Report on "Effective and Efficient Qubit Mapper" by Ayan Barui

1 Introduction

Quantum algorithms must be run on present-day Noisy Intermediate Scale Quantum (NISQ) devices to demonstrate speedup over classical computers. However, the architecture of NISQ devices poses a problem: a quantum algorithm must be searched for possible permutations of $n!$ (n is the number of qubits) in the worst case, coupled with the fact that a qubit has a limited coherence time. To solve this, efficient ways of mapping and routing are being researched to minimize circuit depth. Schemes like the SMT Z3 solver and A* search algorithm have been developed. "Effective and Efficient Qubit Mapper" uses the A* algorithm in a quantum-window-based approach named **SWin** to minimize circuit depth. **SWin** breaks the quantum circuit into different layers and optimizes each layer according to a predetermined window size.

2 The SWin Algorithm

The architectures of the quantum chips represent a Directed Acyclic Graph (DAG), so a quantum circuit P with quantum registers m (represented as Q_l) is also converted into DAG G_P. Note that the operations in P contain only single qubit and CNOT gates. The nodes on this G_P represent quantum gates, and the channels are represented as edges. The depth of this quantum circuit is D_P. However, the quantum chip C contains a qubit set Q_P, and its undirected coupling graph is represented as $G_C = (Q_P, E_P)$. The edge is defined as e_{ij} between q_i and q_j, where $q_i, q_j \in Q_P$ and $e_{ij} \in E_P$. We define a mapping $M = f : Q_l \rightarrow Q_p$ during circuit transformations to physical qubits.

The mapping result is represented as $\{M^I, P^T, C\}$, where M^I is the initial mapping and P^T is the transformed circuit which matches the architecture of quantum chip C. To achieve this mapping, two conditions are proposed, namely:-

1. **Immediate execution** - Executing the current operation if the gate aligns with the quantum chip architecture.
2. **Online SWAP Insertion** - When the quantum chip architecture prevents the execution of the current operation, appropriate SWAP gates are inserted.

The following points describe the use of **SWin** as an algorithm.

1. Given a quantum circuit of depth D_P, the window size S_d is set to an initial value. If the circuit depth $D_P < S_d$, then starting with an initial mapping M^I, we obtain the best final mapping M^F of the given quantum circuit on quantum chip C.
2. However, if $D_P > S_d$, the original circuit is broken into many layers. The first layer is encountered with initial mapping M_0^I, which transforms the circuit into P_0^T, stored into p_0, and with final mapping M_0^F.
3. For the subsequent layers ($i = 1, 2, ...$), the initial mapping M_i^I of each of the layers is the same as that of the final mapping of the previous layer M_{i-1}^F. This is repeated until $D_P = S_d$. The corresponding transformed circuits are stored in p_i. The value of i is incremented by one after mapping each subcircuit.

4. After the second step is executed, the whole circuit is routed for an optimal solution (P^T_{r+1}, M^I_r, C), where r is the number of times the second step was executed. The final transformed quantum circuit is sequentially connected from p_0 to p_i. $P^T = \{p_0, p_1, ..., p_{r+1}\}$

The A* heuristic algorithm [1] is chosen as the optimal circuit mapper for the above algorithm.

3 The A* Algorithm

The A* algorithm can be realized in the following steps.

1. To denote the number and cost of gates in a quantum circuit, a quantity G is defined as $G = g_1 g_2 ... g_{|G|}$. The nearest neighbor cost (NNC) is a measure of the cost of a two-qubit gate. For a certain gate g the NNC is given by Eq. 1

$$NNC(g) = |c - t| - 1 \tag{1}$$

where c is the control of g at line c and t is the target of g at line t. Single qubit gates are said to have $NNC = 0$. The resulting NNC cost of the whole quantum circuit is given as

$$NNC(G) = \sum_{g \in G} NNC(g) \tag{2}$$

This can be understood with a simple quantum circuit.

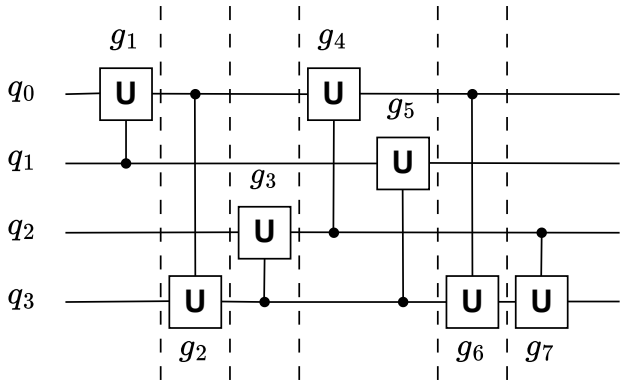


Figure 1: A quantum circuit of four qubits ($n = 4$), with seven two-qubit gates (denoted as $g_1, g_2, ..., g_7$). The circuit is broken into six layers.

Now $NNC(g_1, g_3, g_7) = 0, NNC(g_2, g_6) = 2$ and $NNC(g_4, g_5) = 1$. The total NNC cost of the quantum circuit given in Fig. 2 is given by

$$NNC(G) = \sum NNC(g) = (3 \times 0) + (2 \times 2) + (1 \times 2) = 6 \tag{3}$$

2. To make the quantum circuit nearest neighbor compliant (i.e., to make the $NNC = 0$), we prepare a global rearrangement of the qubits and propose the construction of an adjacency matrix. The adjacency matrix M of Fig. 2 is given by Table 1, of dimension $n \times n$. The entries are denoted by $m_{i,j}$ $(0 \leq i, j < n)$.

For example, the number of gates connecting q_0 and q_3 (denoted by $m_{0,3}$) is two, which can be understood from the given adjacency matrix. ' - ' denotes "Don't

Table 1: Adjacency matrix for Fig. 2

	q_0	q_1	q_2	q_3
q_0	-	1	1	2
q_1	-	-	0	1
q_2	-	-	-	2
q_3	-	-	-	-

care" to avoid information redundancy. The NNC cost using the adjacency matrix is calculated by

$$NNC(M) = \sum_{i=0}^{n} \sum_{j=i+1}^{n} m_{i,j}(j - i - 1) \tag{4}$$

$NNC(M)$ of Fig. 2 is given by $m_{0,1}.0+m_{0,2}.1+m_{0,3}.2+m_{1,2}.0+m_{1,3}.1+m_{2,3}.0 = 6$ where $m_{0,2} = 1, m_{0,3} = 2$ & $m_{1,3} = 1$.

For a permutation of the position of qubits (denoted by π), the cost is determined by

$$NNC(M, \pi) = \sum m_{i,j}.(|\pi(j) - \pi(i)| - 1) \tag{5}$$

3. The A* algorithm considers the problem in terms of state nodes. A feasible solution (with low cost) is represented by the path taken from a root node (initial point) to a goal node (final node). A function $f(x) = g(x) + h(x)$ is defined to determine the cost of a node x. $g(x)$ is the path's cost from the root node to x, and $h(x)$ is the cost of x to the goal node. In Fig., as node B has the lowest cost of 70, it is expanded, and then we have $f(E) = (70 + 50) + 50 = 170$ as the lowest cost, which will be chosen as the final path.

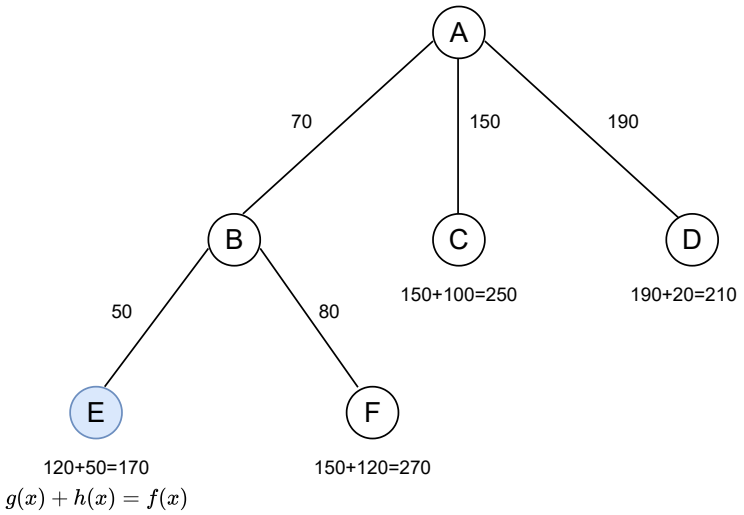


Figure 2: Diagram showing different nodes and their respective costs of $g(x)$ and $h(x)$. the path with the lowest cost is chosen as the feasible solution.

Two strategies, namely the straightforward strategy and elaborate strategy, are further proposed to make this heuristic approach more efficient. Suppose for a node labeled $q_1q_3q_0$, we have to determine the cost of $g(x)$. In this case, the NNC of $g(c, t)$ for which $c, t \in \{0, 1, 3\}$ is calculated by Eq. 5. Using the adjacency matrix, we can find the NNC as

Permuted (π)	0	1	2
Initial	q_1	q_0	q_3

the NNC for the permuted setup is $(m_{0,1}.0 + m_{0,2}.0 + m_{1,2}.1)$. Permutation takes place owing to the insertion of swap gates to make the NNC low.

4 Experimental examples

The coding experiments used quantum circuits obtained from the QASM Bench [2]. The transpiler function in Qiskit is [3], also used in our results. The A* algorithm is inserted in the Munich Quantum Toolkit (MQT) [4]. The architecture used for the experiments is the form of Fig 1a in the reference paper [5].

The coupling map in Table 2 has been constructed according to the above-said architecture. For the IBM Qiskit compilation, the optimization level was set to 3 in all the cases. For the first case, fredkin has only three qubits in its parent circuit. The basis gates used are 't','tdg','x','cx','h', and 'swap'. Still, when applied to the four-qubit architecture, the compilers take the fourth qubit as ancillary and involve SWAP operations that carry out the result. The swap gates are added with the CNOTs, which shows better performance by the A*.

Table 2: Experimental Results

Benchmark Circuit	No of Qubits	Depth	IBM Qiskit		A* Algorithm	
			Depth	No. of CNOT	Depth	No.of CNOT
fredkin	3	12	14	8 (+3 SWAPs)	14	8 (+2 SWAPs)
hs4_n4	4	10	8	4	9	4
bell_n4	4	14	8	7	13	7
cat_state	4	5	5	3	5	3

For the circuit of hs4_n4, the A* algorithm was unable to efficiently synthesize the Hadamard gates (for example, $H.H = I$ was not optimized), which led to an increase in the depth of the compiled circuit. In the experiments conducted in the reference paper [5], the SWin window size was equal to or greater than the depth of the original circuit. Thus, the results of the A* algorithm would be the same as that of the SWin.

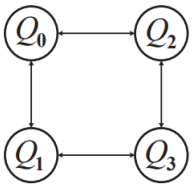
A demo quantum circuit is compiled using different mappers given in Fig. 3. It is to be noted that the SMT solver Z3, when compiling, does not include a SWAP gate.

References

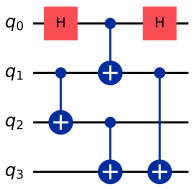
[1] Alwin Zulehner, Stefan Gasser, and Robert Wille. Exact global reordering for nearest neighbor quantum circuits using a. In *International conference on reversible computation*, pages 185–201. Springer, 2017.

[2] Ang Li, Samuel Stein, Sriram Krishnamoorthy, and James Ang. Qasmbench: A low-level qasm benchmark suite for nisq evaluation and simulation. *arXiv preprint arXiv:2005.13018*, 2020.

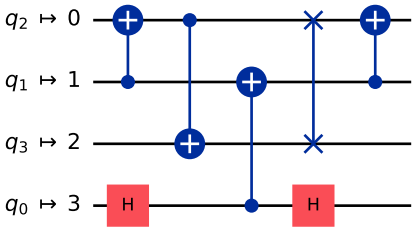
[3] Sajid Anis. Qiskit: An open-source framework for quantum computing. *(No Title)*, 2021.



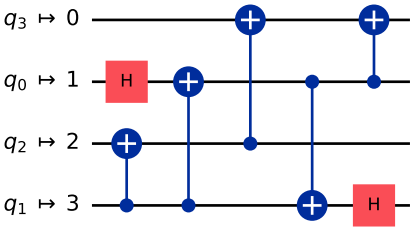
(a) Architecture according to which the circuits are mapped



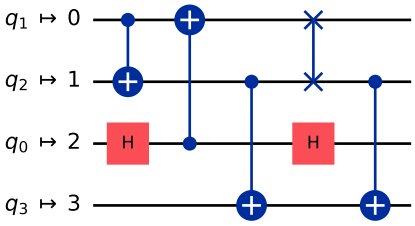
(b) Original circuit



(c) Complied with qiskit (Optimisation level = 3)



(d) Compiled with SMT solver Z3



(e) Compiled with A*

Figure 3: Mapping of demo quantum circuit according to different quantum compilers

[4] Robert Wille and Lukas Burgholzer. Mqt qmap: Efficient quantum circuit mapping. In *Proceedings of the 2023 International Symposium on Physical Design*, pages 198–204, 2023.

[5] Hao Fu, Mingzheng Zhu, Jun Wu, Wei Xie, Zhaofeng Su, and Xiang-Yang Li. Effective and efficient qubit mapper. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2023.