Monte Carlo Qubit Routing

Alexis Hummel and Tristan Cazenave

LAMSADE, Université Paris Dauphine - PSL, CNRS, Paris, France

Abstract. We compare four Monte Carlo Search algorithms for Qubit Routing. We find that Nested MCTS improves on previous MCTS algorithms for Qubit Routing and that simple Monte Carlo Search algorithms not using neural networks can also solve Qubit Routing problems.

Keywords: Monte Carlo Search · Quantum Computing.

1 Introduction

In quantum computing, data are stored in qubits, which can take two basis states $|0\rangle$ and $|1\rangle$. But unlike classical bit, a qubit can be a superposition of two basis states : $\alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in C$ s.t $|\alpha|^2 + |\beta|^2 = 1$. Nowadays, quantum computers devices come in different form of hardware architecture, but there are few problems concerning all of them, which include the limited connection between qubits. We can see a quantum hardware as a graph, where nodes represent a physical qubit and the set of edges the connectivity between them. A quantum algorithm (described as a quantum circuit) is a sequential series of logical qubit with a set of gates operations that act on them. A gate operation can only be performed iff there is an edge (i.e there is a connection) between the physical qubits who are mapped with the logical qubits of the gate.

2 Related Works

2.1 Qubit Routing

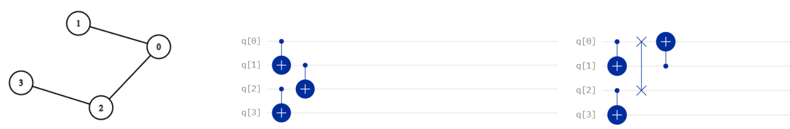
In order to run a quantum circuit with a given QPU we need to compile it, which involves a *decomposition process* : gates in the logical circuit are decomposed or transformed into gates supported by the QPU, and QCT (*quantum circuit tranformation*). QCT splits into two steps : *initial mapping construction* which maps qubits from the logical circuits (logical qubits) to qubits in the given QPU (physical qubits), and *Qubit Routing*. We will focus on this last step.

Given a QPU, and a quantum circuit, Qubit routing problem consists of transforming this circuit by adding SWAPs gates such that all its gates operations satisfy the constraints of the QPU. SWAP(n1,n2) permutes nodes n1 and n2 if they are connected in the graph topology By the decomposition process and the initial mapping constructions, the logical circuit is already mapped with the physical qubits and is divided into layers such that every gates in the same layers can be executed at the same time. A layer (or slice) is composed by several gates (for example $CNOT$) such that none of them share a common qubit. We will talk about the size of a circuit to designate his number of layers.

Formally, with a given the qubits connectivity graph $D = (V, E)$ associated with the QPU, where D is undirected, V denotes the set of qubits and E represents the connectivity, and a quantum circuit C, a Qubit Routing R can be represented like this :

$$R(C, D) \to C'$$

The aim is to try to minimize the size of C' ,in order to make the quantum algorithm faster. An approach for this problem is using the MCTS.



(a) Hardware topology (b) Quantum circuit of size 2 (c) Transformed quantum circuit of size 3

Fig. 1: Very simple example of Qubit Routing
Here, q1 and q2 are not connected in the hardware topology, which make the CNOT(q1,q2) gate not executable. Therefore, an additional step with SWAP(q0,q2) is required.

2.2 Monte Carlo Search for Qubit Routing

In this paper we compare four different Monte Carlo Search algorithms for Qubit Routing.

First MCTS taken from previous work [4, 3, 5]. MCTS iteratively uses four phases : select, expand, playout, and backup. At each iteration, it selects the node which provides the best PUCT value. The PUCT bandit is:

$$PUCT(s, a) = Q(s, a) + c\frac{\sqrt{\sum_a N(s,a)}}{1+N(s,a)}p(s, a)$$

where (s, a) is a state-action pair, $N(s, a)$ is the number of times a has been played from s. $Q(s, a)$ is the average long-term reward and $p(s, a)$ is the prior obtained with the neural network.

The other algorithms we evaluate are random playouts, Nested Monte Carlo Search (NMCS) [2] and Nested MCTS [1]. The principle of Nested MCTS and NMCS is to use MCTS to perform high level playouts. The moves of the high level playouts are chosen by lower level NMCS in the case of NMCS and by

MCTS in the case of Nested MCTS. The principle of NMCS is depicted in Figure 2. At each step of the higher level playout there are three possible actions. The algorithm performs a lower level playout after each action and chooses the action associated to the best score.

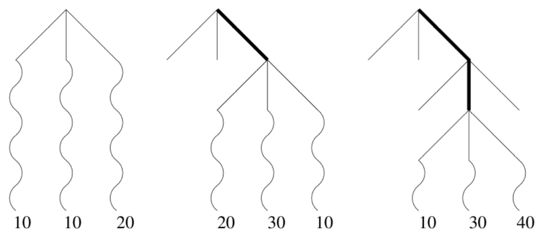


Fig. 2: Nested Monte Carlo Search.

3 Experimental Results

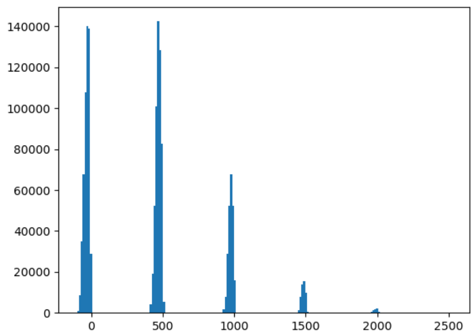
In this section evaluate Monte Carlo search algorithms on a quantum circuit with 3089 CNOT gates. All of the experiments are executed in the same amount of time. Figure 3a give the distribution of scores of random playouts. Figure 3b gives the distribution of the scores of a the MCTS algorithm from [4, 5]. Figure 3c gives the distribution for a NMCS of level 1 with random playouts and Figure 3d gives the distribution for a Nested MCTS. The average score of MCTS runs is 2044 while the average score of Nested MCTS scores is 2219. Random playouts and NMCS do not use a neural network and still have high scores comparable to MCTS which is much more complicated.

4 Conclusion

We have found that random playouts and NMCS using a uniform playout policy and no neural network can perform on par with previous work using MCTS with PUCT and a trained neural network. This is a much more simple approach to efficient Qubit Routing. We also found that Nested MCTS has better results on average than MCTS for Qubit Routing.

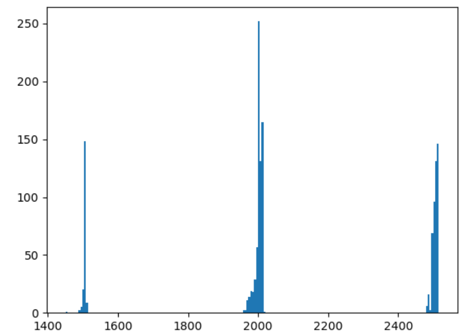
References

1. Baier, H., Winands, M.H.: Nested monte-carlo tree search for online planning in large mdps. In: ECAI. vol. 242, pp. 109–114 (2012)
2. Cazenave, T.: Nested monte-carlo search. In: IJCAI. pp. 456–461 (2009)
3. Sinha, A., Azad, U., Singh, H.: Qubit routing using graph neural network aided monte carlo tree search. arXiv preprint arXiv:2104.01992 (2021)



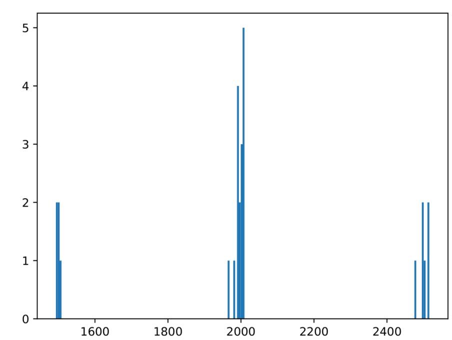
(a) Playouts

Best score = 2516 average score = 401 Execution time = 1500s : 0.00056s per iteration. (1 344 505 iterations)



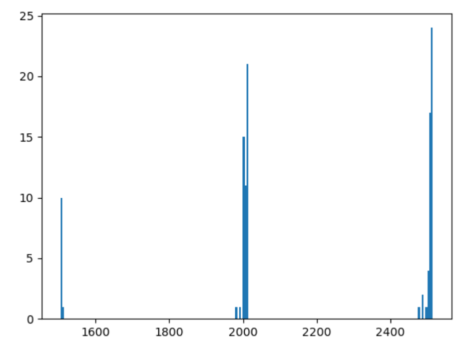
(b) MCTS

Best score = 2516 average score = 2107,76. Execution time = 1501s : 1,11s per iteration. (1352 iterations)



(c) NMCS of level 1

Best score = 2516 average score = 2016,85. Execution time = 1516,8s : 56,17s per iteration. (27 iterations)



(d) Nested MCTS

Best score = 2516 average score = 2182,48. Execution time = 1502s : 13,7s per iteration. (109 iterations)

Fig. 3: Distribution of the scores for the four Monte Carlo Search algorithms.

4. Zhou, X., Feng, Y., Li, S.: A monte carlo tree search framework for quantum circuit transformation. In: Proceedings of the 39th International Conference on Computer-Aided Design. pp. 1–7 (2020)
5. Zhou, X., Feng, Y., Li, S.: Quantum circuit transformation: A monte carlo tree search framework. ACM Transactions on Design Automation of Electronic Systems (TODAES) (2022)