QAOA.jl: Toolkit for the Quantum and Mean-Field Approximate Optimization Algorithms

19 January 2023

Summary

Quantum algorithms are an area of intensive research thanks to their potential of speeding up certain specific tasks exponentially. However, for the time being, high error rates on the existing hardware realizations preclude the application of many algorithms that are based on the assumption of fault-tolerant quantum computation. On such *noisy intermediate-scale quantum* (NISQ) devices (Preskill 2018), the exploration of the potential of *heuristic* quantum algorithms has attracted much interest. A leading candidate for solving combinatorial optimization problems is the so-called *Quantum Approximate Optimization Algorithm* (QAOA) (Farhi, Goldstone, and Gutmann 2014).

`QAOA.jl` is a `Julia` package (Bezanson et al. 2017) that implements the *mean-field Approximate Optimization Algorithm* (mean-field AOA) (Misra-Spieldenner et al. 2023) - a quantum-inspired classical algorithm derived from the QAOA via the mean-field approximation. This novel algorithm is useful in assisting the search for quantum advantage by providing a tool to discriminate (combinatorial) optimization problems that can be solved classically from those that cannot. Note that `QAOA.jl` has already been used during the research leading to (Misra-Spieldenner et al. 2023).

Additionally, `QAOA.jl` also implements the QAOA efficiently to support the extensive classical simulations typically required in research on the topic. The corresponding parameterized circuits are based on `Yao.jl` (Luo et al. 2020), (Luo et al. 2023) and `Zygote.jl` (Innes et al. 2019), (Innes et al. 2023), making it both fast and automatically differentiable, thus enabling gradient-based optimization. A number of common optimization problems such as MaxCut, the minimum vertex-cover problem, the Sherrington-Kirkpatrick model, and the partition problem are pre-implemented to facilitate scientific benchmarking.

Statement of need

The demonstration of quantum advantage for a real-world problem is yet outstanding. Identifying such a problem and performing the actual demonstration on existing hardware will not be possible without intensive (classical) simulations. `QAOA.jl` facilitates this exploration by offering a classical baseline through the mean-field AOA, complemented by a fast and versatile implementation of the QAOA. As shown in our benchmarks, QAOA simulations performed with `QAOA.jl` are significantly faster than those of `PennyLane` (Bergholm et al. 2018), one of its main competitors in automatically differentiable QAOA implementations. While Tensorflow Quantum (Broughton et al. 2023) supports automatic differentiation, there exists, to the authors's knowledge, no dedicated implementation of the QAOA. The class `QAOA` offered by Qiskit (A-tA-v et al. 2021) must be *provided* with a precomputed gradient operator, i.e. it does not feature automatic differentiation out of the box.

Acknowledgements

The authors acknowledge partial support from the German Federal Ministry of Education and Research, under the funding program "Quantum technologies - from basic research to the market", Contract Numbers 13N15688 (DAQC), 13N15584 (Q(AI)2) and from the German Federal Ministry of Economics and Climate Protection under contract number, 01MQ22001B (Quasim).

References

A-tA-v, Sajid Anis, Abby-Mitchell, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, et al. 2021. "Qiskit: An Open-Source Framework for Quantum Computing." https://doi.org/10.5281/zenodo.2573505.

Bergholm, Ville, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, et al. 2018. "PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations." arXiv. https://doi.org/10.48550/arxiv.1811.04968.

Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. "Julia: A Fresh Approach to Numerical Computing." *SIAM Review* 59 (1): 65–98. https://doi.org/10.1137/141000671.

Broughton, Michael, Guillaume Verdon, Trevor McCourt, Antonio J. Martinez, Jae Hyeon Yoo, Sergei V. Isakov, Philip Massey, et al. 2023. "Tensorflow Quantum." *GitHub Repository*. GitHub. https://github.com/tensorflow/quantum.

Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann. 2014. "A Quantum Approximate Optimization Algorithm." *ArXiv e-Prints*.

Innes, Mike, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B Shah, and Will Tebbutt. 2019. "A Differentiable Programming System to Bridge Machine Learning and Scientific Computing." arXiv. https:

//doi.org/10.48550/arxiv.1907.07587.

———. 2023. "Zygote.jl." *GitHub Repository*. GitHub. https://github.com/FluxML/Zygote.jl.

Luo, Xiu-Zhe, Jin-Guo Liu, Pan Zhang, and Lei Wang. 2020. "Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design." *Quantum* 4 (October): 341. https://doi.org/10.22331/q-2020-10-11-341.

———. 2023. "Yao.jl." *GitHub Repository*. GitHub. https://github.com/QuantumBFS/Yao.jl.

Misra-Spieldenner, Aditi, Tim Bode, Peter K. Schuhmacher, Tobias Stollenwerk, Dmitry Bagrets, and Frank K. Wilhelm. 2023. "Mean-Field Approximate Optimization Algorithm." arXiv. https://doi.org/10.48550/ARXIV.2303.00329.

Preskill, John. 2018. "Quantum Computing in the NISQ Era and Beyond." *Quantum* 2 (August): 79. https://doi.org/10.22331/q-2018-08-06-79.