количеством кубитов имеет схожий характер. Работы квантовых схем на 9, 10 и 11 кубитах различие в значениях вероятностей верного срабатывания отличается менее, чем на 10^{-3} . Таким образом, из полученных зависимостей можно предсказать характер такого влияния на схемы с еще большим числом кубитов, устойчивость которых не может быть исследована вследствие недостаточной вычислительной мощности классических компьютеров. **Обсуждение.** Наиболее эффективным способом модификации алгоритма Дейкстры является замена очереди с приоритетом в классической версии алгоритма на некоторую квантовую структуру данных, находящую минимальный элемент для следующей итерации алгоритма. Несмотря на то, что квантовый поиск не дает потенциального прироста в асимптотике, можно использовать иную квантовую модификацию алгоритма Дейкстры. А именно, вместо поддержания и поиска минимума в очереди с приоритетом применить квантовый алгоритм сразу для поиска вершины, которую необходимо добавить в дерево кратчайших путей, что дополнительно даст ускорение обхода графа алгоритмом Дейкстры.

© Хури П.М., Клишева А.Д., Демченко П.С., Бебех К.В., Ходзицкий М.К., Возианова А.В., 2026

Ключевые слова

поиск оптимального пути, графы, алгоритм Дейкстры, квантовые алгоритмы, алгоритм Гровера, квантовое ускорение, несортированное пространство

Благодарности

Работа выполнена в рамках гранта Фонда развития Центра разработки и коммерциализации новых технологий (Фонда Сколково) в рамках Соглашения № 40113/0744-2024 от 19 сентября 2024 г.

Ссылка для цитирования: Хури П.М., Клишева А.Д., Демченко П.С., Бебех К.В., Ходзицкий М.К., Возианова А.В. Поиск с квантовым ускорением кратчайшего пути на графе в несортированном пространстве // Научно-технический вестник информационных технологий, механики и оптики. 2026. Т. 26, № 3. С. 652–661. doi: 10.17586/2226-1494-2026-26-3-652-661

Quantum-accelerated shortest path search on a graph in unsorted space

Paskal M. Khuri¹, Alina D. Klisheva², Petr S. Demchenko³, Ksenia V. Bebekh⁴,
Mikhail K. Khodzitsky⁵, Anna V. Vozianova⁶✉

^{1,2,4,6} ITMO University, Saint Petersburg, 197101, Russian Federation

^{3,5,6} Terahertz Photonics, LLC, Saint Petersburg, 191167, Russian Federation

⁴ IE Bebekh A.A., Saint Petersburg, 198411, Russian Federation

¹ paskal_khuri@mail.ru, <https://orcid.org/0009-0006-9817-2688>

² klishevaad@gmail.com, <https://orcid.org/0009-0008-4860-7019>

³ petr.s.demchenko@gmail.com, <https://orcid.org/0000-0003-4948-2023>

⁴ ksenya-gbd@yandex.ru, <https://orcid.org/0009-0001-7116-9028>

⁵ khodzitskiy@yandex.ru, <https://orcid.org/0000-0001-7261-8350>

⁶ anna-vozanova@yandex.ru✉, <https://orcid.org/0000-0002-4550-7169>

Abstract

Algorithms and data structures are important in information technology (particularly in 6G terahertz wireless communications), logistics, transportation, and route planning in medicine. One of the fundamental algorithms that find application in many practical problems is the shortest path algorithm. The shortest path algorithm is used to find the optimal route between two vertices while minimizing the length or number of intermediate nodes. This problem is one of the 21 NP-hard problems formulated by Richard Karp in 1972. It still poses a serious challenge to science, since there is no polynomial algorithm that can provide an exact solution to this problem. The article considers approaches to solving the problem of finding the shortest path on a graph, and discusses their advantages and disadvantages. Various algorithms, such as the Bellman-Ford algorithm, Dijkstra, and the A* algorithm, are considered; their efficiency and applicability in various situations are analyzed. Issues of optimization of shortest path algorithms are discussed, such as the use of heuristics, graph preprocessing, and other methods that can improve the performance and accuracy of algorithms. The article proposes an optimization of Dijkstra's algorithm with quantum acceleration. The stability of the quantum algorithm and the magnitude of distortions in a circuit with a different number of qubits are estimated. It is shown that with an increase in the intensity of distortions, the probability of detecting the desired element decreases, and the time spent on its search increases. When using Grover's algorithm to optimize a path in a quantum network, these interferences cause an increase in the time it takes to find the optimal path and an increase in the probability of choosing an incorrect route. It is shown that an increase in the number of qubits and, accordingly, the number of iterations makes the algorithm more stable. It is demonstrated that the stability of Grover's algorithm for circuits with a large number of qubits has a similar nature. The work of quantum circuits on 9, 10 and 11 qubits, the difference in the probabilities of correct operation differs by less than 10^{-3} . Thus, from the obtained dependencies, it is possible to predict the nature of such an influence on circuits with an even larger number of qubits, the stability of which cannot be studied due to the insufficient computing power of classical computers. The most efficient way to modify Dijkstra's algorithm is to replace the priority queue in the classical version of the algorithm with some quantum data structure that finds the minimum element for the next iteration of the algorithm. Despite the fact that quantum search does not provide a potential gain in asymptotics, another quantum modification of Dijkstra's algorithm can be used. Namely, instead of maintaining and searching for a minimum in the priority queue, use the quantum algorithm immediately to search for a vertex that needs to be added to the shortest path tree, which will additionally acceleration the graph traversal by Dijkstra's algorithm.

Keywords

optimal path search, graphs, Dijkstra's algorithm, quantum algorithms, Grover's algorithm, quantum acceleration, unsorted space

Acknowledgements

The work was supported by the grant from the Center of the Development and Commercialization of New Technologies (Skolkovo Foundation) under Agreement No. 40113/0744-2024 dated September 19, 2024.

For citation: Khuri P.M., Klisheva A.D., Demchenko P.S., Bebekh K.V., Khodzitsky M.K., Vozianova A.V. Quantum-accelerated shortest path search on a graph in unsorted space. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2026, vol. 26, no. 3, pp. 652–661 (in Russian). doi: 10.17586/2226-1494-2026-26-3-652-661

Введение

В современном мире алгоритмы и структуры данных играют важную роль в информационных технологиях, логистике, транспорте, медицине. Одним из базовых алгоритмов, который находит применение во многих практических задачах, является алгоритм поиска кратчайшего пути на графе [1]. Граф — абстрактная структура данных, представляющая собой набор вершин (узлов) и ребер (связей между вершинами). В контексте поиска кратчайшего пути граф может представлять собой дорожную сеть, электрическую цепь, связи между компьютерами в сети и т. д. Задача поиска кратчайшего пути заключается в определении оптимального маршрута между двумя вершинами графа, минимизируя длину пути или количество промежуточных вершин. Эта задача является одной из 21 NP-трудных задач, которые были сформулированы Ричардом Карпом в 1972 г. [2]. Она все еще представляет собой серьезный вызов для науки, поскольку не существует полиномиального алгоритма, который мог бы обеспечить точное решение этой задачи. Решение задачи поиска кратчайшего пути на графе имеет ряд практических применений [3]. Например, в логистике и транспорте алгоритм может быть использован для определения оптимального маршрута доставки товаров или пассажиров [4, 5], в информационных технологиях — для оптимизации передачи данных в терагерцовых беспроводных системах коммуникаций 6G [6], в медицине — в планировании маршрута для скорой помощи или оптимизации пути пациента в больнице [7].

Одним из известных алгоритмов поиска кратчайшего пути из единичного источника S является алгоритм Беллмана–Форда [8]. Он применим к широкому классу

графов — с весами и без, с ребрами отрицательных весов и неотрицательных. Его пошаговая работа показана на рис. 1.

На каждом шаге графа (рис. 1, *a*) происходит пере-записывание информации, о конечной стоимости достижения вершины V исходя из суммарной стоимости посещенных ребер E графа. Сначала обновляется расстояние до вершины A по первому ребру, расстояние до вершины B по третьему ребру и расстояние до вершины C по пятому ребру (рис. 1, *b*). Затем обновляется расстояние до вершины A через вершину B и второе ребро и расстояние до вершины C через вершину A с нулевым весом и пятое ребро (рис. 1, *c*). Например, на рис. 1, *c* стоимость достижения вершины A стала равна нулю, так как найден оптимальный путь к ней через ребра 3 и 2 с суммарным нулевым весом. Способность обрабатывать ребра с отрицательным весом является основным преимуществом алгоритма Беллмана–Форда. Отметим, что алгоритм Беллмана–Форда имеет значительно большую сложность $O(V \times E)$, чем алгоритм Дейкстры [9], который содержит больше приложений, поскольку графы с отрицательными весами считаются редким случаем. Основное преимущество алгоритма Дейкстры — низкая сложность $O((V + E) \times \log_2 V)$, которая является почти линейной. Использовать данный алгоритм можно только на взвешенных графах с неотрицательными весами. Кроме того, при работе с плотными графами, где число ребер E близко к V^2 сложность становится $O((V + V^2) \times \log_2 V)$, и использование этого алгоритма нежелательно. Работа и описание алгоритма Дейкстры показаны на рис. 2 и в алгоритме 1.

Сначала помещается начальная вершина S в очередь с приоритетом, и устанавливается нулевое расстояние для нее (рис. 2, *a*). Затем она извлекается посещением

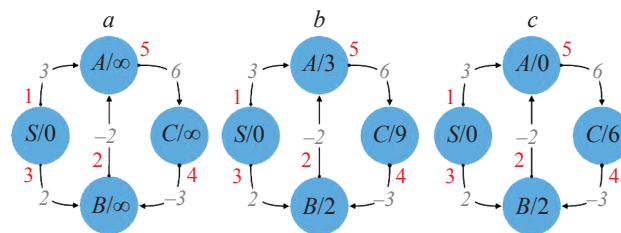


Рис. 1. Маршрутизация на графе с помощью алгоритма Беллмана–Форда.

Вес ребра (стоимость) прохода к следующей вершине по ребру (серые числа); порядок обхода всех вершин графа (красные числа); A, B, C, S — вершины графа; «/...» — вес ребра

Fig. 1. Routing on a graph using the Bellman-Ford algorithm

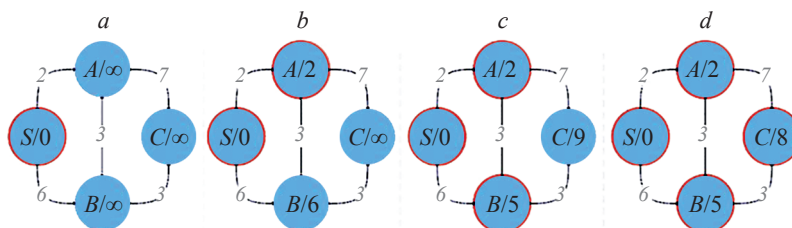


Рис. 2. Маршрутизация на графе с помощью алгоритма Дейкстры.

a, b, c, d — итерации расчета. Красными кругами выделены вершины с вычисленной длиной пути до вершины S

Fig. 2. Routing on a graph using Dijkstra's algorithm

ее соседей и обновлением расстояния до них (рис. 2, *b*). После этого извлекается вершина *A* из очереди приоритетов (поскольку она имеет наименьшее расстояние) и расстояние до ее соседей обновляется, если оно оказывается меньше (рис. 2, *c*). После этого помещаются обновленные значения снова в очередь приоритетов. Следующий узел, который необходимо извлечь, — узел *B* поскольку у него кратчайший путь. Как и на предыдущем шаге, обновляются расстояния до его соседей и при необходимости они помещаются в очередь (рис. 2, *d*). Наконец, извлекается вершина *C* из очереди, ей соответствует правильный кратчайший путь.

Алгоритм 1. Классический алгоритм Дейкстры для поиска кратчайших путей

Algorithm 1. Classical Dijkstra's Algorithm for Single-Source Shortest Path Problem

Вход: Граф $G = (V, E)$, начальная вершина $s \in V$

Выход: Массив кратчайших расстояний $dist$, массив предков $prev$

```

1: для каждой вершины  $v \in V$ :
2:    $dist[v] \leftarrow \infty$ 
3:    $prev[v] \leftarrow \text{None}$ 
4:    $visited[v] \leftarrow \text{False}$ 
5:  $dist[s] \leftarrow 0$ 
6:  $Q \leftarrow \text{createPriorityQueue}()$  //
Создание пустой очереди с приоритетом
7:  $Q.insert(s, 0)$ 
8:
9: пока  $Q$  не пуста:
10:   $u \leftarrow Q.extractMin()$ 
11:  если  $visited[u]$ :
12:    продолжить
13:   $visited[u] \leftarrow \text{True}$ 
14:
15:  для каждого ребра  $(u, v) \in E$ :
16:     $alt \leftarrow dist[u] + w(u, v)$ 
17:    если  $alt < dist[v]$ :
18:       $dist[v] \leftarrow alt$ 
19:       $prev[v] \leftarrow u$ 
20:    если не  $visited[v]$ :
21:       $Q.insert(v, dist[v])$ 
22:
23: вернуть  $(dist, prev)$ 

```

В практических приложениях в основном используются модификации алгоритма Дейкстры, а не его классическая реализация. Одним из ранних и известных применений алгоритма Дейкстры является решение задачи построения различных маршрутов на основе дорожных сетей [1]. Также алгоритм Дейкстры применяется в протоколе динамической маршрутизации изолированных сетей — Open Shortest Path First (OSPF)¹.

¹ RFC 2328 — Internet Standard [Электронный ресурс]. Режим доступа: <https://datatracker.ietf.org/doc/html/rfc2328>, свободный. Яз. англ. (дата обращения: 06.04.2025).

Основной недостаток алгоритма Дейкстры в том, что он не использует никакой информации о свойствах графа и искомого маршрута, а просто исследует все возможные пути. Однако существует способ стимулировать поиск путей в направлении цели и наказывать за отклонение от нее. Эта идея реализована в алгоритме A^* , который является обобщением алгоритма Дейкстры [10]. В отличие от алгоритма Дейкстры, алгоритм A^* фокусируется только на узлах назначения, а не на всех узлах графа, что делает его более эффективным. Поиск оптимального пути ведется, используя эвристическую функцию, которая отдает приоритет узлам, которые должны быть лучше других, учитывая такие параметры, как время, расстояние и т. д. Это позволяет оптимизировать процесс и выбирать наилучшие узлы. При прочих равных алгоритм A^* будет работать быстрее, чем алгоритм Дейкстры, даже если для его работы потребуется больше памяти и больше операций на узел, поскольку он исследует значительно меньше узлов, что является положительным свойством.

Развитие квантовых вычислений привело к появлению алгоритмов, способных решать определенные классы задач — такие как факторизация чисел (алгоритм Шора) и поиск в неструктурированных базах данных (алгоритм Гровера) — экспоненциально или квадратично быстрее, чем это возможно на классических компьютерах. Исходя из этого, появился новый подход к реализации ускорения, в рамках которого наиболее ресурсоемкие этапы классического алгоритма заменяются специализированными квантовыми процедурами. Применение подобного подхода к задачам оптимизации маршрутизации с большой вероятностью приведет к качественному прорыву: вместо локального расчета отдельных путей станет возможным глобальная оптимизация всей сети в реальном времени с непрерывной адаптацией к изменяющимся условиям. На данный момент создание полностью квантовых алгоритмов не представляется технически возможным. Ускорение классических алгоритмов квантовыми вставками, напротив, активно исследуется. Для задач оптимизации маршрутизации важным этапом является создание квантовой модификации алгоритма Дейкстры, потому что он лежит в основе большинства алгоритмов маршрутизации. В работе [11] исследованы возможности квантового ускорения классических графовых алгоритмов и описана идея квантовой модификации алгоритма Дейкстры. На момент написания [11] техническая реализация квантово-ускоренного алгоритма была полностью невозможна. В работе [12] предложена реализация модифицированного алгоритма Дейкстры на квантовом симуляторе Quantum Computing Language (QCL), который на данный момент устарел. Код, написанный на этом фреймворке, нельзя запускать на настоящем квантовом компьютере. В настоящей работе рассмотрена оптимизация алгоритма Дейкстры с квантовым ускорением. Проведена оценка устойчивости квантового алгоритма и величины искажений в схеме с различным количеством кубитов. Алгоритм был запущен на квантовом компьютере IBM.

Метод поиска кратчайшего пути на графе в несортированном пространстве

Выбор вершины с наименьшим расстоянием, в которую релаксирует классический алгоритм Дейкстры, приводит к дополнительной задаче поиска некоторого элемента в неупорядоченной базе данных. Более формально это можно описать так: для данных индексов $0, 1, 2, \dots, N$ и некоторого предиката f , ставящего в соответствие каждому индексу значение 1 или 0, необходимо найти такой индекс j , что $f(j) = 1$. Очевидно, что наивный алгоритм для решения этой задачи требует $O(N)$ операций. Например, для решения задачи эффективной маршрутизации с помощью алгоритма Дейкстры, необходимость простого перебора значений со сложностью $O(N)$ операций возникает каждый раз, когда алгоритм приходит в новую вершину и выбирает ребро, по которому пойдет в следующую. Если он ищет ребро с наименьшим весом, то нужно будет перебрать все ребра и сравнить их между собой, прежде чем найти наименьшее.

Квантовый алгоритм Гровера, предложенный в работе [13], решает задачу поиска за $O(\sqrt{N})$ операций. С помощью алгоритма Гровера можно модифицировать рассмотренные в разделе «Введение» классические алгоритмы маршрутизации, получив гибридный квантово-классический алгоритм с вероятностным выбором оптимального маршрута на графе. Тогда поиск нужного ребра будет проходить с квантовым ускорением. С помощью самого простого двухкубитового алгоритма Гровера можно обойти как небольшой граф из 4 вершин, так и большие графы. Ожидается, что с помощью предложенной модификации алгоритм Гровера на двух кубитах сможет применяться для успешной маршрутизации на графах, подобных представленным на рис. 3. Единственное обязательное условие в таких сетях — количество ребер, исходящих из каждой вершины должно быть не больше, чем число состояний системы $2n$, где n — количество кубитов. Для самой простой квантовой системы из двух кубит число состояний равно 4.

Рассмотрим множество A , пусть оно представляет собой множество элементов, для которых предикат f возвращает 1, и $d = |A|$ [14, 15]. Для того чтобы сформулировать саму проблему в терминах квантовых вычислений, необходимо предположить, что имеется также так называемый «квантовый оракул», осуществляющий преобразование вида:

$$|x\rangle \rightarrow -|x\rangle, \quad x \in A \quad |x\rangle, \quad x \notin A.$$

Идея алгоритма заключается в применении итерации Гровера.

Исходными параметрами алгоритма Гровера являются квантовый оракул O и размер пространства поиска. Для выполнения алгоритма необходимо выполнить следующие шаги.

Шаг 1. Создать квантовый регистр с достаточным количеством кубитов для индексации пространства поиска.

Шаг 2. Установить регистр в состояние $|0\rangle$.

Шаг 3. Применить к регистру преобразование Уолша-Адамара H .

Шаг 4. Пока не произведено необходимое количество итераций Гровера:

- применить оператор квантового оракула O ;
- применить оператор инверсии (диффузии) P .

Шаг 5. Применить итерацию Гровера требуемое количество раз.

Шаг 6. Провести измерение над регистром.

Преобразование Уолша-Адамара используется для получения суперпозиции значений кубита. Применяя на каждом кубите, данное преобразование позволяет получить 2^n состояний при измерении с равной вероятностью. Операция инверсии (диффузии) называется отражением от среднего [15, 16]. Иначе говоря, эта операция многократно увеличивает значение амплитуды вероятности, если оно было значительно меньше среднего значения. В квантовых вычислениях амплитуда вероятности — комплексное число, квадрат модуля которого дает вероятность обнаружения квантовой системы в данном состоянии. Сумма квадратов амплитуд всех состояний равна 1.

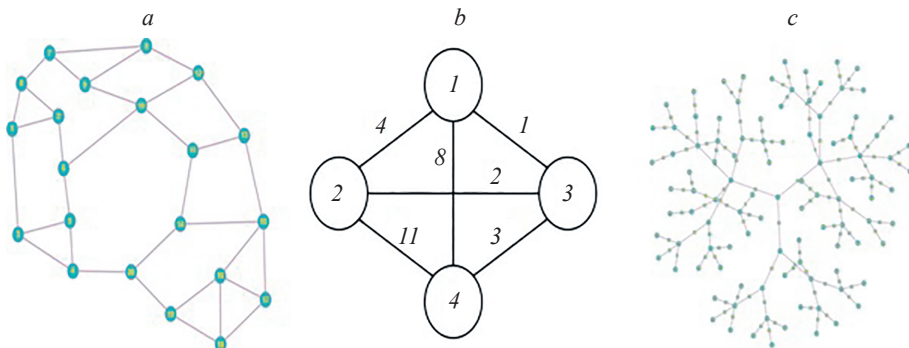


Рис. 3. Примеры видов графов, подходящих для применения алгоритма Гровера на двух кубитах: регулярная ячеистая сеть (граф-решетка) с ограничением степени вершин до 4 (a); малый взвешенный граф из четырех вершин с произвольными весами ребер (b); иерархическая древовидная сеть с ограниченным ветвлением (c)

Fig. 3. Examples of graph types suitable for Grover's algorithm on two qubits: a regular mesh network (lattice graph) with vertex degrees restricted to 4 (a); a small weighted graph of four vertices with arbitrary edge weights (b); a hierarchical tree network with restricted branching (c)

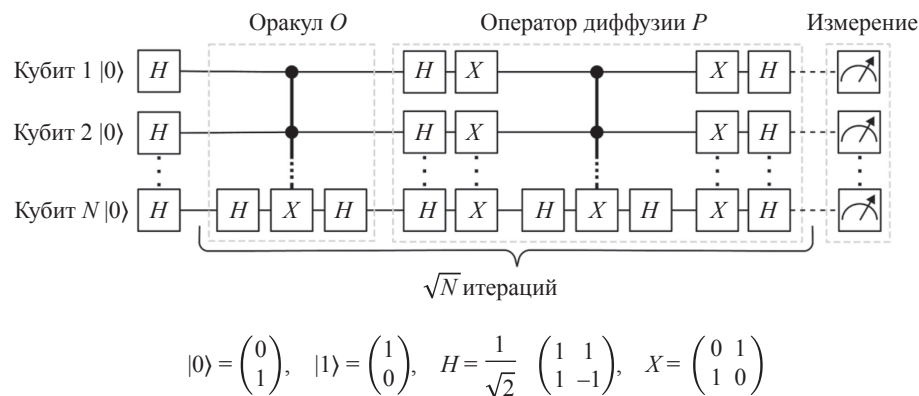


Рис. 4. Блок-схема алгоритма Гровера

$|0\rangle, |1\rangle$ — состояния квантовой системы; гейт «H» — оператор (преобразование) Уолша–Адамара; гейт «X» — оператор инверсии амплитуд состояний квантовой системы

Fig. 4. Grover's algorithm flowchart

Оценка устойчивости квантового алгоритма и величины искажений в схеме с различным количеством кубитов

Для успешного практического применения квантового алгоритма необходимо оценить его устойчивость к возможным ошибкам. Первым шагом алгоритма Гровера является применение оператора Уолша–Адамара, который преобразует исходное состояние в состояние суперпозиции. Чтобы проверить, как алгоритм справляется с ошибками, были внесены искажения в состояния, обрабатываемые на каждой итерации Гровера. Это позволяет имитировать ситуацию, когда оператор Уолша–Адамара работает с небольшими погрешностями, что приводит к различиям в значениях амплитуд вероятностей состояний перед применением итерации Гровера. Независимо от того, какое состояние отмечает оператор оракула, искажение увеличивает начальную амплитуду вероятности одних случайных состояний и уменьшает амплитуды других. В идеальном трехкубитном алгоритме Гровера преобразование Уолша–Адамара приводит к тому, что амплитуды вероятностей всех состояний становятся одинаковыми. Однако если внести искажения, которые влияют на амплитуду вероятности состояния, это

может существенно исказить начальное распределение амплитуд вероятностей состояний, на основе которого происходят все вычисления, и значительно повлиять на конечный результат работы всего алгоритма. Визуализация вносимых в работу оператора Уолша–Адамара искажений показана на рис. 5.

Наличие открытого доступа к четырехкубитному компьютеру IBM Quantum Platform (IBM QP)¹ дало возможность проверить достоверность предположения о характере и порядке вносимых в работу алгоритма возмущений и посмотреть на реальные результаты вычислений оператора Уолша–Адамара, искажения которого моделируются в настоящей работе. На рис. 6 представлены схема и результаты работы преобразования оператора Уолша–Адамара.

Квантовый компьютер IBM QP показывает амплитудное распределение вероятностей, полученное после применения одного оператора Уолша–Адамара на трехкубитовой схеме. Приводя в соответствие с разработанным способом оценки величины вносимых возмущений, такое распределение можно оценить как

¹ IBM Quantum Platform [Электронный ресурс]. Режим доступа: <https://quantum.ibm.com/composer>, свободный. Яз. англ. (дата обращения: 06.04.2025).

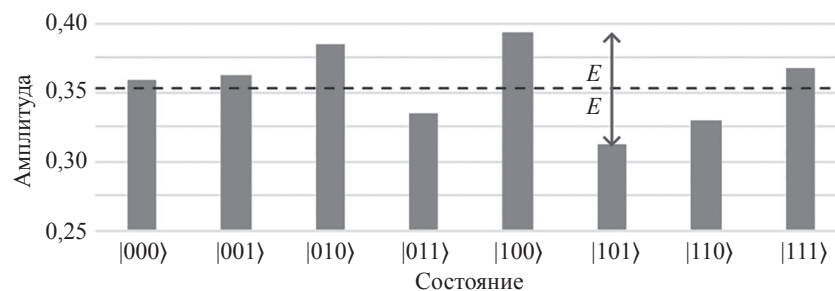


Рис. 5. Графическое представление распределения амплитуд состояний после применения оператора Уолша–Адамара.

E — отклонение реальной величины амплитуды от величины амплитуды идеального случая

Fig. 5. Graphical representation of the distribution of state amplitudes after applying the Hadamard operator.

E — deviation of the actual amplitude value from the ideal case amplitude value

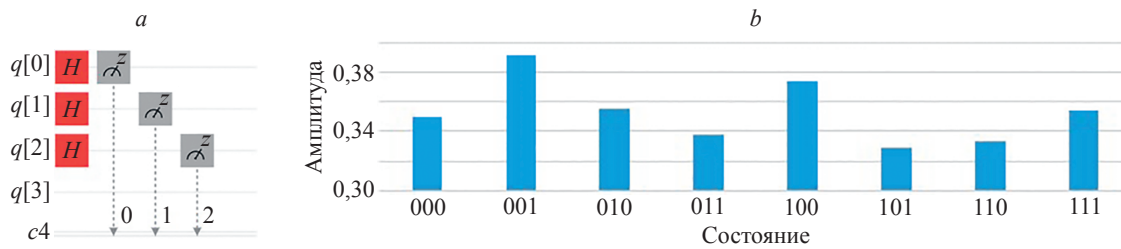


Рис. 6. Схема (a) и распределение амплитуд вероятностей состояний (b) трехкубитового квантового оператора Уолша–Адамара на компьютере IBM QP.

$q[i]$ — кубит с индексом i ; c — классический регистр памяти; серый гейт — измерение кубита в базисе

Fig. 6. Scheme (a) and probability amplitude distribution of states (b) of the three-qubit Walsh-Hadamard quantum operator on an IBM QP computer

7–10 % искажений. Показано, что с возрастанием интенсивности помех вероятность обнаружения искомого элемента уменьшается, а время, затрачиваемое на его поиск, увеличивается. При применении алгоритма Гровера для оптимизации пути в квантовой сети эти помехи становятся причиной увеличения времени поиска оптимального пути и повышения вероятности выбора неверного маршрута.

Схема из трех кубитов (рис. 6, a) имеет всего 8 состояний и требует двух итераций. В то же время, например, 11 кубитов позволяют обрабатывать более 2000 состояний, требуя 35 итераций. Оценим устойчивость алгоритма Гровера, составив квантовую схему оракула для алгоритма Гровера из 9, 10 и 11 кубитов. В результате выполнения алгоритма Гровера с соответствующими оракулами и искусственно созданными возмущениями, имитирующими небольшую погрешность оператора Уолша–Адамара была получена оценка устойчивости к искажениям для 9–11 кубитов (рис. 7). Данную оценку устойчивости можно применить к схемам с большим количеством кубит, устойчивость которых не может быть исследована вследствие недостаточной вычислительной мощности.

Как и ожидалось, вероятность обнаружения определенного состояния снижается по мере увеличения уровня помех. При этом следует отметить, что уве-

личение количества кубитов и, соответственно, количества итераций делает алгоритм более устойчивым. Обнаружено, что устойчивость алгоритма Гровера для схем с большим количеством кубитов имеет схожий характер. В приведенном примере работы квантовых схем на 9, 10 и 11 кубитах различие в значениях вероятностей верного срабатывания отличается менее, чем на 10^{-3} . Следовательно, устойчивость очень слабо зависит от количества кубитов в многокубитной квантовой схеме.

Оптимизация алгоритма Дейкстры с квантовым ускорением

После исследования на устойчивость применим квантовый алгоритм для решения задачи поиска в несортированном пространстве с квантовым ускорением. Так как для поиска кратчайшего пути с одним источником применяется алгоритм Дейкстры, то необходимо выполнить его модификацию. Наиболее очевидным способом модификации алгоритма Дейкстры является замена очереди с приоритетом в классической версии алгоритма на некоторую квантовую структуру данных, находящую минимальный элемент для следующей итерации алгоритма (алгоритм 2).

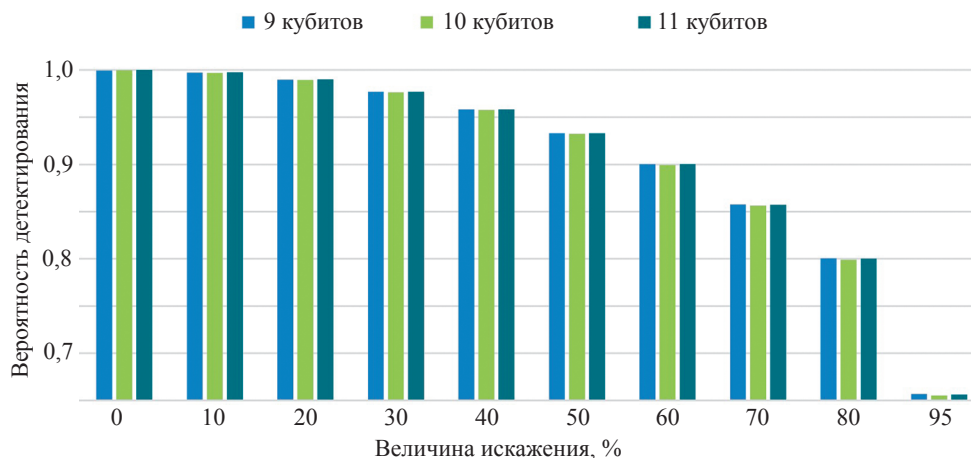


Рис. 7. Зависимость вероятности детектирования верного состояния от величины искажений при различном числе кубитов

Fig. 7. Dependence of the probability of detecting the correct state on the magnitude of disturbances for different numbers of qubits

Алгоритм 2. Алгоритм Дейкстры с квантовым ускорением

Algorithm 2. Dijkstra's Algorithm with Quantum Acceleration

Вход: Граф $G = (V, E)$, начальная вершина $s \in V$

Выход: Массив кратчайших расстояний $dist$

```

1: для каждой вершины  $v \in V$ :
2:    $dist[v] \leftarrow \infty$ 
3:    $visited[v] \leftarrow False$ 
4:  $dist[s] \leftarrow 0$ 
5:
6: пока существует необработанная вершина:
7:   // Квантовый этап: поиск вершины с минимальным  $dist$  среди непосещённых
8:    $u \leftarrow QuantumFindMin(dist, visited)$ 
9:
10:  если  $u = NULL$ : прервать цикл
11:   $visited[u] \leftarrow True$ 
12:
13:  // Классический этап: релаксация рёбер
14:  для каждого ребра  $(u, v) \in E$ :
15:     $alt \leftarrow dist[u] + w(u, v)$ 
16:    если  $alt < dist[v]$ :
17:       $dist[v] \leftarrow alt$ 
18:
19: вернуть  $dist$ 

```

Рассмотрены асимптотики:

— «массив»: $O(V + E)V$;

— «двоичная куча»: $O(V + E)\log_2 V$;

— «Фибоначчиева куча»: $O((Vd + E)\log_d V)$, где $d = \frac{E}{V}$;

— «массив с квантовым поиском»: порядка $O(V + E)\sqrt{V}$.

На рис. 8 изображена асимптотика работы алгоритмов в зависимости от используемой структуры данных в очереди с приоритетом.

Несмотря на то, что квантовый поиск не дает потенциального прироста в асимптотике, можно использовать иную квантовую модификацию алгоритма Дейкстры. А именно, вместо поддержания и поиска минимума в очереди с приоритетом, использовать квантовый алгоритм сразу для поиска вершины, которую необходимо добавить в дерево кратчайших путей, что даст ускорение обхода графа алгоритмом Дейкстры.

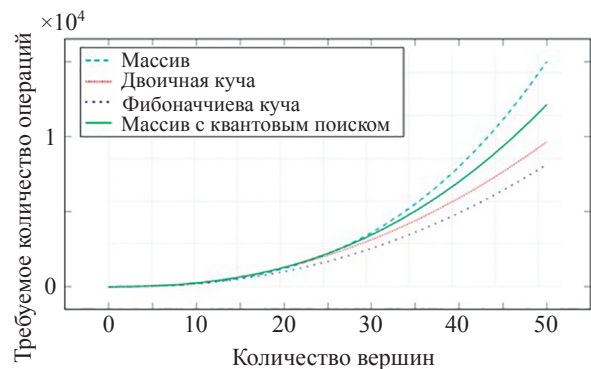


Рис. 8. Асимптотика работы алгоритмов

Fig. 8. Asymptotics of the algorithms

Кроме этого, структуры данных типа «куча» имеют недостатки.

Во-первых, во многих приложениях, в том числе беспроводной коммуникации, элементы системы постоянно изменяются (изменение положения, физических или финансовых значений и тому подобное). Это приводит к регулярной перестройке графа. Следовательно, при использовании структуры данных типа «куча», например бинарная, потребуется перестройка графа для соблюдения основного свойства кучи, что занимает время. Во-вторых, в варианте кучи, такой как Фибоначчиева куча, необходимо хранить большое количество информации, что ускоряет операции, но увеличивает требуемую память для оперирования и хранения. Использование параллелизма квантовых алгоритмов позволяет упростить структуру данных, оптимизировав операции над данными в кучах и объем занимаемой памяти.

Заключение

Для практического применения в задачах поиска кратчайшего пути на графах важно направление поиска, так как это напрямую влияет на эффективность алгоритма. В таких задач оптимально использовать алгоритм A^* и его квантовые модификации. Квантовые вычисления могут значительно сократить время выполнения сложных задач, используя суперпозицию и запутанность для параллельной обработки нескольких состояний одновременно, что позволяет находить решения быстрее, чем классические алгоритмы. Таким образом, квантовые модификации алгоритма A^* предоставляют мощные инструменты для решения сложных задач маршрутизации, значительно сокращая время вычислений и повышая точность результатов. Представленная реализация квантовой модификации Дейкстры является важным шагом для разработки квантовой модификации алгоритма A^* .

Литература

1. Zhan F.B., Noon C.E. Shortest path algorithms: an evaluation using real road networks // *Transportation Science*. 1998. V. 32. N 1. P. 65–73. doi: 10.1287/trsc.32.1.65
2. Karp R.M. Reducibility among combinatorial problems // *50 Years of Integer Programming 1958–2008*. 2010. P. 219–241. doi: 10.1007/978-3-540-68279-0_8
3. Haeupler B., Hladík R., Rozhoň V., Tarjan R.E., Tetěk J. Universal optimality of Dijkstra via beyond-worst-case heaps // *Proc. of the IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*. 2024. P. 2099–2130. doi: 10.1109/focs61266.2024.00125
4. Wei Z. High performance computing simulation of intelligent logistics management based on shortest path algorithm // *Computational Intelligence and Neuroscience*. 2022. V. 2022. P. 7930553. doi: 10.1155/2022/7930553
5. Chen R. Dijkstra's shortest path algorithm and its application on bus routing // *Advances in Economics, Business and Management Research*. 2022. P. 321–325. doi: 10.2991/aebmr.k.220502.058
6. Warriar A., Aljaburi L., Whitworth H., Al-Rubaye S., Tsourdos A. Future 6G communications powering vertical handover in non-terrestrial networks // *IEEE Access*. 2024. V. 12. P. 33016–33034. doi: 10.1109/ACCESS.2024.3371906
7. Kurniawan F., Widyanto R.A., Sukmasetya P. Dijkstra algorithm implementation to determine the shortest route to hospital: a case study in Magelang district Indonesia // *E3S Web of Conferences*. 2024. V. 500. P. 01004. doi: 10.1051/e3sconf/202450001004
8. Bellman R. On a routing problem // *Quarterly of Applied Mathematics*. 1958. V. 16. № 1. P. 87–90. doi: 10.1090/qam/102435
9. Dijkstra E.W. A note on two problems in connexion with graphs // *Numerische Mathematik*. 1959. V. 1. N 1. P. 269–271. doi: 10.1007/BF01386390
10. Hart P.E., Nilsson N.J., Raphael B. A formal basis for the heuristic determination of minimum cost paths // *IEEE transactions on Systems Science and Cybernetics*. 1968. V. 4. N 2. P. 100–107. doi: 10.1109/TSSC.1968.300136
11. Dürr C., Heiligman M., Høyer P., Mhalla M. Quantum query complexity of some graph problems // *SIAM Journal on Computing*. 2006. V. 35. N 6. P. 1310–1328. doi: 10.1137/050644719
12. Ray P. Quantum simulation of Dijkstra's algorithm // *International Journal of Advance Research in Computer Science and Management Studies*. 2014. V. 2. N 9. P. 30–43.
13. Grover L.K. A fast quantum mechanical algorithm for database search // *Proc. of the 28th Annual ACM Symposium on Theory of Computing (STOC)*. 1996. P. 212–221. doi: 10.1145/237814.237866
14. Nielsen M.A., Chuang I.L. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. 702 p. doi: 10.1017/CBO9780511976667
15. Nakahara M., Ohmi T. *Quantum Computing: from Linear Algebra to Physical Realizations*. CRC Press, 2008. 438 p.
16. Linington I.E., Ivanov P.A., Vitanov N.V. Quantum search in a nonclassical database of trapped ions // *Physical Review A*. 2009. V. 79. N 1. P. 012322. doi: 10.1103/PhysRevA.79.012322

Авторы

Хури Паскаль Мишалевич — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, <https://orcid.org/0009-0006-9817-2688>, paskal_khuri@mail.ru

Клишева Алина Дмитриевна — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, <https://orcid.org/0009-0008-4860-7019>, klishevaad@gmail.com

Демченко Пётр Сергеевич — кандидат технических наук, директор по научной работе, ООО «Терагерцовая Фотоника», Санкт-Петербург, 191167, Российская Федерация, <https://orcid.org/0000-0003-4948-2023>, petr.s.demchenko@gmail.com

Бибех (Губайдуллина) Ксения Витальевна — преподаватель, ИП Бибех А.А., Санкт-Петербург, 198411, Российская Федерация; аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, <https://orcid.org/0009-0001-7116-9028>, ksenya-gbd@yandex.ru

Ходзинский Михаил Константинович — кандидат физико-математических наук, генеральный директор, ООО «Терагерцовая Фотоника», Санкт-Петербург, 191167, Российская Федерация, <https://orcid.org/0000-0001-7261-8350>, petr.s.demchenko@gmail.com

References

1. Zhan F.B., Noon C.E. Shortest path algorithms: an evaluation using real road networks. *Transportation Science*, 1998, vol. 32, no. 1, pp. 65–73. doi: 10.1287/trsc.32.1.65
2. Karp R.M. Reducibility among combinatorial problems. *50 Years of Integer Programming 1958–2008*, 2010, pp. 219–241. doi: 10.1007/978-3-540-68279-0_8
3. Haeupler B., Hladík R., Rozhoň V., Tarjan R.E., Tetěk J. Universal optimality of Dijkstra via beyond-worst-case heaps. *Proc. of the IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, 2024, pp. 2099–2130. doi: 10.1109/focs61266.2024.00125
4. Wei Z. High performance computing simulation of intelligent logistics management based on shortest path algorithm. *Computational Intelligence and Neuroscience*, 2022, vol. 2022, pp. 7930553. doi: 10.1155/2022/7930553
5. Chen R. Dijkstra's shortest path algorithm and its application on bus routing. *Advances in Economics, Business and Management Research*, 2022, pp. 321–325. doi: 10.2991/aebmr.k.220502.058
6. Warriar A., Aljaburi L., Whitworth H., Al-Rubaye S., Tsourdos A. Future 6G communications powering vertical handover in non-terrestrial networks. *IEEE Access*, 2024, vol. 12, pp. 33016–33034. doi: 10.1109/ACCESS.2024.3371906
7. Kurniawan F., Widyanto R.A., Sukmasetya P. Dijkstra algorithm implementation to determine the shortest route to hospital: a case study in Magelang district Indonesia. *E3S Web of Conferences*, 2024, vol. 500, pp. 01004. doi: 10.1051/e3sconf/202450001004
8. Bellman R. On a routing problem. *Quarterly of Applied Mathematics*, 1958, vol. 16, no. 1, pp. 87–90. doi: 10.1090/qam/102435
9. Dijkstra E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959, vol. 1, no. 1, pp. 269–271. doi: 10.1007/BF01386390
10. Hart P.E., Nilsson N.J., Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 1968, vol. 4, no. 2, pp. 100–107. doi: 10.1109/TSSC.1968.300136
11. Dürr C., Heiligman M., Høyer P., Mhalla M. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 2006, vol. 35, no. 6, pp. 1310–1328. doi: 10.1137/050644719
12. Ray P. Quantum simulation of Dijkstra's algorithm. *International Journal of Advance Research in Computer Science and Management Studies*, 2014, vol. 2, no. 9, pp. 30–43.
13. Grover L.K. A fast quantum mechanical algorithm for database search. *Proc. of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, 1996, pp. 212–221. doi: 10.1145/237814.237866
14. Nielsen M.A., Chuang I.L. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010, 702 p. doi: 10.1017/CBO9780511976667
15. Nakahara M., Ohmi T. *Quantum Computing: from Linear Algebra to Physical Realizations*. CRC Press, 2008, 438 p.
16. Linington I.E., Ivanov P.A., Vitanov N.V. Quantum search in a nonclassical database of trapped ions. *Physical Review A*, 2009, vol. 79, no. 1, pp. 012322. doi: 10.1103/PhysRevA.79.012322

Authors

Paskal M. Khuri — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, <https://orcid.org/0009-0006-9817-2688>, paskal_khuri@mail.ru

Alina D. Klisheva — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, <https://orcid.org/0009-0008-4860-7019>, klishevaad@gmail.com

Petr S. Demchenko — PhD, Director for Science, Terahertz Photonics, LLC, Saint Petersburg, 191167, Russian Federation, <https://orcid.org/0000-0003-4948-2023>, petr.s.demchenko@gmail.com

Ksenia V. Bebekh (Gubaidullina) — Lecturer, IE Bebekh A.A., Saint Petersburg, 198411, Russian Federation; PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, <https://orcid.org/0009-0001-7116-9028>, ksenya-gbd@yandex.ru

Mikhail K. Khodzitsky — PhD (Physics & Mathematics), Chief Executive Officer, Terahertz Photonics, LLC, Saint Petersburg, 191167, Russian Federation, <https://orcid.org/0000-0001-7261-8350>, petr.s.demchenko@gmail.com

Возианова Анна Викторовна — кандидат физико-математических наук, исполнительный директор, ООО «Терагерцовая Фотоника», Санкт-Петербург, 191167, Российская Федерация; доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, [sc 17436111400](https://orcid.org/0000-0002-4550-7169), <https://orcid.org/0000-0002-4550-7169>, anna-vozianova@yandex.ru

Anna V. Vozianova — PhD (Physics & Mathematics), Executive Officer, Terahertz Photonics, LLC, Saint Petersburg, 191167, Russian Federation; Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, [sc 17436111400](https://orcid.org/0000-0002-4550-7169), <https://orcid.org/0000-0002-4550-7169>, anna-vozianova@yandex.ru

Статья поступила в редакцию 09.04.2025
Одобрена после рецензирования 29.04.2026
Принята к печати 25.05.2026

Received 09.04.2025
Approved after reviewing 29.04.2026
Accepted 25.05.2026



Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»