

THE IMPLEMENTATION OF QUANTUM STATE PREPARATION ALGORITHMS, CONSIDERING THE LIMITATIONS OF MODERN QUANTUM COMPUTERS

A. D. Ivlev, A. V. Linirov

Lobachevsky State University of Nizhny Novgorod,
603950, Nizhny Novgorod, Russia

DOI: 10.24412/2073-0667-2025-2-33-47

EDN: FCJOXG

Preparation of an arbitrary initial state of a system of qubits is an important and actual problem of quantum computing. Its importance because many quantum algorithms require preloading classical data to quantum devices, such as quantum neural networks [1] or solving systems of linear equations [3]. The computational cost of data loading can limit potential quantum acceleration, while the accuracy of the prepared state directly impacts the correctness of algorithmic outcomes. This paper analyzes three contemporary approaches to initial quantum state preparation, their translation into QASM code under the constraints of modern quantum hardware, considering the topology and basic sets of gates. The first of them is a modified algorithm for preparing the quantum state using controlled gates [5]. This algorithm does not use additional qubits, and the size and depth of its quantum circuit are estimated at $O(2^n)$. The main idea of the algorithm is that iteratively, considering the first $i + 1$ qubits at the i step, we distribute 1 (corresponding to the initial state, where all qubits are in the state $|0\rangle$) over the entire vector in the required proportions and add a phase at the last iteration. The second algorithm called “divide and conquer”, proposed in the article [5], is an optimization of the previous algorithm, which, by using $2^n - n - 1$ additional qubits, reduces the asymptotic depth of the circuit to $O(n^2)$, but maintaining its size is $O(2^n)$. The idea is to expand the algorithm in width instead of length due to the tree structure of the algorithm and the gates *CSWAP*. This arrangement makes it possible to perform quantum gates in parallel. The third algorithm is the preparation of a quantum state by solving a more general problem of approximation of a unitary operator using template schemes. This approach, unlike the previous ones, is not theoretically accurate and requires resource-intensive classical calculations. In this paper, all algorithms were implemented considering the limitations of the modern `ibm_sherbrooke` quantum computer [8]. To do this, the basic gates were approximated from theoretical calculations using available gates, the CNOT gates were applied considering the topology, and the problem of the global phase was considered separately. The algorithmic implementation of algorithms for preparing the initial quantum state showed in all cases a shift in the estimation of the asymptotic of the circuit size and its depth for the worse. The algorithm showed the best result in the form of an approximation of the unitary operator with $O(n2^n)$ and $O(2^n)$ accordingly, but it requires preliminary classical preparation of $3n2^n$ circuit parameters. All algorithms showed low accuracy when algorithmically implemented on the architecture of the `ibm_sherbrooke` quantum computer. And at the moment, due to the high asymptotic of practical implementation, they are explicitly suitable only for preparing the states of systems of no more than three qubits. Perhaps by applying various optimization

The work was carried out with the support of the Research and Education Mathematical Center “Mathematics of Future Technologies”, contract No. 075-02-2024-1439.

methods and error compensation algorithms, it will be possible to increase this number, but without new approaches that reduce the asymptotic size of the circuit or its depth in practice, one should not expect significant improvement. At the same time, it is worth remembering that all the described algorithms allow you to prepare an arbitrary state, that is, they are universal.

Key words: quantum computing, quantum state preparation, translation, QASM.

References

1. Hur T., Kim L., Park D. K. Quantum convolutional neural network for classical data classification // *Quantum Machine Intelligence*. 2022, 4(1), 3.
2. Barenco A. et al. Stabilization of quantum computations by symmetrization // *SIAM Journal on Computing*. 1997, 26(5), 1541–1557.
3. Harrow A. W., Hassidim A., Lloyd S. Quantum algorithm for linear systems of equations // *Physical review letters*. 2009, 103(15), 150502.
4. Sun X. et al. Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2023, 42(10), 3301–3314.
5. Araujo I. F. et al. A divide-and-conquer algorithm for quantum state preparation // *Scientific reports*. 2021, 11(1), 6329.
6. Sim S., Johnson P. D., Aspuru-Guzik A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms // *Advanced Quantum Technologies*. 2019, 2(12), 1900070.
7. Cross A. W. et al. Open quantum assembly language // *arXiv preprint arXiv:1707.03429*. 2017.
8. IBM Quantum. [Electron. Res.]: <https://quantum.ibm.com>.
9. Barenco A. et al. Elementary gates for quantum computation // *Physical review A*. 1995, 52(5), 3457.
10. Shende V. V., Bullock S. S., Markov I. L. Synthesis of quantum logic circuits // *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*. 2005, 272–275.
11. Qiskit. [Electron. Res.]: <https://github.com/Qiskit>.
12. Luo J., Li L. Circuit complexity of sparse quantum state preparation // *arXiv preprint arXiv:2406.16142*. 2024.

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПОДГОТОВКИ НАЧАЛЬНОГО СОСТОЯНИЯ СИСТЕМЫ КУБИТОВ С УЧЕТОМ ОГРАНИЧЕНИЙ СОВРЕМЕННЫХ КВАНТОВЫХ КОМПЬЮТЕРОВ

А. Д. Ивлев, А. В. Линева

Национальный исследовательский Нижегородский государственный университет
им. Н. И. Лобачевского,
603950, Нижний Новгород, Россия

УДК 519.688

DOI: 10.24412/2073-0667-2025-2-33-47

EDN: FCJOXG

Подготовка произвольного начального состояния системы кубитов является важной и актуальной задачей квантовых вычислений. Ее значимость обуславливается тем, что многие квантовые алгоритмы требуют предварительной загрузки классических данных на квантовые устройства. Соответственно, вычислительные затраты на загрузку данных могут накладывать ограничения на возможное квантовое ускорение, а точность подготовленного состояния влияет на корректность результатов алгоритмов. В данной работе рассмотрены несколько актуальных подходов к подготовке начального состояния, а также их алгоритмическая трансляция в язык QASM с учетом топологии и базисных наборов гейтов современных квантовых компьютеров. Проведены эксперименты на симуляторе идеального квантового компьютера и симуляторе зашумленного квантового компьютера с характеристиками реальной системы, по результатам которых проведено сравнение теоретических и практических оценок точности, глубины и размеров квантовых схем рассматриваемых алгоритмов.

Ключевые слова: квантовые вычисления, подготовка квантового состояния, трансляция, QASM.

Введение. Подготовка произвольного начального состояния системы кубитов является важной и актуальной задачей квантовых вычислений. Ее значимость обуславливается тем, что многие квантовые алгоритмы требуют предварительной загрузки классических данных на квантовые устройства, например квантовые нейронные сети [1], контроль результатов с помощью гейта SWAP [2] или решение систем линейных уравнений [3]. Соответственно, вычислительные затраты на загрузку данных могут накладывать ограничения на возможное квантовое ускорение, а точность подготовленного состояния влияет на корректность результатов алгоритмов.

Задается нормированный вектор состояния системы кубитов $|\psi\rangle \in C^{2^n}$, где n — количество кубитов. Необходимо построить квантовую схему, которая задает унитарное преобразование U , которое переводит исходное состояние $|0\rangle^{\otimes n}$ в желаемое $|\psi\rangle$ ($U|0\rangle^{\otimes n} = |\psi\rangle$).

Работа выполнена при поддержке Научно-образовательного математического центра «Математика технологий будущего», контракт № 075-02-2024-1439.

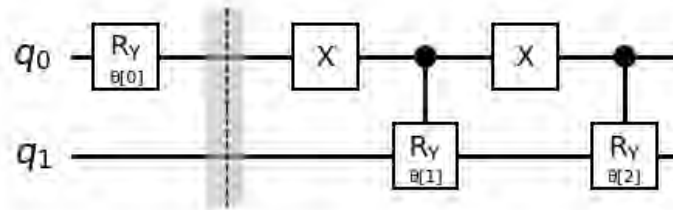


Рис. 1. Квантовая схема первых двух шагов построения вектора модулей

Из данной постановки видно, что первый столбец данного оператора всегда будет вектором состояния, которое мы хотим построить. Соответственно, для решения данной задачи в общем случае достаточно брать унитарный оператор U с первым столбцом $|\psi\rangle$.

Известно, что размер квантовых схем подготовки n -кубитного квантового состояния оценивается $O(2^n)$, при использовании однокубитных гейтов и гейта CNOT в качестве универсального базового набора [4]. Но при этом теоретическая глубина схем может достигать $O(n^2)$ при использовании дополнительных кубитов [5].

В данной работе рассмотрены некоторые актуальные подходы, описанные в статьях [5, 6] и их алгоритмическая трансляция в язык QASM [7] с учетом топологии и базисных наборов гейтов, на примере архитектуры квантовых компьютеров IBM [8]. На основе экспериментов была оценена точность алгоритмов и проведено сравнение теоретических оценок размеров и глубины идеальных квантовых схем и их практической реализации.

1. Подготовка квантового состояния с помощью управляемых гейтов поворота. В данном разделе будет рассмотрен модифицированный алгоритм подготовки квантового состояния с помощью управляемых гейтов [5]. Данный алгоритм не использует дополнительных кубитов, а размер и глубина его квантовой схемы оцениваются $O(2^n)$.

Основная идея алгоритма заключается в том, что мы представляем все комплексные числа вектора состояния в показательном виде, рассматриваем подзадачу построения вектора из действительных чисел, и далее — подзадачу наложения фазы на полученный вектор.

Первая подзадача сводится к распределению 1 (соответствует начальному состоянию, где все кубиты в состоянии $|0\rangle$) по всему вектору в необходимых пропорциях. Разделим первую часть алгоритма на n итераций, на каждой из которых мы будем рассматривать только первые 2^i значений вектора ($i = \overline{1, n}$). То есть на каждой итерации мы будем задействовать только первые i кубитов, причем старший из них изначально будет находиться в состоянии 0. Задача итерации — распределить значения из первой половины на весь подвектор из 2^i значений.

Разобьем 2^i значений на пары по их позициям $(j, 2^{(i-1)} + j)$, $j = \overline{0, 2^{(i-1)} - 1}$. Можно заметить, что данные значения описывают состояния, которые отличаются только значением старшего, для данной итерации, кубита. Следовательно, мы можем выделить данные состояния, используя множественный контроль [8], и применить гейт $Ry(\psi)$ к каждой паре по отдельности. То есть мы свели задачу итерации к переходу пар из состояний вида $(c, 0)$ к (a, b) .

Пример первых двух итераций построения вектора модулей приведен ниже на рис. 1.

Для завершения основной части алгоритма нам осталось определить углы ϕ для применения гейтов $Ry(\phi)$. Рассмотрев данный процесс для одной пары, получим, что

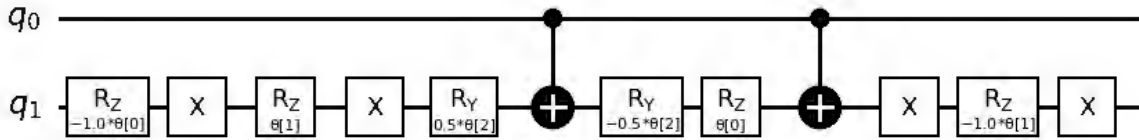


Рис. 2. Блок квантовой схемы, выполняющий поворот вместе с изменением фазы

$a = c * \cos(\frac{\phi}{2})$, $b = c * \sin(\frac{\phi}{2})$. Тогда $\phi = 2 \arccos(\frac{a}{c})$ при $c \neq 0$ ($\phi = 0$ при $c = 0$). На последней итерации ($i = n$) мы знаем a , b для каждой пары, так как это известные нам модули значений желаемого вектора состояний $|\psi\rangle$. Поэтому для определения углов мы будем идти от последней итерации к первой, «сворачивая» вектор к нулевому состоянию, используя соотношение $c = \sqrt{a^2 + b^2}$.

Вторая подзадача в общем случае может быть решена с использованием гейта $P(\phi)$, но ввиду того, что данный гейт редко входит в базовый набор гейтов реальных квантовых систем, будем использовать гейт $Rz(\psi)$, что повлечет за собой накопление глобальной фазы, но данная проблема будет рассмотрена отдельно.

В работе [5] решение предполагает наложение углов по аналогии с конструированием вектора модулей, последовательно применяя после гейтов $Ry(\phi)$ гейты $Rz(\psi)$ с множественным контролем. Мы предлагаем модифицировать последнюю итерацию алгоритма, совместно применяя гейты $Ry(\phi)$ и $Rz(\psi)$.

Углы фазы ψ не требуется вычислять отдельно, так как они нам известны из представления вектора состояния в показательном виде ($r_k e^{i\psi_k}$). Следовательно, на последней итерации $(c, 0) \rightarrow (a e^{i\psi_1}, b e^{i\psi_2})$, то есть мы можем не задействовать дополнительные гейты CNOT для поворота на фазу, так как второе значение в паре $(c, 0)$ равно 0. Пример одного шага последней итерации приведен на рис. 2.

Тогда псевдокод алгоритма без поправки на глобальную фазу примет вид, представленный на рис. 3, где $X_bit_mask(p, q_list)$ — применение гейтов X к кубитам с индексами в списке q_list , соответствующими битовому представлению числа p . Влияние глобальной фазы будет рассмотрено далее.

Данный алгоритм является теоретически точным, то есть его реальная точность ограничена лишь точностью выполнения операций на классическом компьютере при подсчете углов и на квантовом компьютере при исполнении схемы.

Экспериментально оценим погрешность, возникающую в ходе классических вычислений. Под погрешностью будем иметь в виду норму разницы желаемого нормированного вектора $|\psi\rangle \in C^{2^n}$ и первого столбца унитарного оператора U полученной квантовой схемы. Испытания проводились на 10 произвольных векторах $|\psi\rangle$ для систем из $n = \overline{1, 6}$ кубитов, и в табл. 1 заносилось среднее значение погрешности.

Можно заметить аномально высокую погрешность для 1 кубита при подготовке произвольного комплексного вектора состояния, она объясняется тем, что для данного случая мы не всегда можем погасить глобальную фазу (если не использовать дополнительные кубиты). Для прочих n заметно, что с его увеличением погрешность увеличивается в среднем на порядок, хотя и остается достаточно низкой. Наиболее вероятно, что рост погрешности связан с большим объемом вычислений с использованием вещественных значений.

```

input:  $2^n$  dimensional vector phase_ang
input: vector of dimension  $n$  consisting of vectors  $2^i$  ( $i = 0, \dots, n-1$ ) size level_angles
input:  $n$  dimensional vector of qubit numbers for building a circuit q_list
output: quantum state preparation circuit

1 circuit = quantum circuit with  $2^n$  qubits
2 angle = level_angles[0]
3 circuit.RY(angle[0], q_list[0])
4 if(n == 1):
5     circuit.RZ(-phase_ang[0] + phase_ang[1], q_list[0])
6 for i in range(1, n):
7     angles = level_angles[i]
8     for j in range( $2^i$ ):
9         circuit.X_bit_mask( $2^i - j - 1$ , q_list)
10        if(i == n-1):
11            circuit.RZ((-1.)*phase_ang[j], q_list[i])
12            circuit.X(q_list[i])
13            circuit.RZ(phase_ang[j+ $2^i$ ], q_list[i])
14            circuit.X(q_list[i])
15            circuit.RY(angles[j]/2, q_list[i])
16            circuit.CnNOT(i, i, q_list)
17            circuit.RY((-1)*angles[j]/2, q_list[i])
18        if(i == n-1):
19            circuit.RZ(phase_ang[j], q_list[i])
20            circuit.CnNOT(i, i, q_list)
21        if(i == n-1):
22            circuit.X(q_list[i])
23            circuit.RZ((-1)*phase_ang[j+ $2^i$ ], q_list[i])
24            circuit.X(q_list[i])
25            circuit.X_bit_mask( $2^i - j - 1$ , q_list)

```

Рис. 3. Псевдокод алгоритма подготовки квантового состояния с помощью управляемых гейтов поворота

2. Подготовка квантового состояния с помощью стратегии «разделяй и властвуй». Предложенный в статье [5] алгоритм представляет собой оптимизацию предыдущего алгоритма, которая за счет использования $2^n - n - 1$ дополнительных кубитов приводит асимптотику глубины схемы к $O(n^2)$, сохраняя ее размер $O(2^n)$.

Идея заключается в развертке алгоритма в ширину вместо длины за счет древовидной структуры алгоритма и гейтов *CSWAP*. Такая организация позволяет выполнять квантовые гейты параллельно. Алгоритм построения схемы изображен на рис. 4, где $N = 2^n$.

Углы для гейтов *Ry* вычисляются так же, как в предыдущем случае, но углы для *Rz* требуют дополнительных вычислений со схожей идеей [5]. Данный алгоритм сохраняет свойство теоретической точности предыдущего алгоритма.

Таблица 1

Погрешность, возникающая в ходе классических вычислений от количества кубитов, на которых подготавливается состояние

Количество кубитов	1	2	3	4	5	6
Погрешность для комплексных $ \psi\rangle$	0.747	1.055e-15	2.092e-14	2.237e-13	1.721e-12	1.145e-11
Погрешность для вещественных $ \psi\rangle$	1.586e-16	8.941e-16	2.119e-14	2.255e-13	1.734e-12	1.123e-11

```

input :  $N - 1$  dimensional vector  $\text{angle} = \text{gen\_angles}(abs(x))$ 
input :  $N - 1$  dimensional vector  $\text{angle}_z = \text{gen\_angles}_z(phase(x))$ 
output : Quantum circuit to load  $x$  in the amplitudes of a quantum system entangled with ancillary qubits

1 circuit = quantum circuit with  $N - 1$  qubits  $q[0], \dots, q[N - 2]$ 
2 for  $k \leftarrow 0$  to  $N - 2$  do
3    $\lfloor R_y(\text{angle}[k], q[k])$ 
4 for  $k \leftarrow 0$  to  $N - 2$  do
5    $\lfloor R_z(\text{angle}_z[k], q[k])$ 
6  $\text{actual} = \text{parent}(N - 2)$ 
7 while  $\text{actual} \geq 0$  do
8    $\text{left\_index} = \text{left}(\text{actual})$ 
9    $\text{right\_index} = \text{right}(\text{actual})$ 
10  while  $\text{right\_index} < N - 1$  do
11     $\text{cswap}(q[\text{actual}], q[\text{left\_index}], q[\text{right\_index}])$ 
12     $\text{left\_index} = \text{left}(\text{left\_index})$ 
13     $\text{right\_index} = \text{left}(\text{right\_index})$ 
14   $\text{actual} = \text{actual} - 1$ 

```

Рис. 4. Подготовка квантового состояния с помощью стратегии «разделяй и властвуй» [5]

На рис. 5 изображены повернутое дерево углов и схема, созданная с помощью стратегии «разделяй и властвуй» для подготовки состояния трех кубитов. Результат хранится в кубитах q_0, q_1 и q_3 , а прочие кубиты являются дополнительными.

3. Подготовка квантового состояния в виде аппроксимации унитарного оператора. Любая квантовая схема представляет собой некоторый унитарный оператор. Соответственно, получив вид данного оператора для схемы, описанной в разделе 2, можно его аппроксимировать.

Как было сказано ранее, первый столбец данного оператора всегда будет вектором состояния, которое мы хотим построить. Что естественным образом получается из постановки задачи в виде $U|0\rangle^{\otimes n} = |\psi\rangle$, где $|\psi\rangle$ — желаемое состояние. Соответственно, для решения данной задачи в общем случае достаточно брать унитарный оператор U с первым столбцом $|\psi\rangle$.

Для этого решим более общую задачу аппроксимации произвольного унитарного оператора U размера $2^n \times 2^n$ некоторой конечной квантовой схемой на основе некоторого конечного набора базовых гейтов. При наличии дополнительных кубитов это всегда возможно [4].

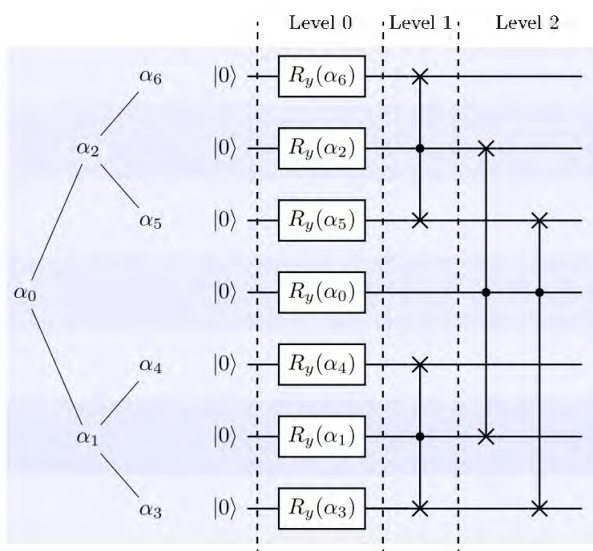


Рис. 5. Повернутое дерево углов и схема, созданная с помощью стратегии «разделяй и властвуй» для подготовки состояния трех кубитов [5]

Так как базовые наборы гейтов для разных квантовых компьютеров часто отличаются, то, представляя один универсальный набор гейтов через другой с помощью данного алгоритма, мы можем обеспечить переносимость квантовых схем на разные квантовые процессоры, что также будет использоваться в данной работе.

Идея алгоритма заключается в том, что мы заготавливаем несколько параметризованных шаблонных схем. И пытаемся оптимизировать параметры, порядок применения и количество повторений данных шаблонов, чтобы полученная в результате матрица схемы минимально отличалась от желаемой. Так как мы хотим аппроксимировать произвольный унитарный оператор, то мы изначально ничего не знаем о его виде, поэтому, для универсальности, за базовый шаблон будем брать некоторые аналоги схемы 2, изображенной на рис. 6.

Следовательно, предложенный универсальный метод может быть неоптимальным, и количество повторений шаблонных схем для достижения высокой точности может достигать 2^n , но, как правило, не более, исходя из оценок, указанных в статье [4].

Здесь и далее под погрешностью аппроксимации будем понимать $\epsilon(p) = \|U - M(p)\|$, где $M(p)$ — матрица параметризованной схемы с набором параметров p и используется норма Фробениуса. Подбор оптимальных параметров будет происходить как решение задачи оптимизации: $\min_p \epsilon(p)$.

4. Влияние ограничений современных квантовых компьютеров на построение алгоритмов. Рассмотрим влияние на построение описанных алгоритмов ограничений, связанных с составом фактически реализованных на квантовом компьютере гейтов. В качестве примера возьмем характеристики квантовых процессоров, доступных через сервис IBM Quantum. На момент подготовки статьи базовым набором гейтов для большинства доступных для общего пользования квантовых компьютеров являлся набор ECR , I , Rz , SX , X . Он отличен от набора, обычно используемого при описании квантовых алгоритмов, но также представляет собой полную систему.

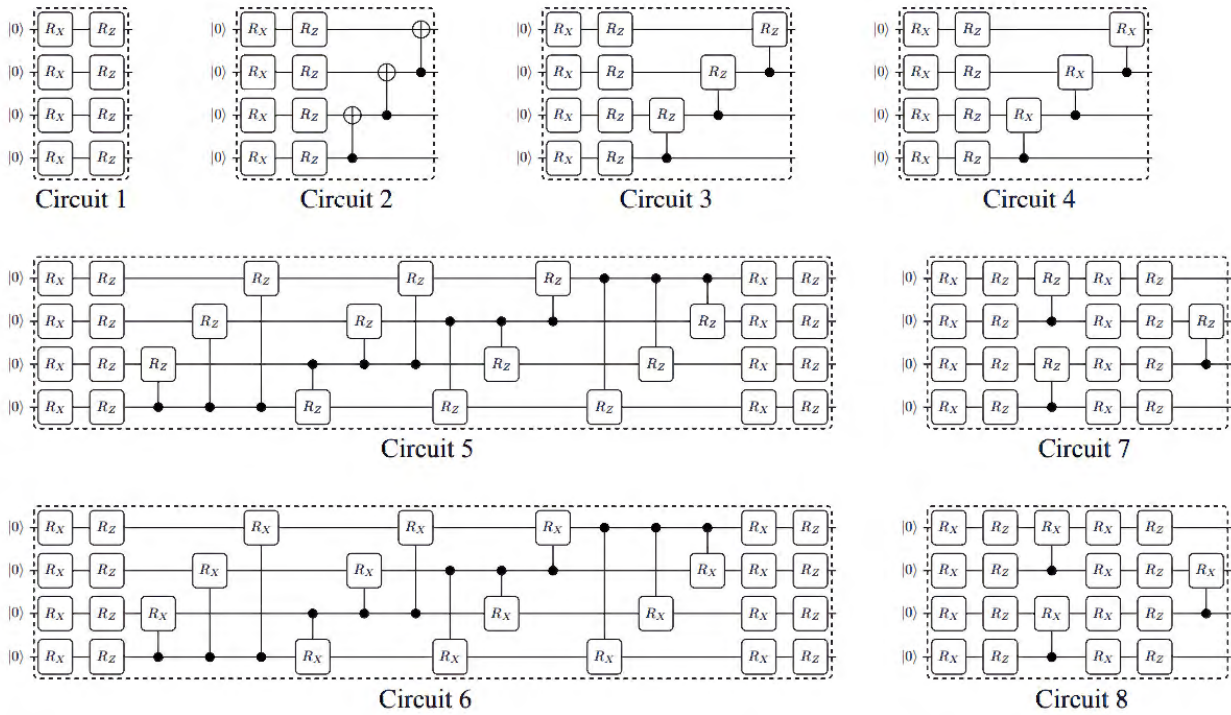


Рис. 6. Типовые примеры шаблонных схем [6]

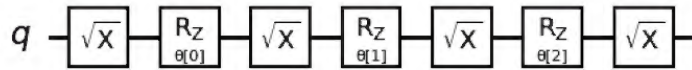


Рис. 7. Шаблонная параметризованная схема для аппроксимации однокубитных гейтов

Ранее в данной работе для построения алгоритмов подготовки начального состояния использовались гейты X , H , Ry , Rz , $CNOT$. Следовательно, необходимо аппроксимировать данный набор доступными гейтами.

Начнем с аппроксимации однокубитных гейтов. X и Rz присутствуют в обоих наборах. Оставшиеся гейты H и Ry попробуем оптимально аппроксимировать, используя шаблонную схему $SX - Rz - SX - Rz - SX - Rz - SX$, которая изображена на рис. 7.

Данная схема была выбрана за шаблонную, так как она с точностью до глобальной фазы аналогична универсальному гейту $U3$. Поэтому достаточно одного ее повторения, а глобальная фаза не станет для нас проблемой, так как данные аппроксимируемые гейты ее не накладывают.

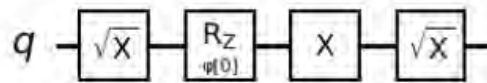
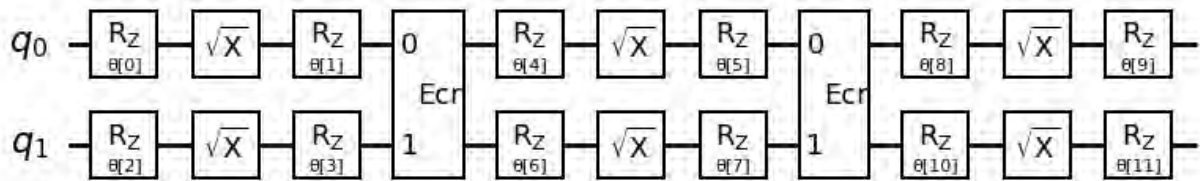
Рассмотрим процесс подробнее для гейта $Ry(\phi)$, так как он зависит от одного параметра. Попробуем провести аппроксимацию при нескольких его значениях, чтобы увидеть закономерность. Результаты представлены в табл. 2.

Заметим, что даже для одного угла результат подбора параметров может быть не однозначен. Но явно видны закономерности: второй параметр всегда 0, модуль разности

Таблица 2

Подбор параметров для аппроксимации гейта $Ry(\phi)$

Угол ϕ	Найденный набор параметров $(\theta[0], \theta[1], \theta[2])$	Погрешность аппроксимации
$\frac{\pi}{7}$	[1.4, 0.0, 0.951]	1.836e-09
$\frac{\pi}{7}$	[2.519, 0.0, 2.07]	2.522e-09
$\frac{\pi}{3}$	[2.057, 0.0, 1.01]	1.374e-09
$\frac{\pi}{3}$	[0.989, 0.0, -0.058]	2.256e-09
$\frac{\pi}{3}$	[0.559, 0.0, -0.488]	9.517e-10
$\frac{\pi}{5}$	[0.638, 0.0, 0.009]	1.305e-09

Рис. 8. Аппроксимация гейта $Ry(\phi)$ Рис. 9. Параметризованная схема для аппроксимации гейтов $CNOT$ (с точностью до глобальной фазы $\frac{\pi}{4}$) и H

первого и третьего параметра равен углу ϕ . Из данных соображений построим оптимальную аппроксимирующую схему, которая изображена на рис. 8.

Оставшийся гейт H аппроксимируется схожим образом. Теперь нам осталось аппроксимировать гейт $CNOT$. Чтобы сохранить все выводы, касающиеся глобальной фазы для алгоритма подготовки начального состояния, которые будут приведены далее, аппроксимируем не чистый $CNOT$, а $CNOT$ со сдвигом глобальной фазы на $\frac{\pi}{4}$, используя шаблонную схему, изображенную на рис. 9.

Результат аппроксимации гейтов $CNOT$ и H представлен на рис. 10. Несложно заметить, что такая замена увеличивает в разы количество используемых гейтов. Например, количество двухкубитных гейтов увеличится ровно в 2 раза.

Следующим ограничением является топология квантового компьютера. В отличие от идеальной модели, связь кубитов представляет собой неполный ориентированный граф, что накладывает ограничения на применение двухкубитных гейтов. Так как любые 2 кубита связаны, либо напрямую, либо по цепочке, то если кубиты не являются соседями, представим исходный $CNOT$ эквивалентной последовательностью гейтов между соседними кубитами [10], но такой подход увеличивает сложность применения $CNOT$ от $O(1)$

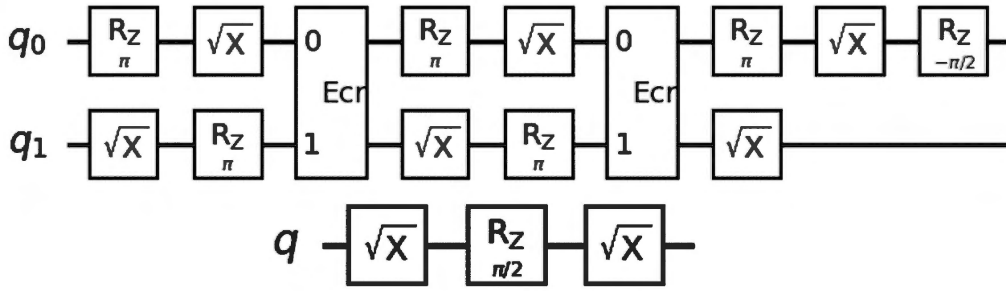


Рис. 10. Аппроксимация гейтов $CNOT$ (с точностью до глобальной фазы $\frac{\pi}{4}$) и H

к $O(p)$, где p — длина цепочки кубитов. Если связь есть только в одну сторону, то мы можем «перевернуть» $CNOT$ с помощью гейтов H , примененных к обоим кубитам до и после обратного $CNOT$.

Также на реальных системах присутствуют ошибки выполнения гейтов. Но в данной работе не рассматриваются оптимизационные алгоритмы и алгоритмы компенсации ошибок, чтобы оценить точность работы подготовки начального состояния в реализации, приближенной к теоретической.

Также может возникнуть проблема наличия глобальной фазы. Она обычно не приводит к ошибкам, так как не влияет на результат измерения, но существуют алгоритмы, которые при возникновении глобальной фазы выдают результат, отличный от желаемого. Например, алгоритмы подготовки начального состояния или алгоритмы квантовой томографии.

Применение гейта X с r управляющими кубитами ($CnNOT$), сконструированного через $CnRz(\pi)$ [8], будет прибавлять глобальную фазу $\frac{\pi}{2^{r+1}}$. Соответственно, применив два раза данный гейт с одинаковыми параметрами, мы не изменим модуль состояния, но увеличим глобальную фазу на $\frac{\pi}{2^r}$.

В нашем случае алгоритм подготовки начального состояния с помощью гейтов поворота не подвержен данной ошибке при $n > 1$, так как на i -ой итерации $r = i$, причем гейт $CnNOT$ применяется 2^{i+1} раз, следовательно, глобальная фаза на итерации равна $2^{i+1} \sum_{k=1}^i \frac{\pi}{2^{k+1}} = \pi 2^i \sum_{k=1}^i \frac{1}{2^k} = \pi(2^i - 1)$. Угол глобальной фазы кратен π , следовательно, он равен 0 (при $i = 0$) или π , тогда требуется его погасить последовательностью гейтов $X - Z - Y - X - Z - Y$, примененных к любому кубиту. Алгоритм «разделяй и властвуй» унаследует данное свойство, а алгоритм подготовки начального состояния в виде аппроксимации не подвержен глобальной фазе, если шаблонная схема подобрана с учетом базисных гейтов и ее количество повторений не меньше 2^n .

5. Результаты экспериментов. Построение и симуляция исполнения квантовых схем реализованы с помощью библиотеки qiskit [11]. Далее под идеальным симулятором будет пониматься AerSimulator, а под симулятором реального квантового компьютера FakeSherbrooke, который отражает поведение реальной системы ibm_sherbrooke [8].

Решение задачи оптимизации для аппроксимации унитарного оператора производится численно с использованием библиотеки scipy, с помощью метода, вызываемого по умолчанию функцией minimize. За базовый шаблон взят аналог схемы 2 (рис. 6) в базисных гейтах квантового компьютера FakeSherbrooke (рис. 11), а количество ее повторений будем считать фиксированным, равным 2^n . Тогда количество параметров схемы равно $3n2^n$.

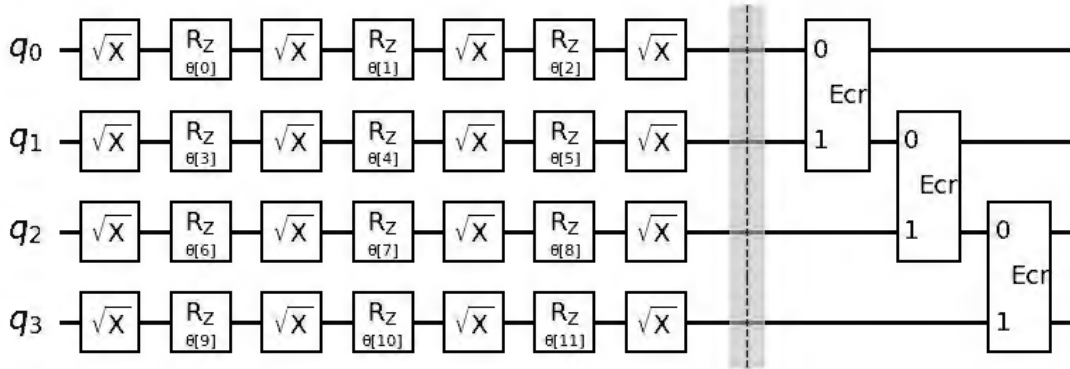


Рис. 11. Пример универсального шаблонного блока, используемого при экспериментах

Таблица 3

Сравнение размера полученных схем, количества гейтов ECR и глубины схем

Количество кубитов		1	2	3	4	5	6	7	8
Управляемые повороты	размер, <i>ECR</i> ,	17, 0,	148, 8,	2204, 200,	23476, 2184,	184152, 16648,	1240956, 110600,	7820778, 692744,	47883542, 4230664,
	глубина	17	93	1247	12087	92893	621351	3900877	23838361
Разделяй и властвуй	размер, <i>ECR</i> ,	5, 0,	212, 20,	1141, 110,	6505, 793,	13086, 1579,	30273, 3699,	82370, 10509,	
	глубина	5	117	426	1433	1827	3074	5609	
Аппроксимация	размер, <i>ECR</i> ,	14, 0,	95, 3,	266, 14,	853, 45,	2205, 124,	5397, 315,	12749, 762,	29381, 1785,
	глубина	14	49	112	337	751	1645	3563	7657

Прежде рассмотрим размеры и глубину квантовых схем в зависимости от количества кубитов. Результаты указаны в табл. 3, где также указано количество двухкубитных гейтов *ECR*. Параметр ширины схемы не рассматривается, так как он вычисляется явно и равен n для алгоритма с управляемыми гейтами поворота и аппроксимирующего алгоритма, и 2^{n-1} для алгоритма, реализующего стратегию «разделяй и властвуй».

Алгоритм подготовки квантового состояния с помощью управляемых гейтов при действиях для обхода ограничений приобрел сложность $O(n^n)$ за счет того, что он плохо ложится на линейную топологию.

Алгоритм «разделяй и властвуй» оказался более устойчивым к топологии, так как большее количество двухкубитных гейтов было применено между соседними кубитами. Но в отличие от теоретических оценок глубины схемы $O(n^2)$, он перешел к $O(\frac{2^n}{n})$, а размер схемы к $O(n2^n)$, и имеет схожую асимптотику с алгоритмом в виде аппроксимации унитарного оператора, но с большими константами. Также уже при $n = 8$ алгоритму не хватило дополнительных кубитов на используемой модели системы *ibm_sherbrooke*, которая имеет 127 кубитов.

За счет того, что шаблонная схема была заготовлена сразу с учетом ограничений квантовых компьютеров IBM, то алгоритм подготовки состояния в виде аппроксимации уни-

Таблица 4

Погрешность работы алгоритмов (симулятор идеального квантового компьютера, симулятор реального квантового компьютера FakeSherbrooke)

Количество кубитов	1	2	3
Управляемые повороты	0.00255, 0.00735	0.00617, 0.08812	0.00841, 0.44880
Разделяй и властвуй	0.00382, 0.00898	0.00489, 0.11497	0.00852, 0.29804
Аппроксимация	0.00294, 0.00956	0.00727, 0.02402	0.01213, 0.25297

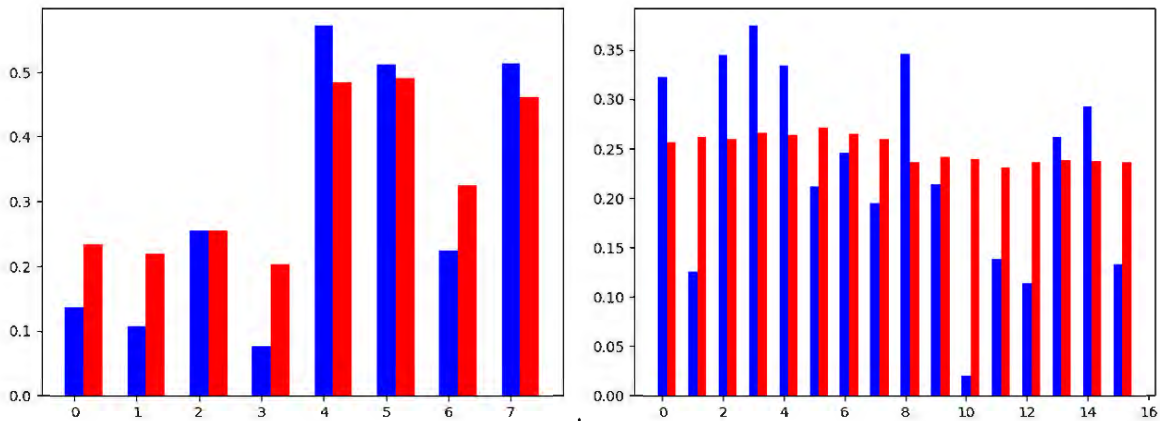


Рис. 12. Заданный вектор состояния (синий) и полученный на симуляторе реального квантового компьютера алгоритмом «разделяй и властвуй» (красный) для систем из 3 и 4 кубитов

тарного оператора достигает меньшего количества используемых гейтов, сохраняя оценку размера схемы $O(n2^n)$ и ее глубины $O(2^n)$. Но главная сложность данного подхода — это решение классической задачи оптимизации для подбора $3n2^n$ параметров.

Сравним точность данных алгоритмов с использованием идеального симулятора и симулятора реального квантового компьютера в зависимости от количества кубитов. Зададим количество измерений квантовых схем в 20000, так как данное значение является максимальным при запуске на реальных квантовых компьютерах ИВМ. Исходя из размеров квантовых схем и количества измерений, эксперименты были проведены для подготовки модулей состояний систем до трех кубитов (табл. 4). Для каждого размера схемы проведены испытания на 10 произвольных векторах. Результат представляется как средняя норма разницы целевых и полученных векторов.

Точность исполнения схемы на симуляторе идеального квантового компьютера обуславливается только погрешностью выполнения классических вычислений и количеством измерений. При исполнении на симуляторе реального квантового компьютера добавляется погрешность исполнения квантовых гейтов, и так как она относительно велика, то вносит существенный вклад из-за размеров схем. Соответственно, алгоритм подготовки состояния в виде аппроксимации унитарного оператора показывает несколько лучшие результаты, но если посмотреть на абсолютные значения результатов, то ошибка является существенной. Например, на рис. 12 изображены результаты, полученные алгоритмом «разделяй и властвуй» при подготовке состояния трех кубитов (ошибка 0.24173) и четырех кубитов (ошибка 0.38991).

Если при ошибке около 0.25 результат еще различим, то из-за особенностей квантового компьютера при исполнении больших схем результат может стремиться к равновероятному состоянию, и ошибка будет расти.

Заключение. В данной работе были рассмотрены три возможных алгоритма подготовки начального состояния. Их алгоритмическая реализация во всех случаях показала смещение оценки асимптотики размера схемы и ее глубины в худшую сторону. Лучший результат показал алгоритм подготовки состояния в виде аппроксимации унитарного оператора с $O(n2^n)$ и $O(2^n)$ соответственно, но он требует ресурсоемкой предварительной классической подготовки $3n2^n$ параметров схемы.

Все подготовленные алгоритмами квантовые схемы, адаптированные под характеристики квантового компьютера `ibm_sherbrooke`, показали существенную погрешность при подготовке начального состояния — от 0,74–0,96 % для однокубитной системы до 25–45 % для трехкубитной. Соответственно, на настоящий момент из-за недостаточной точности практической реализации гейтов в таком виде они пригодны лишь для подготовки состояний систем из не более чем трех кубитов. Возможно, применяя различные оптимизационные методы и алгоритмы компенсации ошибок, удастся увеличить это число, но без новых подходов, которые снизят асимптотику размера схемы или ее глубины на практике, не стоит ожидать значительного улучшения. При этом стоит помнить, что все описанные алгоритмы позволяют подготавливать произвольное состояние, то есть являются универсальными. Можно достичь лучших оценок для подготовки некоторых специальных квантовых состояний, например при подготовке квантового состояния в виде нормального распределения или разреженного квантового состояния [11].

Список литературы

1. Hur T., Kim L., Park D. K. Quantum convolutional neural network for classical data classification // Quantum Machine Intelligence. 2022, 4(1), 3.
2. Barenco A. et al. Stabilization of quantum computations by symmetrization // SIAM Journal on Computing. 1997, 26(5), 1541–1557.
3. Harrow A. W., Hassidim A., Lloyd S. Quantum algorithm for linear systems of equations // Physical review letters. 2009, 103(15), 150502.
4. Sun X. et al. Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2023, 42(10), 3301–3314.
5. Araujo I. F. et al. A divide-and-conquer algorithm for quantum state preparation // Scientific reports. 2021, 11(1), 6329.
6. Sim S., Johnson P. D., Aspuru-Guzik A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms // Advanced Quantum Technologies. 2019, 2(12), 1900070.
7. Cross A. W. et al. Open quantum assembly language // arXiv preprint arXiv:1707.03429. 2017.
8. IBM Quantum. [Electron. Res.]: <https://quantum.ibm.com>.
9. Barenco A. et al. Elementary gates for quantum computation // Physical review A. 1995, 52(5), 3457.
10. Shende V. V., Bullock S. S., Markov I. L. Synthesis of quantum logic circuits // Proceedings of the 2005 Asia and South Pacific Design Automation Conference. 2005, 272–275.
11. Qiskit. [Electron. Res.]: <https://github.com/Qiskit>.
12. Luo J., Li L. Circuit complexity of sparse quantum state preparation // arXiv preprint arXiv:2406.16142. 2024.



Ивлев Александр Дмитриевич — лаборант в отделе твердотельной электроники и оптоэлектроники лаборатории теории наноструктур НИФТИ ННГУ им. Лобачевского. Магистр по направлению

«Прикладная математика и информатика» в ИИТММ ННГУ им. Лобачевского. Область научных интересов: квантовые вычисления, нейронные сети. E-mail: ivlev-sd@mail.ru. Тел.: +7 (930) 740-36-55.

Ivlev Alexander Dmitrievich — laboratory assistant in the Department of Solid-state Electronics and Optoelectronics at the Laboratory of Nanostructure Theory NIFTI Lobachevsky State University of Nizhny Novgorod. Master's degree in «Applied Mathematics and Computer Science» at the IITMM Lobachevsky State University of Nizhny Novgorod. Research interests: quantum computing, neural networks. E-mail: ivlev-sd@mail.ru. Tel.: +7 (930) 740-36-55.



Линеv Алексей Владимирович — Руководитель Приволжского научно-образовательного центра суперкомпьютерных технологий института Информационных технологий, математики и механики

Нижегородского государственного университета им. Н.И. Лобачевского. Область научных интересов: высокопроизводительные вычисления, математическое моделирование, квантовые вычисления. E-mail: alin@unn.ru. Тел.: +7 (831) 462-33-56.

Liniov Alexey Vladimirovich — Head of Volga Research and Education Center for Supercomputing Technologies, Institute of Information Technology, Mathematics and Mechanics of Lobachevsky State University of Nizhny Novgorod. Area of scientific interests: high-performance computing, mathematical modeling, quantum computing. E-mail: alin@unn.ru. Tel.: +7 (831) 462-33-56.

Дата поступления — 01.02.2025