Bridging Gates over Qubits

Seyed Sajad Kahani
Supervisor: Prof. Dan Browne

August 12, 2023

Contents

1 **Introduction** **1**

2 **Background** **3**
2.1 Quantum Compilation . . . 3
2.1.1 Gate synthesis . . . 4
2.1.2 Qubit Allocation and Routing . . . 5
2.1.3 Optimizations . . . 6
2.2 Two-qubit gates . . . 6
2.2.1 Entanglement Power . . . 6

3 **Related Works** **8**

4 **Problem Statement** **10**

5 **Discussion** **11**

6 **Implementation** **13**

7 **Conclusion** **14**

Abstract

Chapter 1

Introduction

Quantum computation (QC) is an emerging field that aims to use quantum mechanics to solve problems that are intractable for classical computers. Since the earliest conceptualization of quantum computation [15], it has been believed that quantum computers could revolutionize the way we solve problems, particularly those involving simulating nature. Over time, it has become clear that quantum computers have applications far beyond physical simulations. There are algorithms for search and traversing graphs, solving linear equations [25], and methods for machine learning and optimization [19].

Despite significant efforts, we are still far from fully utilizing these algorithms. Our current hardware technology has not yet achieved the desired accuracy and number of qubits necessary for quantum computers to outperform classical computers in solving useful problems. The current situation is commonly referred to as the "noisy intermediate scale quantum" (NISQ) era [32], characterized by restricted resources, including a limited number of qubits, constrained qubit connectivity, hardware-specific gate sets, and limited circuit depth due to noise [13].

The restricted qubit resources and excessive noise susceptibility of NISQ devices necessitate optimal compilers to have any hope of useful near-term quantum computation. A huge amount of research has been conducted to tackle different aspects of the compilation problem, including qubit allocation [18, 38, 29, 48, 24], routing [7, 49, 18, 11, 27, 21] and gate synthesis [35, 43, 44, 36, 4, 14]. These aspects are deeply intertwined and one may not distinguish between them, but all of them are in some sense a circuit transformation from a higher-level circuit (with fewer imposed constraints) to a lower-level circuit (with more imposed constraints) [17].

While the knowledge of classical compilation is adopted and the divergent points (no-cloning [**TODO**] and reversibility [37]) are studied and addressed well, another important distinction has received less attention - the cost of SWAP operations. In classical circuit synthesis, SWAPs simply rearrange wires at negligible cost, compared to two-bit gates. But in the quantum realm, SWAP gates require double entangling [**TODO**] interactions between qubits, making them the most expensive two-qubit quantum logic gates [**TODO**].

Despite extensive research into minimizing the overall number of SWAP operations [**TODO**, 7], there is little work addressing the inherent cost of each SWAP gate. A few recent works have proposed techniques to reduce the cost of SWAP gates, such as embedding SWAPs within other 2-qubit gates in 2QAN compiler [23], or optimization of SWAP decompositions into CNOT gates [21, 27]. In this work, we aim to address the primary usage of SWAP gates - enabling connectivity between non-adjacent qubits. We analyze the possibility of simplifications for different connectivity cases.

Here we define a problem called bridging that is to find a circuit that applies a two-qubit gate on two non-adjacent qubits. By utilizing the framework of [21] and the extensive literature of network coding [**TODO**] show that in the classical case, the cost of bridging over n bits is $4n$ to $6n$ (upto a $O(1)$ constant). We also present a circuit that achieves the lower bounds. We then attempt to extend the results to the quantum regime, by presenting a circuit to bridge two-qubit gates with Schmidt number 2 over n with optimal number of CNOTs.

This advancement will lead to 33% reduction in the cost of the most expensive two-qubit gate in many situations. To demonstrate the practicality of our results, we implement the algorithms and benchmark with application-oriented dataset of circuits.

The rest of this thesis is organized as follows. Chapter 2 reviews the related works. In Chapter 5, we present the algorithms and prove their correctness and optimality for the classical and quantum cases. We implement the algorithms and benchmark against state-of-the-art techniques in Chapter 6. Finally, Chapter 7 concludes and discusses avenues for future work.

Chapter 2

Background

We should be concerned with two key findings from the literature review on quantum compilation. First, the way that they broke down the problem and the assumption they made to simplify the problem space and introducing some structure to that. On the other hand, is the alogrithms that they involve techniques from QC, graph theory and more.

Here we try to review both of these aspects to draw a big picture of our notion of quantum compilation and also to review the existing techniques related to our approach, like the special techniques for two-qubit gates.

2.1 Quantum Compilation

The term "quantum compilation" can refer to any process that transforms a higher-level description of a quantum algorithm into a lower-level description [17]. In the majority of works [50, 7, 12, 39, 33, 30], circuits are the description used and thereby the compilation is done by transforming a general quantum circuit into a circuit that is compatible with a specific hardware. Given that the problem of finding the most optimal circuit (with respect to a sense of complexity like depth or number of gates) is proven to be NP-hard [38], the research primarily centers on deconstructing this problem into smaller components and devising techniques to effectively balance between the agility of the process and the quality of the solution. This mirrors the approach adopted in classical compilation [2].

However, the clear structure for this breakdown has not been firmly established, with numerous diverse approaches in play. As such, our goal is to provide a comprehensive overview of the issue and identify common patterns in the existing literature. This overall picture will then serve as a reference point throughout the rest of this thesis. To achieve this overview, it's crucial to define **circuit transformation** as a process that preserves the essential meaning of the circuit, ensuring that the circuit's output remains consistent for any input before and after the transformation. Yet, this process alters the circuit to adhere to specific constraints or optimize it for particular goals.

Thus, **quantum compilation** in our essay will be a sequence of circuit transformations where each transformation either enforces a constraint or optimizes the circuit (where optimization itself can be perceived as a soft constraint). The primary constraints imposed upon the circuit arise from the characteristics of the hardware and are listed below:

- **Gate set**: The set of gates physically available in the hardware.
- **Connectivity**: The connectivity between qubits in the hardware topology.

Furthermore, the optimizations that corresponds to different degrees various degrees of freedom (e.g., qubit assignments, choosing among equivalent subcircuits) can be applied. They are commonly pursuing these goals:

- **Complexity reduction**: Reducing the number of gates or depth of the circuit.
- **Preparation for constraints**: Minimizing violation of the mentioned constraints before imposing them.
- **Error mitigation**: The error of the circuit with respect to the hardware, especially in NISQ devices.

Quantum compilation frameworks mainly use a similar approach, they introduce transformations that are each is somehow seeking one of the goals defined before, tket even uses the term predicates for the constraints that are similar to these [39]. Typical transformations in their compilation process are, decomposing gates (imposing gate set constraint), assigning qubits (optimization of connectivity as a soft constraint), routing (imposing connectivity constraint), and further optimizations (reducing complexity). Yet the order of these transformations is not the same everywhere, for example, some [23, 33] deffer the gate synthesis after the routing while some [46, 39] does it otherwise.

With this background established, we now dive into details on gate set and connectivity constraints, and circuit optimizations techniques for reducing complexity.

2.1.1 Gate synthesis

Imposing the gate set constraints which is known as the gate synthesis, is one of the oldest subroutines in the quantum compilation. The problem here is to decompose a general n-qubit gate into a sequence of gates from the physically available gates on the device. Here we review the results for the synthesis of one, two, and more than two-qubit gates.

While most of the devices have the capability to perform arbitrary one-qubit gates, with a neglectible cost, there is a famous result for the synthesis of one-qubit gates, called Solovay-Kitaev theorem [14]. It states that any one-qubit gate can be approximated by a sequence of gates from a universal gate set with an error of ϵ using $O(\log^c(1/\epsilon))$ gates, where c is a constant. Here as like as other results, we assume that all of one-qubit gates are available.

For two-qubit gates, it is known that the set of all one-qubit gates together with CNOT can produce any two-qubit gate and it can be done by upto three CNOTs (and it is optimal) [43, 45]. One of the famous gates that needs three CNOTs is SWAP. Using CNOT as the only available two-qubit gate although is common in theoretical studies of compilation [50, 38, 24, 48, 49, 18, 26] there are other continuous gate sets such as fSim(θ, ϕ) [16] or other Hamiltonians evolution [9]. Also, the evolution of XYZ Heisenberg interaction Hamiltonian, has been used as an intermediate gate set or a tool for analytical analysis in some works [40, 45]. The importance of this Hamiltonian is because of a decomposition (Cartan's KAK decomposition, Cirac-Kraus decomposition, or Khaneja-Glaser decomposition) that states any two-qubit gate can be made by only one evolution of XYZ Hamiltonian and four one-qubit gates [22, 20].

For more than two-qubit gates, first TOFOLLI gate was used to prove the universality [4], later it was proven that it is not necessary. Although there are a few ways to synthesis a general n-qubit gate [40, 35], they are not commonly used because of its inherent inefficiency. It is known that it could not be better than $O(4^n)$ gates. [35]

Synthesis of local Hamiltonian evolution, as a special case of n-qubit gate, is also studied as it is important for simluation purposes [10]. Most of the researchs are built upon Suzuki-Trotter decomposition [42, 41], that approximates the time evolution of a Hamiltonian with a sequence of time evolutions of its terms. While most of the works rely on the first-order Suzuki-Trotter decomposition (ak Lie-Trotter formula) [39, 33], there are a few efforts to analysis higher-order errors [8] or to use other decomposition, such as a random sampling decomposition (QDRIFT) [6]. Beside the gate synthesis, Hamiltonian compilation has implications that can be used in other parts of the compilation process, for example [23] defines routing and scheduling algorithms specifically for Hamiltonian compilation.

2.1.2 Qubit Allocation and Routing

Connectivity constraint as the hardest constraint to impose, is the main focus of many researches. Research often break it down into two subproblems, qubit allocation and routing. The definition of them may vary in the literature, but we can roughly define the **qubit allocation** qubit allocation we roughly mean a mapping from logical qubits of a quantum circuit to physical qubits of a device in way that minimizes the circuit's complexity overhead due to routing. And by **routing** we mean the problem of finding an optimal circuit that is compatible with device connectivity and is equivalent to the circuit.

Qubit allocation shares common traits with other resource allocation problems [1],[2, pp. 440-444] and it is not wondering that it is NP-hard for arbitrary connectivity graphs, this can be shown easily by a reduction from graph isomorphism [38]. Note that real-world devices are not arbitrary graphs and it might be exactly solvable in reasonable time for certain families of graphs.

There have been attempts to find the optimal solution by searching for arbitrary connectivity graphs, which can be feasible for small devices [38]. But, the most common approach in the research is to use a heuristic [48, 18, 11, 29, 26] together with a search algorithm (such as BFS, A^*[50], simulated annealing[49], or others[24]) to find a reasonable solution. Some of these efforts have also been implmented in current quantum compilers [33, 39].

Results suggest that the qubit allocation could affect the complexity of the circuit by 10% in realistic scenarios [29]. This fact justifies such a break-down of connectivity constraint into two subproblems furthermore.

2.1.3 Optimizations

Circuit optimization is often achieved by applying simplification rules to the circuit [31]. These simplification rules are usually based on the commutation relations of the gates [18].

Yet, this simplification rules are implemented as a pattern matching, therefore the hidden patterns that can be revealed by another simplification rules will not be found.

2.2 Two-qubit gates

Another important fact about the gate synthesis of two-qubit gate, which is fruitful for many theoretical purposes is KAK (aka Khaneja-Glaser [20] or Kraus-Cirac [22]) decomposition.??

Kraus-Cirac Decomposition

The Kraus-Cirac decomposition [22] helps us create arbitrary two-qubit gates. It states that any two-qubit gate can be created by a Heisenberg model and local gates.

Theorem 1 (Kraus-Cirac Decomposition)**.** *For any $U \in SU(4)$, there exist $V_1, V_2, V_3, V_4 \in SU(2)$ together with $\alpha, \beta, \gamma \in \mathbb{R}$ such that*

$$U = V_1 \otimes V_2 e^{\alpha X \otimes X + \beta Y \otimes Y + \gamma Z \otimes Z} V_3 \otimes V_4. \tag{2.1}$$

This theorem, together with the basic properties of entanglement power, will also lead to many more results on communication using two-qubit gates [5] and also the universality and optimality of three CNOTs for two-qubit gates [43].

2.2.1 Entanglement Power

We already know that there are so many measures for the entanglement of a bipartite state, building upon those measures, we can define entanglement power.

Entanglement power is a measure of the ability of a two-qubit gate to create entanglement. It has multiple definitions, such as the maximum amount of ebits that can be created from product states [34], or the average amount of them (with respect to a Haar distribution) [47], or even the number of terms in a Schmidt decomposition (which will be equal to the number of non-zero terms in the Kraus-Cirac decomposition) [28].

This measure, assigns a number to each gate, and then, by composing gates, the entanglement power could not exceed the summation of the entanglement powers of the gates that are used to create the gate. This fact is used to prove many tight bounds for decomposition of two-qubit gates.

Moreover, these efforts implicitly define a hierarchy of two-qubit gates based on the number of non-zero terms in the interaction. For example, CNOT has one non-zero term while SWAP has three.

To understand the current state of the art in quantum compilation, we will first look at the general compilation problem.

Chapter 3

Related Works

Routing and Bridging

In most cases, the treatment of the initial mapping and subsequent mappings is different [49, 24]. This is because while subsequent mappings can be seen as a routing problem in the permutation space, the initial mapping is a search to find the best starting point.

Another approach is to use partial permutations, which allows for the use of the same algorithm for both initial mapping and subsequent mappings [50].

Definition 1 (Partial Mapping)**.** *A partial mapping, is a partial injective function from the logical qubits to the physical qubits. It means that some of the logical qubits may not be mapped to any physical qubit.*

Routing, by child: However, real-world devices are not arbitrary, and by imposing some restrictions on the connectivity graph, we can solve the problem in polynomial time [7]. For example, the problem is solvable in polynomial time for path graphs, complete graphs, tree graphs, and product graphs. These results has already been known for similar classical problems like token swapping and routing via matching [3].

Problem 1 (Routing via Matching)**.** *Given a graph and a set of pebble, each of them at each node, a permutation is given and we need to achive the permutation by moving the pebbles. each move consists of swapping two pebbles at adjacent nodes. The problem is to find the minimum number of moves (a.k.a. routing number).*

The solution to the qubit allocation problem will not necessarily specify the SWAPs that are needed to change the mapping, although some approaches do so [7, 24, 49]. In other cases, we need to use a routing or search algorithm to find the SWAPs that are required to change the mapping [50, 39].

Moreover, bridge gates can be used as an alternative in cases where we need to SWAP back and forth between two qubits. While most papers use only bridge

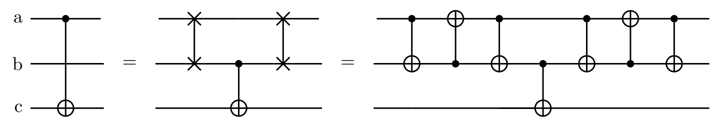

Figure 3.1: Applying a CNOT gate on (a, c) using a SWAP gate

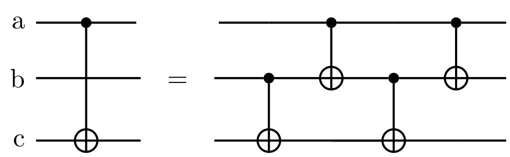

Figure 3.2: Applying a CNOT gate on (a, c) using a bridge gate

gates for three qubits [39, 18, 35, 38] (one qubit in between), the general case of bridge gates is also studied [49, 27].

Figure 3 and 3 show the difference between using a SWAP gate and a bridge gate to perform a CNOT gate for the case of three qubits. The bridge gate is more efficient in terms of the number of CNOT gates, but it is less efficient in terms of depth.

Now, the generalized bridge gate is defined as follows [27]:

Definition 2. *Generalized Bridge Gate A CNOT between qubit* 1 *and* n *can be performed using a generalized bridge gate as follows:*

$$\mathrm{Bridge}(1, n) = \prod_{i=1}^{n-1} (\mathrm{CNOT}(i+1, i) \prod_{i=n-2}^{2} \mathrm{CNOT}(i+1, i))^2 \tag{3.1}$$

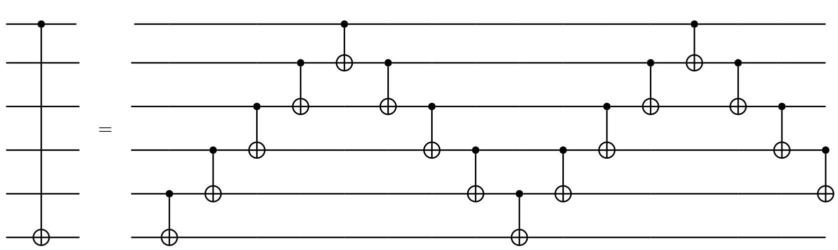

Figure 3.3: The bridge gate for $n = 6$

Chapter 4

Problem Statement

In addition to CNOT gates, it is also inevitable to use SWAP gates while allocating and routing the logical qubits through the physical qubits. However, there is another technique that could be used in some scenarios, called bridge gates [39, 18, 35, 38] or remote CNOTs [49, 27], which will be studied in detail in the next section.

While reviewing

Definition 3 (Schmidt Rank)**.** *Schmidt rank of a two-qubit gate U is defined as d, in the following Schmidt decomposition:*

$$U = \sum_{i=1}^{d} \lambda_i A_i \otimes B_i \tag{4.1}$$

For example, CNOT gate has Schmidt rank $d = 2$, while SWAP gate has Schmidt rank $d = 4$.

Definition 4 (Bridging Gate)**.** *Given a connectivity constraint as a graph $G = (V, E)$, and a target gate T that is a two-qubit gate defined on two qubits $a, b \in V$, then a sequence of gates defined on qubits in V that implements T is called a bridging gate.*

Chapter 5

Discussion

Here we introduce a sequence of gates that can be used to bridge an specific family of two qubit gates (those with Schmidt rank $d = 2$) over n qubits, under a linear connectivity constraints. This circuit uses $4n + O(1)$ CNOTs with the depth $n + O(1)$. Later, we show that this circuit is optimal. After that we develop this idea further for any connectivity connectivity and we study its implication on bridging a CNOT in a fully-classical reversible circuit scheme.

While we know that, the easiest way to bridge a target gate T over n qubits with linear connectivity is to use consequtive SWAPs, which will need $2n+O(1)$ SWAPs ($6n+O(1)$ CNOTs) and has at least $3n+O(1)$ depth (as we are assuming CNOTs as the only two-qubit gate in our gate set). This solution has been shown in Figure 5.1.

First assume qubits are arranged in a line, naming them $1, 2, \ldots, n$. We aim to apply two-qubit gate T on qubits $1, n$ while the only allowed to apply CNOT gates on adjacent qubits. We use the following circuit to achieve this goal:

This circuit has depth $n+O(1)$ and uses $4n+O(1)$ CNOT gates. It can also be

ht

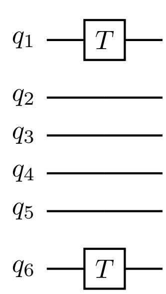

Figure 5.1: Naive bridging of a two-qubit gate T over $n = 6$ qubits with linear connectivity.

Assume a set of qubits - The main circuit (linear connectivity) - The main theorem (linear connectivity) - no-go for Schmidt number 1 - Extenstion to arbitrary connectivity - Corollary for classical circuits

While the question of optimal bridging with linear connectivity for Schmidt number $d > 1$ is left unanswered, we conjecture that it is impossible to bridge them any better than the naive solution with SWAPs. ...

- no-go for Schmidt number ¿ 1 - Ancilla qubits and network coding

Chapter 6

Implementation

Chapter 7

Conclusion

TODO

Bibliography

[1] Mansoor Alicherry and T.V. Lakshman. “Network aware resource allocation in distributed clouds”. In: *2012 Proceedings IEEE INFOCOM*. IEEE, Mar. 2012. DOI: 10.1109/infcom.2012.6195847.

[2] Randy Allen and Ken Kennedy. *Optimizing Compilers for Modern Architectures: A Dependence-based Approach*. English. 1st edition. San Francisco: Morgan Kaufmann, Oct. 2001. ISBN: 9781558602861.

[3] Indranil Banerjee and Dana Richards. “New Results on Routing via Matchings on Graphs”. In: *Fundamentals of Computation Theory*. Springer Berlin Heidelberg, 2017, pp. 69–81. DOI: 10.1007/978-3-662-55751-8_7.

[4] A. Barenco et al. “Elementary gates for quantum computation”. In: *Physical Review A* 52.5 (Nov. 1, 1995). ZSCC: 0005107, pp. 3457–3467. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.52.3457. arXiv: quant-ph/9503016. URL: http://arxiv.org/abs/quant-ph/9503016 (visited on 08/10/2023).

[5] Dominic W. Berry and Barry C. Sanders. “Relation between classical communication capacity and entanglement capability for two-qubit unitary operations”. In: *Phys. Rev. A. 68, 032312 (2003)* 68.3 (July 11, 2002), p. 032312. DOI: 10.1103/physreva.68.032312. arXiv: quant-ph/0207065 [quant-ph].

[6] Earl Campbell. “Random Compiler for Fast Hamiltonian Simulation”. In: *Physical Review Letters* 123.7 (Aug. 2019), p. 070503. DOI: 10.1103/PhysRevLett.123.070503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.070503 (visited on 01/22/2023).

[7] Andrew M Childs, Eddie Schoute, and Cem M Unsal. “Circuit Transformations for Quantum Architectures”. In: (), p. 29.

[8] Andrew M. Childs et al. “A Theory of Trotter Error”. In: *Physical Review X* 11.1 (Feb. 1, 2021). ZSCC: 0000002, p. 011020. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.11.011020. arXiv: 1912.08854[cond-mat,physics:physics,physics:quant-ph]. URL: http://arxiv.org/abs/1912.08854 (visited on 03/24/2023).

[9] Andrew M. Childs et al. "Characterization of universal two-qubit Hamiltonians". In: *Quantum Information and Computation* 11.1 (). ZSCC: 0000037. ISSN: 15337146, 15337146. DOI: 10.26421/QIC11.1-2. arXiv: 1004.1645[quant-ph]. URL: http://arxiv.org/abs/1004.1645 (visited on 08/11/2023).

[10] Andrew M. Childs et al. "Toward the first quantum simulation with quantum speedup". In: *Proceedings of the National Academy of Sciences* 115.38 (Sept. 18, 2018). ZSCC: 0000363, pp. 9456–9461. DOI: 10.1073/pnas.1801723115. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1801723115 (visited on 03/17/2023).

[11] Alexander Cowtan et al. "On the qubit routing problem". en. In: (2019). arXiv:1902.08091 [quant-ph], 32 pages. DOI: 10.4230/LIPIcs.TQC.2019.5. arXiv: 1902.08091 [quant-ph]. URL: http://arxiv.org/abs/1902.08091 (visited on 02/03/2023).

[12] Andrew Cross et al. "OpenQASM 3: A Broader and Deeper Quantum Assembly Language". In: *ACM Transactions on Quantum Computing* 3.3 (Sept. 30, 2022), pp. 1–50. ISSN: 2643-6809, 2643-6817. DOI: 10.1145/3505636. (Visited on 11/11/2022).

[13] Andrew W. Cross et al. "Validating quantum computers using randomized model circuits". In: *Phys. Rev. A* 100 (3 Sept. 2019), p. 032328. DOI: 10.1103/PhysRevA.100.032328. URL: https://link.aps.org/doi/10.1103/PhysRevA.100.032328.

[14] C.M. Dawson and M.A. Nielsen. "The Solovay-Kitaev algorithm". In: *Quantum Information and Computation* 6.1 (Jan. 2006), pp. 81–95. DOI: 10.26421/qic6.1-6.

[15] Richard P. Feynman. "Quantum mechanical computers". In: *Foundations of Physics* 16.6 (June 1, 1986). ZSCC: 0001758, pp. 507–531. ISSN: 1572-9516. DOI: 10.1007/BF01886518. URL: https://doi.org/10.1007/BF01886518 (visited on 03/23/2023).

[16] B. Foxen et al. "Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms". In: *Phys. Rev. Lett. 125, 120504 (2020)* 125.12 (Jan. 23, 2020), p. 120504. DOI: 10.1103/physrevlett.125.120504. arXiv: 2001.08343 [quant-ph].

[17] Robert Hundt. "Quantum Languages, Compilers, and Tools". In: *Quantum Computing for Programmers*. Cambridge University Press, 2022, pp. 292–321. DOI: 10.1017/9781009099974.010.

[18] Toshinari Itoko et al. "Optimization of Quantum Circuit Mapping using Gate Transformation and Commutation". In: (July 2019). DOI: 10.48550/ARXIV.1907.02686. arXiv: 1907.02686 [quant-ph].

[19] S. Jordan. *Quantum Algorithm Zoo*. Mar. 17, 2023. URL: https://quantumalgorithmzoo.org.

[20] Navin Khaneja and Steffen J. Glaser. "Cartan decomposition of SU(2n) and control of spin systems". In: *Chemical Physics* 267.1-3 (June 2001), pp. 11–23. DOI: 10.1016/s0301-0104(01)00318-4.

[21] Aleks Kissinger and Arianne Meijer-van de Griend. *CNOT circuit extraction for topologically-constrained quantum memories*. arXiv:1904.00633. ZSCC: 0000062 type: article. arXiv, May 26, 2019. arXiv: 1904.00633[quant-ph]. URL: http://arxiv.org/abs/1904.00633 (visited on 07/19/2023).

[22] B. Kraus and J. I. Cirac. "Optimal Creation of Entanglement Using a Two–Qubit Gate". In: *Physical Review A* 63.6 (May 2001). arXiv:quant-ph/0011050, p. 062309. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.63.062309. URL: http://arxiv.org/abs/quant-ph/0011050 (visited on 02/13/2023).

[23] Lingling Lao and Dan E. Browne. *2QAN: A quantum compiler for 2-local qubit Hamiltonian simulation algorithms*. Nov. 7, 2021. arXiv: 2108.02099.

[24] Gushu Li, Yufei Ding, and Yuan Xie. "Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices". In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '19: Architectural Support for Programming Languages and Operating Systems. Providence RI USA: ACM, Apr. 4, 2019, pp. 1001–1014. ISBN: 978-1-4503-6240-5. DOI: 10.1145/3297858.3304023. (Visited on 11/11/2022).

[25] Ashley Montanaro. "Quantum algorithms: an overview". In: *npj Quantum Information* 2.1 (Jan. 2016). DOI: 10.1038/npjqi.2015.23.

[26] Prakash Murali et al. "Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers". In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, Apr. 2019. DOI: 10.1145/3297858.3304075.

[27] Beatrice Nash, Vlad Gheorghiu, and Michele Mosca. "Quantum circuit optimizations for NISQ architectures". In: *Quantum Science and Technology* 5.2 (Mar. 31, 2020). ZSCC: 0000081, p. 025010. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab79b1. arXiv: 1904.01972[quant-ph]. URL: http://arxiv.org/abs/1904.01972 (visited on 03/24/2023).

[28] Michael A. Nielsen et al. "Quantum dynamics as a physical resource". In: *Physical Review A* 67.5 (May 2003), p. 052301. DOI: 10.1103/physreva.67.052301.

[29] Alexandru Paler. *On the Influence of Initial Qubit Placement During NISQ Circuit Compilation*. Jan. 30, 2019. arXiv: 1811.08985.

[30] Alexandru Paler, Alwin Zulehner, and Robert Wille. “NISQ circuit compilation is the travelling salesman problem on a torus”. In: *Quantum Science and Technology* 6.2 (Mar. 2021). ZSCC: 0000010, p. 025016. ISSN: 2058-9565. DOI: 10.1088/2058-9565/abe665. URL: https://dx.doi.org/10.1088/2058-9565/abe665 (visited on 08/10/2023).

[31] Jessica Pointing et al. “Quanto: Optimizing Quantum Circuits with Automatic Generation of Circuit Identities”. In: (Nov. 22, 2021). DOI: 10.48550/ARXIV.2111.11387. arXiv: 2111.11387 [quant-ph].

[32] John Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.

[33] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

[34] Yi Shen and Lin Chen. “Entangling power of two-qubit unitary operations”. In: *Journal of Physics A: Mathematical and Theoretical* 51.39 (Aug. 2018), p. 395303. DOI: 10.1088/1751-8121/aad7cb.

[35] Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. “Synthesis of Quantum Logic Circuits”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25.6 (June 2006). arXiv:quant-ph/0406176, pp. 1000–1010. ISSN: 0278-0070, 1937-4151. DOI: 10.1109/TCAD.2005.855930. URL: http://arxiv.org/abs/quant-ph/0406176 (visited on 02/03/2023).

[36] Vivek V. Shende, Igor L. Markov, and Stephen S. Bullock. *Minimal Universal Two-qubit Quantum Circuits*. ZSCC: NoCitationData[s0]. Mar. 12, 2004. DOI: 10.1103/PhysRevA.69.062321. arXiv: quant-ph/0308033. URL: http://arxiv.org/abs/quant-ph/0308033 (visited on 07/25/2023).

[37] Vivek V. Shende et al. *Reversible Logic Circuit Synthesis*. arXiv:quant-ph/0207001. ZSCC: NoCitationData[s0] type: article. arXiv, Feb. 27, 2003. DOI: 10.48550/arXiv.quant-ph/0207001. arXiv: quant-ph/0207001. URL: http://arxiv.org/abs/quant-ph/0207001 (visited on 07/26/2023).

[38] Marcos Yukio Siraichi et al. “Qubit allocation”. In: *Proceedings of the 2018 International Symposium on Code Generation and Optimization*. CGO ’18: 16th Annual IEEE/ACM International Symposium on Code Generation and Optimization. Vienna Austria: ACM, Feb. 24, 2018, pp. 113–125. ISBN: 978-1-4503-5617-6. DOI: 10.1145/3168822. (Visited on 11/11/2022).

[39] Seyon Sivarajah et al. “t$—$ket$\rangle$: A Retargetable Compiler for NISQ Devices”. In: *Quantum Science and Technology* 6.1 (Jan. 1, 2021). ZSCC: NoCitationData[s0], p. 014003. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab8e92. arXiv: 2003.10611[quant-ph]. URL: http://arxiv.org/abs/2003.10611 (visited on 03/24/2023).

[40] P. B. M. Sousa and R. V. Ramos. *Universal quantum circuit for n-qubit quantum gate: A programmable quantum gate*. arXiv:quant-ph/0602174. ZSCC: 0000045 type: article. arXiv, May 29, 2006. DOI: 10.48550/arXiv.quant-ph/0602174. arXiv: quant-ph/0602174. URL: http://arxiv.org/abs/quant-ph/0602174 (visited on 08/11/2023).

[41] Masuo Suzuki. "General theory of fractal path integrals with applications to many-body theories and statistical physics". In: *Journal of Mathematical Physics* 32.2 (Feb. 1991), pp. 400–407. DOI: 10.1063/1.529425.

[42] Hale F Trotter. "On the product of semi-groups of operators". In: *Proceedings of the American Mathematical Society* 10.4 (1959), pp. 545–551.

[43] Farrokh Vatan and Colin Williams. "Optimal Quantum Circuits for General Two-Qubit Gates". In: *Physical Review A* 69.3 (Mar. 22, 2004). ZSCC: 0000332, p. 032315. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.69.032315. arXiv: quant-ph/0308006. URL: http://arxiv.org/abs/quant-ph/0308006 (visited on 03/24/2023).

[44] Farrokh Vatan and Colin P. Williams. *Realization of a General Three-Qubit Quantum Gate*. arXiv:quant-ph/0401178. ZSCC: 0000035 type: article. arXiv, Jan. 30, 2004. DOI: 10.48550/arXiv.quant-ph/0401178. arXiv: quant-ph/0401178. URL: http://arxiv.org/abs/quant-ph/0401178 (visited on 07/19/2023).

[45] G. Vidal and C. M. Dawson. "A universal quantum circuit for two-qubit transformations with three CNOT gates". In: *Physical Review A* 69.1 (Jan. 2004). arXiv:quant-ph/0307177, p. 010301. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.69.010301. URL: http://arxiv.org/abs/quant-ph/0307177 (visited on 02/13/2023).

[46] Robert Wille, Stefan Hillmich, and Lukas Burgholzer. "Efficient and Correct Compilation of Quantum Circuits". In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2020 IEEE International Symposium on Circuits and Systems (ISCAS). ZSCC: 0000010 ISSN: 2158-1525. Oct. 2020, pp. 1–5. DOI: 10.1109/ISCAS45731.2020.9180791.

[47] Paolo Zanardi, Christof Zalka, and Lara Faoro. "On the Entangling Power of Quantum Evolutions". In: *Phys.Rev.A62:030301,2000* 62.3 (May 8, 2000), p. 030301. DOI: 10.1103/physreva.62.030301. arXiv: quant-ph/0005031 [quant-ph].

[48] Chi Zhang et al. "Time-optimal Qubit mapping". In: *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '21: 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. Virtual USA: ACM, Apr. 19, 2021, pp. 360–374. ISBN: 978-1-4503-8317-2. DOI: 10.1145/3445814.3446706. (Visited on 11/11/2022).

[49] Xiangzhen Zhou, Sanjiang Li, and Yuan Feng. "Quantum Circuit Transformation Based on Simulated Annealing and Heuristic Search". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.12 (Dec. 2020), pp. 4683–4694. ISSN: 0278-0070, 1937-4151. DOI: 10.1109/TCAD.2020.2969647. URL: https://ieeexplore.ieee.org/document/8970267/ (visited on 11/11/2022).

[50] Alwin Zulehner, Alexandru Paler, and Robert Wille. "Efficient mapping of quantum circuits to the IBM QX architectures". In: *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden, Germany: IEEE, Mar. 2018, pp. 1135–1138. ISBN: 978-3-9819263-0-9. DOI: 10.23919/DATE.2018.8342181. URL: http://ieeexplore.ieee.org/document/8342181/ (visited on 11/11/2022).