Sequencing Universal Quantum Gates for Arbitrary 2-Qubit Computations

Taegun An
Dept. of CSE
Korea University
Seoul, Korea
antaegun20@korea.ac.kr

Hoon Ryu
Korea Institute of
Science and Technology Information
Daejeon, Korea
elec1020@kisti.re.kr

Changhee Joo*
Dept. of CSE
Korea University
Seoul, Korea
changhee@korea.ac.kr

***Abstract*—With recognition of quantum computer's enormous computational ability, it is of paramount importance to develop fault-tolerant quantum computing systems for their practical use. Recently, it has been shown that fault-tolerant systems can be achieved using a small set of basic quantum operations. This, however, incurs technical difficulties in finding an optimal sequence of basic operations toward a specific target computation and may limit possible quantum computations. In this work, we aim to achieve arbitrary target quantum computations under the restriction of four universal quantum gates of Pauli-X, -Y, -Z and SWAP. We develop two gate-sequence search methods based on the fidelity measure and deep neural networks. We verify the performance of our proposed methods through numerical results comparing total search space and the number of searched nodes.**

I. Introduction

Quantum computing exploits the quantum mechanical properties such as entanglement or superposition in computation process. Such properties enable polynomial-time solution for classical NP-hard combinatorial problems like integer factorization [1] or black-box optimization [2]. With its potential on such combinatorial problems, quantum computing has been considered in various domains: RSA encryption-decryption in cybersecurity or black-box optimization on big social data [3].

Classical computers have been used bits, which take either 0 or 1 exclusively, as a base unit for computation. In quantum computers, a qubit which can be in superposition state of 0 or 1, is used as a base unit. Superposition state contains probability information of 0 and 1 simultaneously, represented as 2x1 matrix. As a single qubit can contain twice more information than a bit, n qubits can contain information of 2^n values simultaneously with entanglement. To deal with superposition and entanglement of n qubits, there are special operations represented as a 2^nx2^n matrix. For example, the single-qubit operation of Pauli-X gate replaces probability of 1s with 0s, and 0s with 1s, respectively, like the NOT gate in classical bit-operation circuit. Such logical operations on the qubits are called quantum logic gates.

The information capability of quantum computing is promising and will facilitate computation-intensive tasks. However, there are challenging reliability issues due to quantum errors from decoherence and other quantum noises. Quantum decoherence is caused by interaction between quantum system and environment, causing quantum information lost. Other quantum noises such as Johnson and shot noise are related with temperature and tunnel junctions respectively [4]. To deal with such errors, Quantum Error Correction (QEC) codes for detecting and correcting errors [5], [6] are proposed. Quantum syndrome measurement (SM) is one of effective schemes to tackle quantum errors by measuring parity over multiple qubits [7], but it often consumes substantial amount of time and resources, decreasing the overall performance.

Recently, the possibility to build a fault-tolerant quantum computing system without SM has been sought, and shown to be achievable if the quantum operations consist of a set of finite standard gates [8]. This approach, however, has a couple of technical challenges: (i) which set of standard gates will be enough to realize an arbitrary quantum computation, and (ii) how one can find a sequence of the gates for the given target quantum computation.

In this work, we consider a set of 4 standard gates of Pauli-X, -Y, -Z and SWAP in a 2-quit quantum system and address the problem of gate sequencing, which can be formulated as a combinatorial optimization problem. Considering the exponentially increasing complexity in finding the whole gate-sequence at a time, we take a step-by-step approach: find one appropriate gate at a time and repetitively build a whole sequence of gates. This approach fits well with reinforcement learning (RL) techniques. In the literature, there have been several interesting RL works for combinatorial optimization problems, including local rewriting algorithm iteratively improving sequences toward an optimal one [9], and the work addressing Rubik's cube problem along with A^* search to efficiently handle large search space [10].

In this work, we develop a deep-RL based framework to accomplish an arbitrary target quantum computation by using only 4 basic quantum gates of Pauli-X, -Y, -Z and SWAP. We use A^* search as in [10], but consider two different methods to approximately measure distance between states: computing fidelity and approximating the number of necessary gates with neural networks. We demonstrate that several important quantum gates, such as Controlled-Z (CZ) and CNOT, can

C. Joo is the corresponding author. This work is supported in part by the NRF grant funded by the Korea government (MSIT) (No. NRF-2021R1A2C2013065)

be constructed with the 4 basic gates and their sequences can be found with a small budget of computation. Although the results are preliminary, our technique is promising in automating the design of quantum circuits for fault tolerant quantum computing system.

II. System Model

We first describe the system settings and formulate the problem. Then we develop our solution and verify it through numerical results.

We consider a quantum system with 2 qubits, which have four possible quantum states of $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. A system state can be represented by a complex-number amplitude vector $(a_{00}, a_{01}, a_{10}, a_{11})$, whose element-wise square corresponds to a distribution over the quantum states. Since a quantum operation on a single qubit can be represented by a 2x2 matrix, an arbitrary target quantum computation in the two-quibit system can be represented as a 4x4 matrix. For example, the frequently-used quantum gates of Controlled-Z (CZ) gate and CNOT gate can be represented as follows:

$$\text{CZ}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{pmatrix},\ \text{CNOT}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

In this work, given a target quantum computation as a 4x4 matrix, we aim to find the shortest sequence of quantum gates that is composed of only four unitary basic gates of Pauli-X, -Y, -Z and SWAP operations.

$$\text{X}=\begin{pmatrix}0&1\\1&0\end{pmatrix},\ \text{Y}=\begin{pmatrix}0&-i\\i&0\end{pmatrix},\ \text{Z}=\begin{pmatrix}1&0\\0&-1\end{pmatrix},\ \text{SWAP}=\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix} \quad (1)$$

Note that Pauli gates are a single-qubit rotation gate and can be applied to either of the two qubits. In two-qubit system, the operation of a Pauli gate to one qubit can be expressed by a Kronecker product with the 2x2 identity matrix. Considering the combination of basic gates with two qubits, we have total 7 different basic gate operations: 3 Pauli gates to each qubits plus one SWAP gate operation. Let X_i denote the operation of applying Pauli-X gate to qubit $i \in \{1, 2\}$. Similarly, let Y_i, Z_i be Pauli-Y and -Z operation on qubit $i \in \{1, 2\}$. Also let S_{12} denote the SWAP operation that is applied to both qubits as shown in (1).

Although any quantum circuit can be composed of the 7 basic gate operations [11], we also consider their rotation phases. Since Pauli-X, -Y and -Z gates have physical meaning of rotating particles, their rotating operations can be further differentiated by *rotation phase* ϕ. For example, with $\phi = \pi$, rotation gate matrices are as in (1), and with $\frac{\pi}{2}$, they become the square root matrix to (1). we consider six rotation phases for ϕ; $\pm\pi, \pm\frac{\pi}{2}, \pm\frac{\pi}{4}$, which are frequently used to compose many quantum gate operations including Hadamard gate and T gate. Applying the six rotation phases to the 7 basic operations, we have total 42 operation options[1] in the action space.

[1]We note that the 42 options are in fact redundant and subject to further optimization. For example, $+\pi$ and $-\pi$ means the same rotation by π, and $\pm\frac{\pi}{4}$ for SWAP is not widely used.

Our goal is to find the sequence of these 42 operation options composed of 4 basic quantum gates, and accomplish an arbitrary target quantum computation in the two-qubit quantum system. We evaluate the performance of a sequence by its length and fidelity. Fidelity measures the similarity of two quantum computations (e.g., x and y), and defined as $|Tr(x^T y)|^2$. In two-qubits system, the more similar the matrices are, larger the fidelity is, up to 16. On the other hand, a shorter length of sequence saves the execution time and resources, and thus reduces the amount of noise as well as the cost [12].

III. Proposed Method

We define an (intermediate) quantum computation as a state, which can be represented by a 4x4 matrix. Starting from an arbitrary quantum computation, also given as a 4x4 matrix, we try to find a sequence of 42 operation options to reach the identity matrix. Once we find the sequence, the reverse order of the operations is the gate sequence that we are looking for.

We use A^* search to find the sequence to the identity matrix. It is a tree-based graph search algorithm that finds the shortest path from the root node to a terminal node. In our case, each node n is a state represented by a 4×4 matrix. We use 'node' and 'state' interchangeably. We define two sets of 'open set' and 'close set' where the former is the set of unexplored nodes and the latter is the set of explored non-terminal nodes. Starting from root node x_0 (which corresponds to the target quantum computation), the search is guided by the value of nodes $f(x)$ in the open set, which is the sum of two functions $g(x)$ and $h(x)$, where $g(x)$ is the number of basic blocks up to state x, and $h(x)$ is an estimate on the number of basic blocks from x to the terminal state t, i.e., the identity matrix. We will describe two estimation methods for $h(x)$ later. At each step t, A^* search chooses node x in the open set with the lowest value $f(x)$, create at most 42 nodes in the open set (for each operation option), and then moves x to the close set. The search repeats until we search a predetermined number of nodes the terminal state. We use a batch setting, where N nodes of lowest values $f(n)$ are selected for expansion at a step.

For the estimation of $h(x)$, we use two different estimation methods: fidelity-based estimation and neural networks (NN)-based estimation.

A. Fidelity based value estimation

Fidelity is a measure about how similar two quantum computations are. In the fidelity-based estimation, we use $h(x_t) = w_f \cdot h_f(x_t)$, with $h_f(x_t) = 16 - |Tr(x_t^T x_T)|^2$, where x_t^T is the conjugate transpose of state x_t, and x_T is the terminal state of the identify matrix. This, however, does not tell us how many basic operations are required to reach x_T, and instead, tell us how similar state x_t is with the terminal state. The weight factor $w_f = \frac{5}{8}$ is used to compensate the differences between the number of basic operations and the similarity.

B. Neural network based value estimation

An alternative way to estimate $h(x)$ is to make use of a deep neural network (NN), to predict the number of necessary basic operations to the terminal state from state x.

We design a neural network that consists of 3 linear layers followed by 4 residual blocks. After each linear layer, a batch normalization and a rectified linear unit are applied. It can be denoted as a function parameterized by θ as $h_{nn}(x;\theta)$. We train the network to predict the number of basic operations from x to the terminal state t. We generate training data as follows. First we randomly choose an integer $d \in [1, 7]$, and obtain a sequence of d basic operations randomly sampled from 42 options allowing duplicates. Then we compute the matrix of state x corresponding to the sequence, yielding one sample data (x,d). By repeating this procedure, we can synthesize labeled data (x,d) to train the neural network, which is used to optimize parameter θ. After training, we set $h(x_t) = h_{nn}(x_t;\theta)$.

IV. Numerical Results and Conclusion

In this section, we numerically evaluate our proposed algorithms. We fix a length d of target quantum computation, and generate a target quantum computation by randomly sequencing d basic operation options. Then we give the target quantum gate matrix of the sequence as input of search algorithm, and measure how many searches will be proceeded to find the sequence. We repeat this for different 50 test sequences for each of $d \in [1, 7]$ length. Additionally, we include the widely-known gates of CZ and CNOT in our test. We use Python 3.8.5 with CUDA 11.2, CUDNN 8.1.1, and pytorch 1.8.0. At each step of A^* search, we selected 16,000 nodes of the smallest $f(\cdot)$, and expand them.

We measure the number of searched nodes until the search algorithms find the target quantum gate matrix. The results in Fig. 1 show that our search algorithms with fidelity-based estimation (green) and with NN-based estimation (purple) find the target sequence with almost-linear complexity for substantial sequence length ($d > 5$).

Next, we focus on the problem of finding sequences for CZ (with $d = 5$) and CNOT (with $d = 6$) gates using the fidelity-based method. For the same target quantum circuit, there can be many different circuit implementations with different sequence of basic operation options. We limit the number of searched nodes to 10^7, and measure how many sequences can be found under the constraint and how quickly the algorithm achieves the first finding. The results are shown in Fig. 2(a), which confirms that there are many different circuit implementations (up to 36 for CZ). In comparison between the fidelity-based and the NN-based algorithm, the results show that, under the given constraint on the searched nodes, the fidelity-based one can find more circuits and also achieves the first finding earlier, at least for the CZ and CNOT cases.

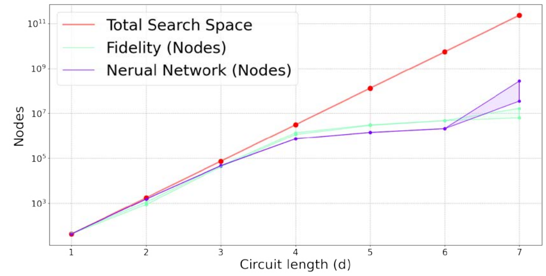
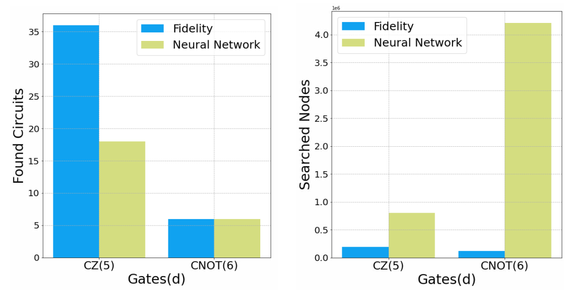


Fig. 1. Performance of the proposed search algorithm with fidelity-based estimation and NN-based estimation.



(a) Number of found circuits (b) Average searched nodes

Fig. 2. Performance comparison of fidelity-based estimation and NN-based estimation in searching for 2-qubit quantum circuits of CZ (with $d = 5$) and CNOT (with $d = 6$).

References

[1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing*, vol. 26, no. 5, p. 1484–1509, Oct 1997.

[2] G. Brassard and P. Hoyer, “An exact quantum polynomial-time algorithm for simon’s problem,” in *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*, 1997, pp. 12–23.

[3] F. Bova, A. Goldfarb, and R. G. Melko, “Commercial applications of quantum computing,” *EPJ Quantum Technology*, vol. 8, no. 1, Jan 2021.

[4] R. J. Schoelkopf, A. A. Clerk, S. M. Girvin, K. W. Lehnert, and M. H. Devoret, *Qubits as Spectrometers of Quantum Noise*, 2003, pp. 175–203.

[5] J. Roffe, “Quantum error correction: an introductory guide,” *Contemporary Physics*, vol. 60, no. 3, pp. 226–245, 2019.

[6] A. Steane, “A tutorial on quantum error correction a tutorial on quantum error correction 2,” vol. 162, 01 2006.

[7] Y. S. Weinstein, “Syndrome measurement strategies for the [[7,1,3]] code,” *Quantum Information Processing*, vol. 14, no. 6, Jun 2015.

[8] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, “Universal quantum control through deep reinforcement learning,” *npj Quantum Information*, vol. 5, no. 1, p. 33, Apr 2019.

[9] X. Chen and Y. Tian, “Learning to perform local rewriting for combinatorial optimization,” in *NeurIPS*, vol. 32, 2019.

[10] F. Agostinelli, S. McAleer, A. Shmakov, and P. Baldi, “Solving the rubik’s cube with deep reinforcement learning and search,” *Nature Machine Intelligence*, vol. 1, no. 8, pp. 356–363, Aug 2019.

[11] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 2010.

[12] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O’Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, “Superconducting quantum circuits at the surface code threshold for fault tolerance,” *Nature*, vol. 508, no. 7497, pp. 500–503, Apr 2014.