BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

Quantum Circuit Synthesis using MILP and SMT Encodings

Jakub Havlík*

Abstract
Quantum circuit synthesis is a wide field of study in quantum computing. More specifically, quantum architecture search is a complex problem that aims to implement optimal, resource-efficient quantum circuits and quantum algorithms, since even with hundreds of qubits, a poorly optimized circuit renders the hardware's power useless. Furthermore, quantum algorithms are hard to design by hand, and automated synthesis is a key step to creating efficient implementations.

*xhavlij00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

The field of quantum computing [1] is at a continuous state of growth, and many progressions in the realization of quantum computers have been made in recent years. However, the race for the highest number of supported physical *qubits* is not enough to achieve the *quantum advantage*. To leverage the power of quantum hardware, resource-efficient implementations of quantum algorithms and circuits are needed.

The problem of quantum synthesis includes the input specification of a quantum circuit, which can be a set of vectors, a unitary matrix, or a quantum program, as well as a target gate set and some objective function – *gate count*, *T-count*... Naturally, this specification spans a huge state space, which needs to be explored to achieve optimality, and thus efficient synthesis methods are required to achieve scalable circuits.

The state-of-the-art of quantum circuit synthesis distinguishes *quantum architecture search* (*QAS*), and *quantum circuit optimization* or *compilation*. The main difference is that compilers and optimizers use rewriting rules to make a better circuit (but usually not optimal). On the other hand, QAS always builds a completely new circuit from the ground up. This means that QAS methods produce small, optimal circuits, but they lack the scalability of optimizers. Our implemented methods belong to the field of QAS, and one of the most notable works from this field is the tool `Quokka#` [2], which utilizes *#SAT* encoding.

Our solution aims to implement gate-count optimal quantum circuits using *SMT* and *MILP* encodings of quantum circuit synthesis, with the emphasis on a specific class of quantum circuits – *Repeat-until-success* (*RUS*) circuits [3].

We have managed to further optimize various state-of-the-art RUS circuits from [3], achieving exactly equivalent, but more resource-efficient operations. The comparison with the tool `Quokka#` shows promising results, since one approach cannot be directly treated as better than the other.

2. SMT and MILP encodings

This section describes our implemented approach to quantum circuit synthesis. The main parts of our encoding are the representations of quantum gates, the representations of complex numbers, and the evaluation of equivalence with a certain target.

2.1 Layer encoding

A quantum circuit can be directly encoded as a sequence of *layers* (or just *operations*). In our encoding, a layer at a depth d consists of a set of input vectors I^d, a set of output vectors I^{d+1}, and a set of Boolean selection variables Sel_d. In each layer, we allow only one selection variable to be True, and that variable essentially maps the operation on the input vectors to the output vectors. More specifically, this is encoded as implications in `SMT`, and as *indicator constraints* or `big-M` formulations in `MILP`.

2.2 Complex number representations

A key decision regarding the encoding of quantum gates and equivalence checking is the representation of complex numbers in the vectors. The standard notation $a + bj$, where $a, b \in \mathbb{R}$, can represent all operations linearly, but falls behind when it comes to SMT performance.

Another, more promising representation, is the one using a five-tuple of integers (a, b, c, d, k), where $a, b, c, d, k \in \mathbb{Z}$. These numbers belong to the ring $\mathbb{D}[\omega]$, shown in (1), which is enough to represent all standard Clifford+T operations only using integers. There are many notable optimizations, such as the fact that the k coefficient can be shared by the whole vector, reducing the representation to (a, b, c, d). This representation allows the use of QF_LIA or pure ILP formulations of the problem. This representation also avoids multiplications of complex numbers in some quantum gates, but rather utilizes a special rotation operation.

2.3 Equivalence encoding

The evaluation of the equivalence of the synthesized circuit with a certain target can be either *exact*, or *approximate*. In our work, we support both types of equivalences, but emphasize the exact equivalence, since the encoding of approximate equivalence is hard to solve, and requires the use of non-linear constraints and real variables.

The exact equivalence can be further distinguished into checking *identity* or equivalence *up to a global phase*. Encodings of these types heavily differ in the different complex number representations, since the five-tuples require a certain *rescaling* operation shown in (2), which is the main bottleneck of this representation.

3. Repeat-until-success circuits

An interesting class of circuits is that of *repeat-until-success* (*RUS*) circuits, which implement non-deterministic approximations of single-qubit operations using *ancilla qubits* and *quantum measurements*. RUS circuits show promising results in implementing complex single-qubit rotations in a very efficient way, which leads to various use-cases, such as the use in quantum neural networks.

The full *RUS protocol* can be seen in Figure 1. The protocol includes measuring all ancilla qubits and branching based on the results. If a result indicating success has been measured, the target operation is implemented on the target qubit. In case of a failure, a recovery operation has to be performed to restore the input state, and the procedure is repeated.

Our observations show that the full matrix describing RUS circuits has a very specific form and that the circuits need to be synthesized by more input states than initially expected. We have achieved further optimizations for RUS circuits from [3]. An example of a such circuit can be seen in Figure 2, with further experiments and comparisons in Table 1.

4. Experimental results

The experiments shown in Figure 3 show a comparison with our implemented methods with Quokka#. The experiments included nearly 500 randomly generated circuits up to 6 qubits. The size of the circuits ranges from 2 to 18 gates. The graph shows the cumulative instances solved in a certain time. Each circuit has been synthesized with a 300-second timeout. The results show better overall scalability of Quokka#. However, further analyses show that each approach is faster on some subset of the input circuits, so it cannot be said that one approach is better than the other.

5. Conclusions and Future Work

The implemented methods show promising results compared to the state-of-the-art tools. However, the scalability of the synthesis is still highly limited by the number of qubits. We show that state-of-the-art circuits, such as those implementing the RUS protocol, can be further optimized by our approach, achieving minimal and cheaper implementations. The main concern for future work is to expand the database of optimal RUS circuits, which directly includes improving the scalability, since we need to support longer circuits. Naturally, this task does not have a set path, but there is room for further experiments – changing the basis from vectors to density matrices, changing the circuit representation ...

Acknowledgements

I would like to thank my supervisor doc. Ing. Ondřej Lengál, PhD. for his guidance, regular consultations, helpful insights, and for the help with solving various obstacles we faced.

References

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information 10th*

Anniversary Edition. Cambridge University Press, 2010.

[2] Dekel Zak, Jingyi Mei, Jean-Marie Lagniez, and Alfons Laarman. Reducing quantum circuit synthesis to #sat. In Maria Garcia de la Banda, editor, *31st International Conference on Principles and Practice of Constraint Programming, CP 2025, Glasgow, Scotland, August 10-15, 2025*, volume 340 of *LIPIcs*, pages 38:1–38:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[3] Adam Paetznick and Krysta M. Svore. Repeat-until-success: non-deterministic decomposition of single-qubit unitaries. *Quantum Inf. Comput.*, 14(15-16):1277–1301, 2014.