title: 'QAOA.jl: Toolkit for the Quantum and Mean-Field Approximate Optimization Algorithms' tags: - Julia - quantum algorithms - automatic differentiation - optimization authors: - name: Tim Bode orcid: 0000-0001-8280-3891 corresponding: true affiliation: 1 - name: Dmitry Bagrets affiliation: "1, 2" - name: Aditi Misra-Spieldenner affiliation: 3 - name: Tobias Stollenwerk affiliation: 1 - name: Frank K. Wilhelm affiliation: "1, 3"
affiliations: - name: Institute for Quantum Computing Analytics (PGI-12), Forschungszentrum Jülich, 52425 Jülich, Germany index: 1 - name: Institute for Theoretical Physics, University of Cologne, 50937 Cologne, Germany index: 2 - name: Theoretical Physics, Saarland University, 66123 Saarbrücken, Germany index: 3 date: 19 January 2023 bibliography: paper.bib

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 QAOA to enable the efficient classical simulation typically required in research on the topic. It is based on `Yao.jl` (Luo et al. 2019), (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.

Additionally, `QAOA.jl` is the first package to implement the *mean-field Approximate Optimization Algorithm* (mean-field AOA) (Misra-Spieldenner et al. 2023), which is a quantum-inspired classical algorithm derived from the QAOA via the mean-field approximation. Note that `QAOA.jl` has already been used extensively during the research leading to (Misra-Spieldenner et al. 2023). This novel algorithm can be useful in assisting the search for quantum advantage since it provides a tool to discriminate (combinatorial) optimazation problems that be solved classically from those that cannot.

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. This makes a fast and versatile implementation of the QAOA rather desirable. As shown in Figure 1, `QAOA.jl` is significantly faster than `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 author'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.

As already mentioned, `QAOA.jl` is also the first package to implement the mean-field AOA, which is thus made available to researchers working on the topic.

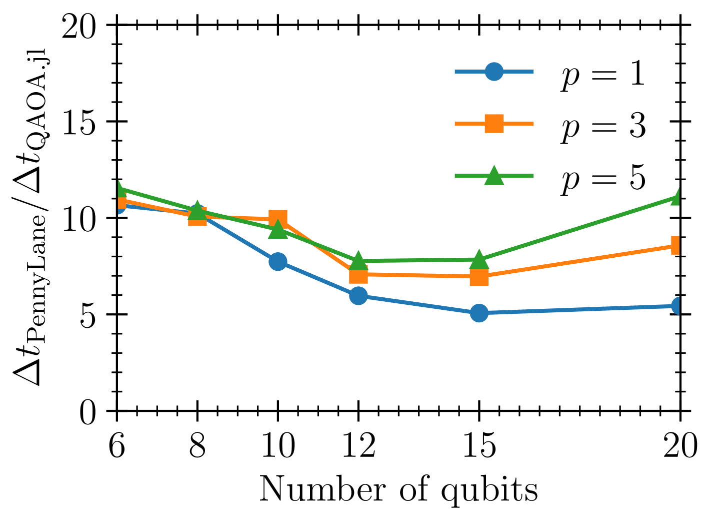

Figure 1: Comparison of run times between `PennyLane` (Bergholm et al. 2018) and `QAOA.jl` on an Apple M1 processor. The benchmarks Δt are retrieved by performing 128 steps with the respective gradient optimizer on the same instance of size N of the minimum vertex-cover problem.

Mathematics

QAOA

The cost function of the QAOA for a general quadratic optimization problem is typically defined as

$$\hat{C} = \sum_{i=1}^{N} \left[h_i + \sum_{j>i} J_{ij} \hat{Z}_j \right] \hat{Z}_i,$$

where the h_i, J_{ij} are real numbers encoding the problem in question, and $\hat{Z}_{i,j}$ denote Pauli matrices. Similarly, the conventional *mixer* or *driver* of the QAOA is given by

$$\hat{D} = \sum_{i=1}^{N} \hat{X}_i,$$

where the $\hat{X}_i$ are again Pauli matrices. We also introduce the initial quantum state

$$|\psi_0\rangle = |+\rangle_1 \otimes \cdots \otimes |+\rangle_N.$$

Note that this is the maximum-energy eigenstate of the driver $\hat{D}$ since $\langle\psi_0|\hat{D}|\psi_0\rangle = N$. With these prerequisites, the variational quantum state of the QAOA becomes

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \exp\left(-\mathrm{i}\beta_p \hat{D}\right) \exp\left(-\mathrm{i}\gamma_p \hat{C}\right) \cdots \exp\left(-\mathrm{i}\beta_1 \hat{D}\right) \exp\left(-\mathrm{i}\gamma_1 \hat{C}\right) |\psi_0\rangle.$$

The goal is then to *maximize* the expectation value

$$E_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})|\hat{C}|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$$

over the variational parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}$. Note that `QAOA.jl` furthermore supports others drivers, e.g.

$$\hat{D} = \sum_{(i,j)\in\mathcal{E}} \left(\hat{X}_i \hat{X}_j + \hat{Y}_i \hat{Y}_j \right),$$

where $\mathcal{E}$ is the set of connections or *edges* for which the coupling matrix J_{ij} is non-zero.

Mean-Field AOA

In close analogy to the QAOA, the mean-field Hamiltonian reads

$$H(t) = \gamma(t) \sum_{i=1}^{N} \left[h_i + \sum_{j>i} J_{ij} n_j^z(t) \right] n_i^z(t) + \beta(t) \sum_{i=1}^{N} n_i^x(t).$$

The mean-field evolution is then given by

$$\boldsymbol{n}_i(p) = \prod_{k=1}^{p} \hat{V}_i^D(k) \hat{V}_i^P(k) \boldsymbol{n}_i(0),$$

where the initial spin vectors are $\boldsymbol{n}_i(0) = (1,0,0)^T$ for all $i = 1, ..., N$, and the rotation matrices $\hat{V}_i^{D,\,P}$ are defined as

$$\hat{V}_i^D(k) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(2\Delta_i\beta_k) & -\sin(2\Delta_i\beta_k) \\ 0 & \sin(2\Delta_i\beta_k) & \cos(2\Delta_i\beta_k) \end{pmatrix}$$

and

$$\hat{V}_i^P(k) = \begin{pmatrix} \cos(2m_i(t_{k-1})\gamma_k) & -\sin(2m_i(t_{k-1})\gamma_k) & 0 \\ \sin(2m_i(t_{k-1})\gamma_k) & \cos(2m_i(t_{k-1})\gamma_k) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

with the magnetization

$$m_i(t) = h_i + \sum_{j=1}^{N} J_{ij} n_j^z(t).$$

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. 2019. "Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design." *ArXiv e-Prints*.

———. 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.