Synthesis of Approximate Parametric Circuits for Variational Quantum Algorithms

Blake Burgstahler & Frank Mueller
Departments of Operations Research & Computer Science
North Carolina State University
Raleigh, USA
bburgst@ncsu.edu & fmuelle@ncsu.edu

Scott Pakin
Computer, Computational, and Statistical Sciences Division
Los Alamos National Laboratory
Los Alamos, USA
pakin@lanl.gov

***Abstract*—This work develops a novel approach to exploit synthesized, approximate circuits for the ansatz of variational quantum algorithms (VQA) and demonstrates its effectiveness for NchooseK, a domain-specific language supporting quantum-based solving of constraint-based problems. Synthesis is generalized to produce *parametric* circuits of short depth in close approximation of the original circuit *offline*. This removes synthesis from the critical path (online) between repeated quantum circuit executions of VQA while reducing circuit depth, thereby resulting in higher fidelity results than the baseline without synthesis. Simulation experiments indicate improvements of 98% on average. Further, experiments indicate that this approach can obtain viable solutions when the baseline could not. All of this is achieved with an average variation in circuit depth of less than 10%.**

***Index Terms*—circuit-model quantum computing, quantum annealing, programming models**

I. Introduction

Variational quantum algorithms (VQA) [3] are promising to solve relevant quantum chemistry and optimization algorithms [2] on contemporary noisy intermediate-scale quantum (NISQ) devices. However, current noise sources limit such algorithms to short-depth circuits, i.e. small problems. Meanwhile, circuit synthesis approaches provide opportunities for finding shorter circuit approximations, at the expense of searches. The iteration nature of VQA requires the ansatz to use different single qubit rotational angles in each iteration obtained from classical gradient optimization executing between repeated quantum circuit jobs. Adding online synthesis at each iteration would add significantly increased overhead to the critical path of any VQA framework.

This work develops a novel approach to address current shortcomings and opportunities for exploiting commonalities between iterative ansatz patterns. By exploiting synthesis, a latent parametric structure of the ansatz is sought. This parametric ansatz is instantiated at runtime with low cost overhead between VQA iterations, effectively taking synthesis out of the runtime loop. This work demonstrates the feasibility of such an approach by offline synthesis of approximate and parametric circuits for the ansatz of VQA. It then develops an integral workflow for inclusion in a domain-specific quantum framework, NchooseK, to solve constraint-satisfaction problems via QAOA.

This research was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20210397ER and is released under LA-UR-23-27212. Los Alamos National Laboratory is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy (contract no. 89233218CNA000001). This work was also supported in part by LANL subcontract 725530 and by NSF awards DMR-1747426, PHY-1818914, MPS-2120757, and CCF-2217020.

II. Design

We begin with a parameterized circuit with parameters $\beta \in \mathbb{R}^p$, where p is the number of parameters. We then repeat the following: select parameters to be assigned and apply a synthesis/instantiation workflow to this circuit to produce a new parameterized circuit with parameters $\alpha \in \mathbb{R}^q$ assigned.[1] This process is repeated in order to generate a set of representative synthesized circuits of different structures, which can heuristically be sampled as a replacement to the original circuit favoring low Hilbert-Schmidt (HS) distance, repeated structures, and shallow circuits.

III. Implementation

We leverage existing Python-based tools to implement the previously mentioned algorithms. First, NchooseK [7] is used to generate problem-specific circuits and as a baseline to compare results against for correctness. We have added capabilities to NchooseK in order to map the underlying QUBO to a parameterized quantum circuit.

Then, the Qiskit [5] development kit is used for execution of these circuits on simulators and hardware. Additionally, Qiskit provides a wrapper to the SciPy Python package for many classical optimization approaches, as well as many of its own implementations of optimizers.

Finally, the Berkeley Quantum Synthesis Toolkit (BQSKit) [8] is a Python framework that we leverage as it was specifically developed to efficiently tackle both synthesis and instantiation problems and provides methods for converting to and from Qiskit-based circuits.

[1]Note: p and q are allowed to be different as each circuit may require a different number of parameters, and even if they require the same number of parameters, they may not have identical gate placements/structures.

This generated circuit, together with the QUBO-based objective function encoded in the problem Hamiltonian, serves as inputs to all approaches.

IV. Framework

With our implementation set, we compare each of our novel approaches to typical QAOA results. This is realized using a subset of the benchmarks used in [7], specifically as set of max-cut and min-vertex cover problems. We search for generated circuits that return the solutions to these graph-based problems with high probability using our novel approaches. In both cases, the vertices of the graph directly correlate to a qubit in the generated graph and the edge constraints become two qubit interactions within the circuit. Using this framework, we generate problems ranging from 3 to 21 qubits and compare the solution sets generated by our algorithms to the solutions generated by Microsoft's classical Z3 theorem prover [6]. We compare the most likely solutions generated by our circuits to the solution provided by Z3 in two steps:

1) Is the candidate solution feasible?
2) Is the candidate solution "as good as" the Z3 solution?

The former is confirmed by validating that all hard constraints within the NchooseK environment are satisfied, while the latter is confirmed by ensuring the same number of soft constraints is satisfied. A solution is considered optimal moving forward if both conditions above are satisfied. In this way, we never have to solve the underlying NP-hard QUBO created by NchooseK and rather verify generated solutions in linear time by comparing to constraint. Further, this verification framework can be applied to noise-free, noisy, and hardware results.

V. Results

While our parameterized synthesis based approaches rarely improve the depth of the circuit when compared to transpiling original QAOA problems to CNOT and U3 gates, we do see that not only we can increase the probability of most likely solutions being optimal solutions, we also see stronger resilience against noise as compared to the original algorithm.

At worst, our methods can double the depth, which greatly increases the average depth. But at best, our methods produce circuits which reduce path by 10%. Note that instantiation becomes intractably slow at 9 qubits due to scaling of the number of parameters.

We see that synthesis greatly increases the probability of having optimal solutions among the top 5 most probable ones (henceforth referred to as "top 5"). We also observe this trend in the evaluation of the resulting circuits on a noisy simulator, seen in Fig. 1.

We believe that the absence of correct solutions at and above 12 qubits stems from two main sources. First, at this scale we have introduced enough partitions amid synthesis it is possible that the compounding errors from each block are becoming significant once we have a sufficient number of qubits. [1], [4] benchmark their synthesis approaches with some QAOA circuits, but neither use a QAOA circuit larger than 10 qubits, so there is no existing evidence to suggest that these issues are limited to us. Second, since our approaches seeks to improve upon QAOA results by synthesizing approximations, it is reasonable to expect that if QAOA does not yield valid solutions nor will approximations of it especially when using a local search optimizer. In other words, when QAOA is close to the correct solutions with a single iteration, we can improve upon it even further.

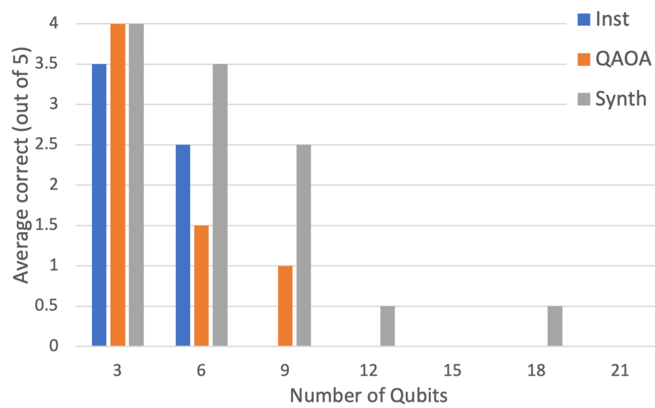

Fig. 1. Average Number of 'Top 5' peaks which are valid solutions when result circuit is run on a **noisy** simulator. Instantiation results beyond 6 qubits omitted.

VI. Conclusions

We introduced and surveyed the current state of VQA's and presented two synthesis-based algorithms for producing quantum circuits that increase the probability of finding optimal solutions. We showed that these algorithms can be applied in the specific context of QAOA and confirmed their viability over traditional QAOA.

Generally, if QAOA found a solution, our algorithms would increase the likelihood of also observing more optimal solutions, in some cases quadrupling this likelihood as compared to traditional QAOA. While our approaches do not significantly improve on the depth of circuits, they do consistently outperform the current standards.

References

[1] Marc G. Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. Towards optimal topology aware quantum circuit synthesis, 2020.

[2] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[3] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503, 2018.

[4] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. Quest: Systematically approximating quantum circuits for higher output fidelity, 2022.

[5] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

[6] Microsoft Research. Z3: an efficient theorem prover, 2023.

[7] Ellis Wilson, Frank Mueller, and Scott Pakin. "combining hard and soft constraints in quantum constraint-satisfaction systems", November 2022.

[8] Ed Younis, Costin C Iancu, Wim Lavrijsen, Marc Davis, Ethan Smith, and USDOE. Berkeley quantum synthesis toolkit (bqskit) v1, April 2021.