

Supplementary Material: Effects of Quantum Noise on Quantum Approximate Optimization Algorithm

Cheng Xue,^{1,*} Zhao-Yun Chen,^{1,2} Yu-Chun Wu,¹ and Guo-Ping Guo^{1,†}

¹Key Laboratory of Quantum Information, Chinese Academy of Sciences, School of Physics, University of Science and Technology of China, Hefei, Anhui, 230026, P. R. China

²Origin Quantum Computing Hefei, Anhui 230026, P. R. China

I. QAOA

The QAOA was developed by Farhi, Goldstone, and Gutmann[1]. It gives a way to solve approximately in polynomial time combinatorial optimization problems such as the Max-Cut problem, the traveling salesman problem, and the 3-SAT problem .

The QAOA can be regarded as a simplified version of an adiabatic evolution, which involves the continuous evolution of a quantum state from a ground state of a Hamiltonian to a ground state of another Hamiltonian. As the QAOA is discrete, it drives the ground state of one Hamiltonian to the ground state of another Hamiltonian in discrete steps. In the respect, the QAOA is an approximation algorithm. In general, the output state of the QAOA is just close to the ground state of the target Hamiltonian. The QAOA quantum circuit has a parameterization. Changing the parameters of the circuit adjusts the direction of each step of the QAOA. In particular, the QAOA quantum circuit can be written

$$U(\vec{\gamma}, \vec{\beta}) = \prod_{j=1}^n e^{-iH_d\beta_j} e^{-iH_p\gamma_j}, \quad (1)$$

where n denotes the QAOA step number, and H_p the problem Hamiltonian. For a specific problem, the ground state of H_p is the solution of the problem. H_d is the drive Hamiltonian of the QAOA, the initial state of the QAOA being the ground state of H_d . Generally, we define H_d in the form

$$H_d = \sum_i -X_i, \quad (2)$$

where X represents the Pauli X matrix. The initial state of the QAOA is then $|\phi_{in}\rangle = |+\rangle^{\otimes m}$, m being the qubit number of the QAOA quantum circuit, and $\vec{\gamma}$ and $\vec{\beta}$ the two $[n, 1]$ parameter vectors. The purpose of the QAOA is to find the optimal values of $\vec{\gamma}$ and $\vec{\beta}$ that ensures the output quantum state is closest to the ground state of the problem Hamiltonian H_p . Generally, we use a classical optimization algorithm to find the optimal $\vec{\gamma}$ and $\vec{\beta}$. The cost function is the expectation of H_p and may be written

$$f(\vec{\gamma}, \vec{\beta}) = \langle \phi_{in} | U^\dagger(\vec{\gamma}, \vec{\beta}) H_p U(\vec{\gamma}, \vec{\beta}) | \phi_{in} \rangle. \quad (3)$$

When solving specific problems, the whole execution of the QAOA comprises five steps:

- (1) Construct problem Hamiltonian H_p . The ground state of H_p is the solution of the problem.
- (2) Use H_p, H_d to construct the QAOA quantum circuit—the initial values of $\vec{\gamma}$ and $\vec{\beta}$ may be chosen randomly.
- (3) Execute QAOA quantum circuit multiple times and compute the cost function $f(\vec{\gamma}, \vec{\beta}) = \langle H_p \rangle$.
- (4) Optimize $\vec{\gamma}$ and $\vec{\beta}$ to minimize the cost function $f(\vec{\gamma}, \vec{\beta})$ with the classical optimization algorithm.
- (5) Execute the optimized QAOA quantum circuits multiple times—the optimal result is the solution of the problem.

*Electronic address: itachi@mail.ustc.edu.cn

†Electronic address: gpguo@ustc.edu.cn

II. SIMULATION METHOD

In our numerical simulations, we use a noisy quantum virtual machine (QVM) in the quantum programming architecture for the NISQ-device application (QPanda) as our noisy quantum simulator[2]. QPanda’s noisy QVM supports all kinds of quantum noise models. Users can also define their own quantum noise model. QPanda also contains the variational quantum network (VQNet)—a framework to construct quantum-classical hybrid neural networks[3]. We use VQNet to realize the QAOA.

A. Noisy Simulator

First, we introduce the method of QPanda’s noisy QVM. In noisy quantum circuits, the quantum state is a mixed state and represented by a density matrix. In QPanda’s noisy QVM, we still use the state vector to represent the quantum state and combine the Monte Carlo method to simulate mixed-state evolution under noisy quantum operations[4]. The specific method is as follows:

- (1) Assume the input state is a pure state denoted by $|\phi\rangle$ and U a unitary quantum gate; the quantum noise is represented by Kraus operators $K = \{K_i\}, i = 1, 2, \dots, s$.
- (2) Compute $p_i = \langle\phi|K_i^\dagger K_i|\phi\rangle, i = 1, 2, \dots, s$. We note that $\sum_i p_i = 1$.
- (3) Generate a uniformly distributed random number r in the range $[0, 1)$, then find l that satisfies

$$\sum_{i=1}^{l-1} p_i \leq r \leq \sum_{i=1}^l p_i. \quad (4)$$

Here we assume $\sum_{i=1}^0 p_i = 0$.

- (4) The expression for the evolution of state $|\phi\rangle$ is

$$|\phi_l\rangle \rightarrow U \frac{1}{\sqrt{p_l}} K_l |\phi\rangle. \quad (5)$$

- (5) Repeat procedure (2) ~ (4) M times. M_l is the number of times that K_l was selected. When M is large enough, we know that $M_l \approx Mp_l$, and hence the output quantum state ρ becomes

$$\rho \approx \sum_i \frac{M_l}{M} |\phi_l\rangle\langle\phi_l|. \quad (6)$$

The noisy QVM realized by this method needs to run the quantum circuit multiple times. The density matrix of the output state may then be computed approximately from the statistical distribution of the results of multiple runs.

B. VQNet

We use VQNet to realize QAOA. VQNet supports ideal and noisy QVM. The flow chart of QAOA is shown in Fig. S1. $\vec{\gamma}$ and $\vec{\beta}$ are QAOA quantum circuit parameters, “Variational Quantum Circuit” represents the QAOA quantum circuit, “Quantum Operator” is a basic operator in VQNet, and the inputs of “Quantum Operator” are the “Variational Quantum Circuit” and the “Graph Hamiltonian”. The output is the expectation of the input Hamiltonian after running the “Variational Quantum Circuit”. The VQNet supports back propagation to get the derivatives of the cost function with respect to $\vec{\gamma}$ and $\vec{\beta}$. We then use the gradient descent algorithm to optimize parameters $\vec{\gamma}$ and $\vec{\beta}$.

-
- [1] E. Farhi, J. Goldstone, and S. Gutmann (2014), arXiv: 1411.4028, URL <http://arxiv.org/abs/1411.4028>.
 - [2] Z.-Y. Chen and G.-P. Guo (2019), arXiv: 1901.08340, URL <http://arxiv.org/abs/1901.08340>.
 - [3] Z.-Y. Chen, C. Xue, S.-M. Chen, and G.-P. Guo (2019), arXiv: 1901.09133, URL <http://arxiv.org/abs/1901.09133>.
 - [4] *Noise and Quantum Computation — pyQuil 2.8.0 documentation*, URL <https://pyquil.readthedocs.io/en/latest/noise.html>.

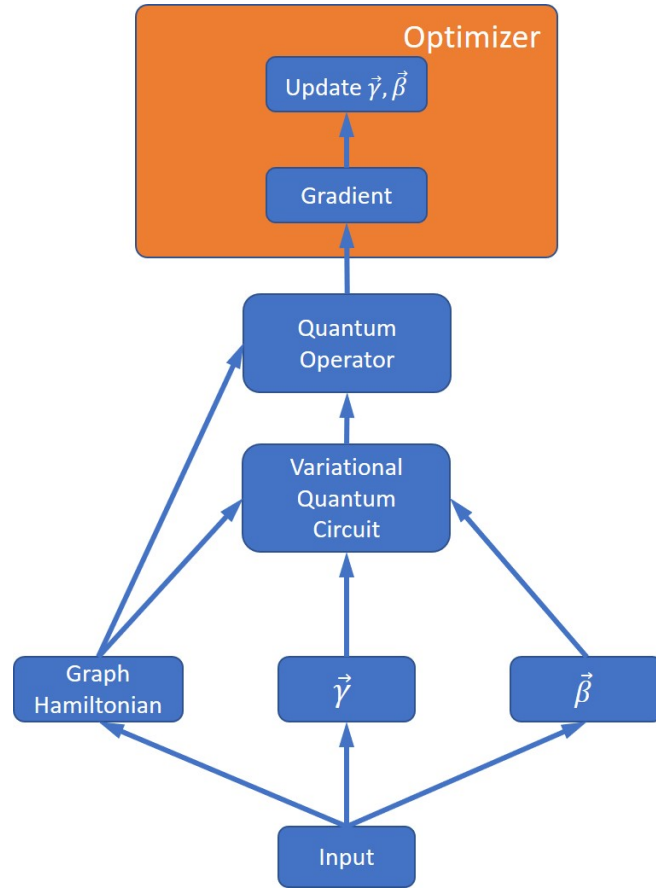


FIG. S1: QAOA Flow Chart