

Qubit Arrangement Problems for Topological Quantum Computation

Shigeru Yamashita¹ *

Shinnosuke Hiratsuka¹ †

Simon Devitt² ‡

Kae Nemoto² §

¹ College of Information Science and Engineering, Ritsumeikan University

² National Institute for Informatics

Abstract. We formulate the logic level circuit optimization problem for *topological quantum computation*. Observing the properties of *braiding operations* in topological quantum computation, we formulate our problem as to find a good gate order and a good initial qubit one-dimensional layout in the conventional quantum circuit model. The optimization of initial qubit order is NP-hard, thus we then propose an efficient optimization method by utilizing an algorithm for clique finding. Moreover, we show that using two dimensional qubit layouts can optimize the circuit further, and propose a heuristic to find a good qubit layout order for topological quantum computation.

Keywords: Topological Quantum Computation, Qubit Arrangement, Two-dimensional

1 Introduction

To realize a quantum computation, we need to have *fault-tolerant* quantum gates, i.e., quantum gates with a very low operational error rate. We use quantum error correction codes for that purpose in the conventional quantum circuit model.

Topological quantum computation [1] is another possible way to have fault-tolerant quantum gates. Recently, this model of quantum computation has been considered to be much more promising than conventional model of quantum circuits in terms of error corrections. The way of encoding logical qubits in topological quantum computation is very different from the conventional quantum circuit model, and thus the logical primitive operations are also very different. The primitive operation called a *braiding operation* can be seen as drawing a line between logical qubits with some special rules. We need to consider such special rules when we design a quantum circuit. Therefore, it should be difficult to utilize the conventional quantum circuit design naively, and thus it is desirable to have a dedicated quantum circuit optimization method in logic level with considering special rules of braiding operations.

2 Our Contribution

In our work, we formulate a quantum circuit optimization problem especially for the topological quantum computation. First we observe that our design strategy should be different from the conventional quantum circuit design as follows. To understand the difference, let us see the circuit in Fig. 1. In the conventional quantum circuit model, we often assume that multiple CNOT gates can be performed at the same time if their interacting qubits are different, and the *depth* of a circuit is calculated based on this assumption. For example, we can perform g_1 and g_2 in the circuit in Fig. 1 at the same time. The important observation here is that such a relation of two gates does not change even if we change

the qubit order (qubit layout). Thus, in the conventional quantum circuit design, we do not need to consider the qubit order.

Contrary to the above, in topological quantum computation, we can assume any two gates can be performed parallelly only if their gate symbols on the circuit diagram are not overlapped in the horizontal direction. (The rigorous discussion can be found in the attachment.) For example, we cannot perform g_1 and g_2 in the circuit in Fig. 1 at the same time unlike the conventional circuit model.

Therefore, the qubit orders (i.e., qubit layout) may be really important for the computation time for topological quantum computation. Our problem is intuitively to find a good qubit order for a given circuit as shown in Fig. 1. By our proposed method, we can optimize the circuit in Fig. 1 to the one in Fig. 2. Here, the two circuits are logically equivalent but with different initial qubit orders. The number of the logical time steps in the circuit in Fig. 1 is 8, which is optimized by our method to be 3 as shown in Fig. 2.

Moreover we consider two-dimensional qubit layout. In one-dimensional qubit layout, it can be shown that one single qubit order does not allow us to perform the above circuit with two time steps. For example, at the one-dimensional qubit layout of the qubit order as shown in Fig. 2, the three gates, g_4 , g_1 and g_5 (or g_7), are overlapped with each other; we need at least three time steps. In contrast, if we layout the qubits two-dimensionally as shown in Fig. 2, we can perform the circuit with only two logical time steps. This is because the two-dimensional qubit layout allows us to perform g_1 , g_2 , g_3 and g_4 at the same time as shown on the left-hand side of Fig. 3. Our proposed method can find such a good two-dimensional qubit layouts efficiently. As far as we know, our method is the first systematic synthesis method for topological quantum circuits by considering two-dimensional qubit layouts. Both of our optimization methods for one-dimensional and two-dimensional layouts utilize clique finding efficiently.

*ger@cs.ritsumei.ac.jp

†kita@ngc.is.ritsumei.ac.jp

‡devitt@nii.ac.jp

§nemoto@nii.ac.jp

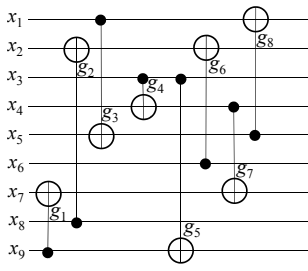


Figure 1: An Initial Circuit: 8 Steps.

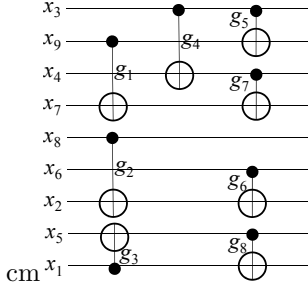


Figure 2: The Optimized One-Dimensional Qubit Layout by Our Method: 3 Steps.

3 Optimization Algorithms for Qubit Arrangement Problems

The k -th CNOT gate in a given CNOT-based circuit is denoted by g_k , and the target and the control qubits of gate g_k are denoted by $T(g_k)$ and $C(g_k)$, respectively. First we introduce a terminology “overlapped” as follows.

Definition 1 A pair of consecutive gates g_{k+1} and g_k are said to be **overlapped** with a given qubit order if the group of qubits placed between $T(g_k)$ and $C(g_k)$, and the group of qubits placed between $T(g_{k+1})$ and $C(g_{k+1})$ do not have a common qubit with the given qubit order. If g_{k+1} and g_k are not overlapped, they are said to be **non-overlapped** with each other.

Our algorithm for one-dimensional layouts is as follows. Let Q be a given circuit. In our method for one-dimensional layouts, we maintain a set of qubit orders, denoted by P . The initial P is the set of all the possible qubit orders. Let the current target circuit to find parallel gates with the largest group size be C . The initial C is set to Q .

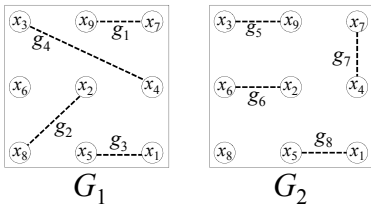


Figure 3: The Optimized Two-Dimensional Qubit Layout by Our Method: 2 Steps.

Step 1. Construct a graph where each node corresponds to each gate in C , and we have an edge between two nodes iff the corresponding two gates in C are adjacentable in the given Q and non-overlapped with at least one qubit order in P .

Step 2. Find a maximum clique in the graph constructed at Step 1. (If there are more than one maximum clique, just choose one of them randomly.)

Step 3. Let G be the set of gates corresponding to the nodes in the maximum clique at Step 2.

Step 4. Let P' be the set of qubit orders such that all the gates in G can be non-overlapped with the qubit orders. Update P as $P \cap P'$. Also update C by removing all the gates of G from C . If C becomes empty, finish the procedure and the solution can be one of the qubit orders in P . Otherwise, goto Step 1 with the updated C and P .

In the case of two-dimensional layouts, we need to modify the terminology “overlapped” as follows: A pair of consecutive gates g_{k+1} and g_k are said to be **overlapped** with a given two-dimensional qubit layout if the line between $T(g_k)$ and $C(g_k)$ and the line between $T(g_{k+1})$ and $C(g_{k+1})$ cross each other in the given two-dimensional qubit layout. If g_{k+1} and g_k are not overlapped, they are said to be **non-overlapped** with each other.

Let C be a given circuit, and our algorithm for two-dimensional layouts is as follows.

Step 1. Construct a graph where each node corresponds to each gate in C , and we have an edge between two nodes iff the corresponding two gates in C are adjacentable.

Step 2. Partition the graph obtained at Step 1 into minimal number of cliques, C_1, C_2, \dots, C_m . (This is called a clique cover problem.)

Step 3. Divide the gates in C into groups, G_1, G_2, \dots, G_m from the beginning of the circuit such that each G_i corresponds to one of the cliques obtained at Step 2. In other words, the original circuit is equivalent to performing the gates in G_1, G_2, \dots, G_m in this order.

Step 4. For each G_i there is at least one one-dimensional qubit order such that all the gates in G_i can be performed at the same time. Let the condition of such a qubit order for G_i be $Cond_i$. Find a two-dimensional qubit layout such that as many $Cond_i$ as possible can be satisfied.

References

[1] Austin G. Fowler, Ashley M. Stephens, and Peter Groszkowski. High threshold universal quantum computation on the surface code. *PHYS.REV.A*, 80:052312, 2009.