A loop Quantum Approximate Optimization Algorithm with Hamiltonian updating

2 authors, including:

Zhang Danbo
South China Normal University

31 PUBLICATIONS   **54** CITATIONS

Some of the authors of this publication are also working on these related projects:

Hybrid quantum-classical algorithm View project

Quantum-data science View project

A loop Quantum Approximate Optimization Algorithm with Hamiltonian updating

Fang-Gang Duan[1] and Dan-Bo Zhang[1, 2, *]

[1] *Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials,*
School of Physics and Telecommunication Engineering,
South China Normal University, Guangzhou 510006, China
[2] *Guangdong-Hong Kong Joint Laboratory of Quantum Matter,*
Frontier Research Institute for Physics, South China Normal University, Guangzhou 510006, China
(Dated: September 27, 2021)

Designing noisy-resilience quantum algorithms is indispensable for practical applications on Noisy Intermediate-Scale Quantum (NISQ) devices. Here we propose a quantum approximate optimization algorithm (QAOA) with a very shallow circuit, called loop-QAOA, to avoid issues of noises at intermediate depths, while still can be able to exploit the power of quantum computing. The key point is to use outputs of shallow-circuit QAOA as a bias to update the problem Hamiltonian that encodes the solution as the ground state. By iterating a loop between updating the problem Hamiltonian and optimizing the parameterized quantum circuit, the loop-QAOA can gradually transform the problem Hamiltonian to one easy for solving. We demonstrate the loop-QAOA on Max-Cut problems both with and without noises. Compared with the conventional QAOA whose performance will decrease due to noises, the performance of the loop-QAOA can still get better with an increase in the number of loops. The insight of exploiting outputs from shallow circuits as bias may be applied for other quantum algorithms.

I. INTRODUCTION

Recent advances on quantum computing technology advocate a growing interest in finding quantum algorithms that can be implemented effectively on NISQ devices [1]. One such strong candidate is variational quantum algorithm (VQA) [2–6], which leverages up the expressive power of parameterized quantum circuits and a hybrid quantum-classical optimization procedure for finding solutions. Variational quantum algorithms have wide applications for solving quantum many-body systems [7, 8], quantum chemistry [9–13], optimization problems [14–17], and so on. Among them, the quantum approximate optimization algorithm (QAOA) stands out as a VQA for tackling hard combinatorial optimization problems [18–26].

The QAOA applies an alternating evolution of mixer and problem Hamiltonians with optimized evolution periods that its outputs contain the target solution with a large weighting. By exploiting quantum effects such as interference, QAOA is believed to show quantum advantages on combinatorial optimization problems at intermediate depths [15, 27]. One center challenge for implementing QAOA, however, is that noises on near-term quantum processors will be detrimental to the performance. For instance, the recent research of the Google team shows that QAOA performs worse at depths larger than $p = 3$ [16]. Moreover, the non-convex and high-dimensional parameter landscape caused by large p may bring huge difficulty for optimizing [28, 29]. Without efficient approaches for optimization, the hybrid algorithms will lose it's potential advatages [30].

Those challenges addressed at intermediate depths of QAOA have being tackled recently, e.g., the issue of noise can be reduced by error mitigation techniques [31, 32], and some constructive optimization procedures have been proposed [15, 33]. An alternative approach, however, is to ask whether we can exploit very shallow QAOA to achieve good performance under noise. As the circuit depth is very shallow, there is no difficulty of variational parameter optimization, and effects of noises can also be small. Remarkably, even the lowest depth ($p = 1$) of QAOA has been proved that classical computers can't simulate efficiently [27]. This suggests a direction for designing noise-resilience QAOA using shallow circuits while the quantum computing power may still be exploited.

Our approach is to explore both the parameterized quantum circuit and the problem Hamiltonian in QAOA for finding the target solution. In the conventional QAOA, the Hamiltonian which encodes the problem is usually fixed. However, it is possible that many different Hamiltonians can have the same solutions or approximated solutions but may be solved with QAOA at different degrees of difficulty. Motivated by this, we may first use a shallow QAOA to get some rough estimations of the solution, and then update the problem Hamiltonian with biased terms. By iterating the procedure of updating the problem Hamiltonian and finding the optimized parameters of the quantum circuit, there is a feedback loop between those two subroutines and the solution may get better with an increase of loops. In this regard, we may call it as *loop-QAOA*.

In this work, we propose a kind of noise-resilience quantum approximate optimization algorithm, the loop-QAOA, for solving combinatorial optimization problems with very shallow circuits, with a feedback loop between Hamiltonian updating and parameter optimization. Dif-

ferent strategies for constructing bias field terms are utilized in the Hamiltonian updating. We test the loop-QAOA and compare it to conventional QAOA on typical Max-Cut graph instances up to 12-vertex instances, simulated both with and without noise. The results suggest that loop-QAOA can be a practical quantum algorithm on noisy quantum devices, where the conventional QAOA may fail to improve its performance by increasing the circuit depth. Our work gives a new sight for designing hybrid quantum-classical algorithms using feedback from shallow quantum circuits.

The paper is organized as follows: we firstly review QAOA and the Max-Cut problem in Sec. II. Then we will propose loop-QAOA and some strategies for constructing bias field terms in Sec. III. The simulated results are shown in Sec. IV. Finally, conclusions are given in Sec. V.

II. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM REVISITED

The Quantum Approximate Optimization Algorithm (QAOA) by Farhi et al. is a hybrid quantum-classical algorithm, designed to produce approximated solutions for combinatorial optimization problems. The solutions are encoded as the ground state of H_P, the problem Hamiltonian or the phase Hamiltonian. The H_P is diagonal in the computational basis. As various hard combinatorial optimization problems can be encoded in such diagonal Hamiltonians [34], the QAOA is often considered as a potential method of solving combinatorial optimization problems on quantum computers. These problems are usually defined on N-bit binary strings $\boldsymbol{z} = z_1 z_2 \cdots z_N$. The goal is to find a string which maximizes the classical objective functions $C(\boldsymbol{z}) : \{+1, -1\}^{\otimes N} \mapsto \mathbb{R}$.

As the first step of the QAOA, the classical objective function is encoded in a quantum Hamiltonian H_P defined on N-qubits by replacing each variable z_i with a single-qubit σ_i^z. The ground state of H_P is parametrized with a quantum circuit which is constructed as follows. For a p-layer QAOA, the state is initialized as $|+\rangle^{\otimes N}$, where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. Let $(\vec{\boldsymbol{\beta}}, \vec{\boldsymbol{\gamma}}) = (\beta_1, \cdots \beta_p, \gamma_1 \cdots, \gamma_p)$ be real positive numbers for the layer p. The phase Hamiltonian H_P and a mixer Hamiltonian, often chosen as $H_M = \sum_i \sigma_i^x$, are then applied alternately to generate a variational wavefunction

$$|\Psi_p(\vec{\boldsymbol{\beta}}, \vec{\boldsymbol{\gamma}})\rangle = \prod_{j=1}^{p} e^{-i\beta_j H_M} e^{-i\gamma_j H_P} |+\rangle^{\otimes N}. \quad (1)$$

The expectation value H_P with regard to the variational state is

$$F_p(\vec{\gamma}, \vec{\boldsymbol{\beta}}) = \langle \Psi_p(\vec{\boldsymbol{\beta}}, \vec{\gamma})| H_P |\Psi_p(\vec{\boldsymbol{\beta}}, \vec{\gamma})\rangle, \quad (2)$$

which can be obtained by measurements in the computational basis. The goal is to make $F_p(\vec{\gamma}, \vec{\boldsymbol{\beta}})$ minimal. The parameters $\beta_1, \cdots \beta_p, \gamma_1 \cdots, \gamma_p$ are adjusted in re-

peated learning loops in a classical computer to minimize the objective function Eq. (2) evaluated with a quantum computer.

We use the Max-Cut to illustrate the QAOA more concisely. Consider a graph G with N vertices over a vertex set V and edge set E. The Max-Cut problem is to divide the vertexes V into two disjoint subsets such that the number of cuts (edges connecting two disjoint subsets) is maximized. Let ω_{ij} be the weight of the associated edge $\langle i, j \rangle \in E$. The objective function can be defined as a sum of edge terms in the graph as

$$C(\boldsymbol{z}) = \sum_{\langle i,j \rangle \in E} C_{ij} = \frac{1}{2} \sum_{\langle i,j \rangle \in E} \omega_{ij}(1 - z_i z_j) \quad (3)$$

where $z_i = \pm 1$. By maximizing $C(\boldsymbol{z}$, the cut set $S(z) \in E$ can be obtained for the Max-Cut problem.

The Max-Cut Problem is in the APX-complete complexity class, meaning that there is no polynomial-time approximation scheme to find the optimal solution, unless P = NP [35–37]. But we may find a polynomial-time approximation algorithm with a solution of a string $\boldsymbol{z}'$ that achieves a desired approximation ratio

$$\frac{C(\boldsymbol{z}')}{\max\{C(\boldsymbol{z})\}} \geq r^*. \quad (4)$$

The objective function of the Max-Cut problem can be encoded as a problem Hamiltonian,

$$H_P = -\frac{1}{2} \sum_{\langle i,j \rangle \in E} \omega_{ij}(1 - \sigma_i^z \sigma_j^z). \quad (5)$$

Note an addition minus sign in H_P turns the task of maximizing the objective function $C(\boldsymbol{z})$ to finding the lowest energy of H_P. If spins σ_i^z and σ_j^z are anti-aligned, an edge $\langle i, j \rangle$ contributes with an energy $-\omega_{ij}$; otherwise, the energy contribution is zero.

It has been proven NP-hard to design an algorithm that guarantees an approximation ratio of $r^* \geq 16/17 \approx 0.941$ for Max-Cut on all graphs [35]. The best-known classical approximation algorithm, of Goemans and Williamson (GW) [38], guarantees an approximation ratio of $r^* \approx 0.878$ using semidefinite programming and randomized rounding. For more special graphs the GW can do better, r^* can reach 0.932 for u3R graphs [39]. Farhi et al. [14] showed that QAOA at level $p = 1$ the worst-case approximation ratio is 0.6924 for u3R graphs. It is known the performance of QAOA gets better with an increase of the depth p [14]. The goal of quantum approximate optimization is to raise r^* that is beyond the capacity of classical algorithms.

The QAOA often requires quantum circuits of intermediate depths to achieve the desired accuracy for combinatorial optimization problems. However, the benefit of increasing p in QAOA will be hampered due to noises in quantum processors. The performance in the presence of noises may get worse for p larger than a certain small

value [16]. Even without noise, optimization of variational parameters for large p can be a great challenge. The two issues can be avoided if a shallow circuit is used in QAOA, which would be explored in the next section.

III. LOOP-QAOA WITH PROBLEM HAMILTONIAN UPDATING

In QAOA, it is both the problem Hamiltonian and the parameterized quantum circuit to determine the solution, as seen in Eq. (2). Usually, the problem Hamiltonian is given beforehand, and what is left is to increase the depth p or design novel variational ansatz. Nevertheless, there is plenty of room for handling the problem Hamiltonian. Since the same solution may correspond to different problem Hamiltonians. It is possible that the problem Hamiltonian can be transformed to a new one with the same ground state but easier to solve. Instead of increasing the depth p, one can fix QAOA with a very small p, while the problem Hamiltonian is improved. Guided by this insight, we propose the loop-QAOA, where the problem Hamiltonian is updated in an iterative fashion with the feedback from shallow-circuit QAOA.

Let us first elaborate on the case of one loop to illustrate the mechanism behind loop-QAOA. Throughout the paper, we use the least shallow circuit in loop-QAOA, namely $p = 1$, while the case $p > 1$ can be investigated similarly. In practice, the $p = 1$ QAOA can show much better performance than randomly guessing. Moreover, it gives a probability distribution over all possible bitstrings. The output of $p = 1$ QAOA thus can reveal the underlying structure of the problem to some degree, which can be exploited as a bias for the combinatorial optimization problems. For instance, the bitstrings with high probabilities are informative for constructing bias terms for updating the problem Hamiltonian.

The procedure of training $p = 1$ QAOA and updating the problem Hamiltonian can be iterated multiple times. Schematic of the loop-QAOA can be found in Fig. 1a. The aim is to gradually transform the Hamiltonian that the chance to get the target solution will increase with more loops. To make this possible, the bias terms and the scheme of Hamiltonian updating should be designed properly. We note that using bias terms to accelerate the quantum optimization has been considered in the context of quantum annealing [40].

We illustrate the procedure of Hamiltonian updating with bias terms with the Max-Cut problem. Suppose we have a connected weighted graph $G = (V, E)$ with N nodes, and the weight set of its edges is $W = \{\omega_{ij} : \langle i, j \rangle \in E\}$. The output of $p = 1$ QAOA is a probability distribution $Q = \{(\boldsymbol{z}_i, p_i)\}$, where $\boldsymbol{z}_i$ is the bitstring and p_i is the corresponding probability. We will choose the results with p_i greater than a certain threshold p^*, which are informative for constructing bias terms.

The construction of bias terms is based on how the distribution of bitstrings output from QAOA contributes to the objective function. Consider a string $\boldsymbol{z}_1$ for in-
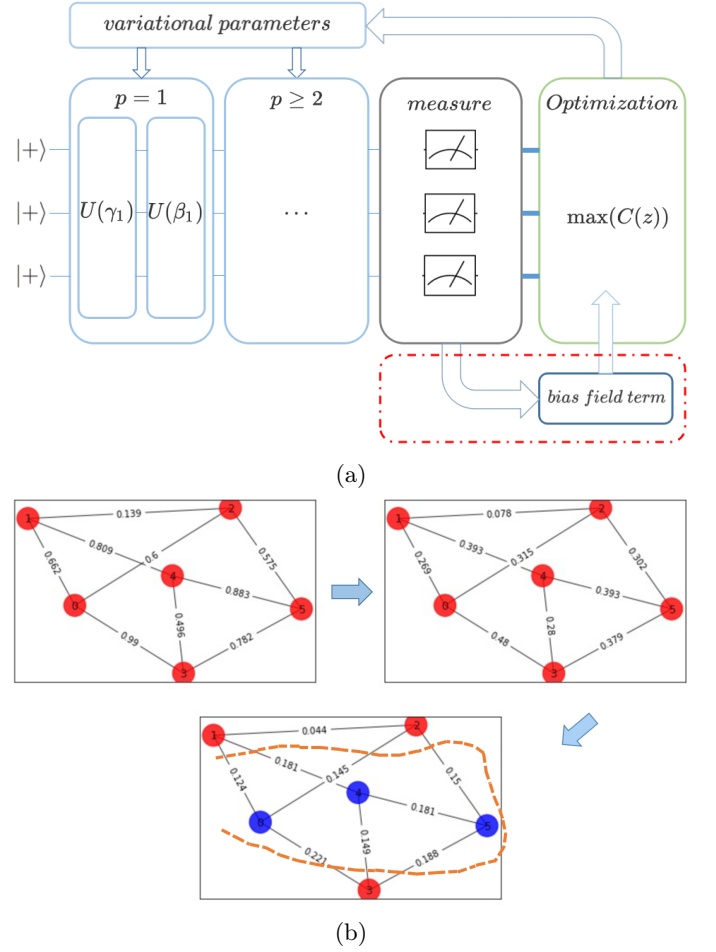

(a)

(b)

FIG. 1: **Schematic of a p-level QAOA and loop-QAOA strategy.** (a) The circuit diagram in the solid frame represents QAOA. A quantum circuit takes input $|+\rangle^{\otimes N}$ and alternately applies $U(\beta_i) = e^{-i\beta_i H_P}$ and $U(\gamma_i) = e^{-i\gamma_i H_M}$, and measure the final state to obtain expectation value with respect to the objective function $C(z)$. This result is optimized by a classical optimizer to get the maximum value. The dashed box represents the loop-QAOA strategy. We add bias field terms to objective Hamiltonian through the measurement results of each loop. (b) Schematic of loop-QAOA strategy applied to a 6-vertex w3R graph, where weights are updated at each loop.

stance. If two of the spins are opposite, $\boldsymbol{z}_{1i} = 0, \boldsymbol{z}_{1j} = 1$ or $\boldsymbol{z}_{1i} = 1, \boldsymbol{z}_{1j} = 0$ (based on the $\mathbb{Z}_2$ symmetry of the Max-Cut problem), then the $\langle i, j \rangle$ edge will make a contribution of ω_{ij} to the objective function. If the spins are the same, namely $\boldsymbol{z}_{1i} = \boldsymbol{z}_{1j}$, the contribution is 0. Considering the distribution from measurement results, $\langle i, j \rangle$ in z_1 makes a contribution of $p_{z_1}\omega_{ij}$. Then the contribution of the edge $\langle i, j \rangle$ to the objective function in total

is

$$C_{ij} = \sum_{k=1}^{N} p_{z_k} \omega_{ij} \qquad (6)$$

After going through all the edges in G, we get a set $\{C_{ij}\}$. We can reward the edge with a larger contribution and punish the edge with a smaller contribution.

The reward or punishment is realized by adding bias terms to the problem Hamiltonian. For this, we introduce a mapping $F : W \mapsto W'$, which tunes the weights of the graph. As a schematic, Fig. 1b shows how weights of a 6-nodes w3R graph are updated in the loop-QAOA, which finally becomes easy to solve. There can be different mapping schemes. Apparently, different mapping methods or different mapping parameters will lead to different results. Here, we list some bias field methods: (1). Bias field and antibias field: $W' = \{\omega_{ij} \pm kC_{ij}\}$ (k can be a constant or a function, for controling the bias field terms.); (2). Fixed bias field: $W' = \{\omega_{ij} + \text{fixed bias terms}\}$. We adopt a modified version of the first scheme with bias fields that updates edge weight in each loop via the following mapping,

$$\omega'_{ij} = (1 - \tau \sum_{z_{ki}=z_{kj}}^{N} p_{\boldsymbol{z}_k}) \omega_{ij}, \qquad (7)$$

where τ is a parameter to control the strength of bias.

Using this scheme, the solution space will be reduced by punishing a bitstring $\boldsymbol{z}_k$ if there are many pairs of spins that are the same $z_{ki} = z_{kj}$ on the edges. Note that the weights may approach zeroes for a large number of loops. To avoid this situation, we can control the bias strength τ to make the strategy get the desired results with a small number of loops. A heuristic formula of bias strength is proposed by interpolation:

$$\tau = \frac{1}{2^{n-2}}[1 - \frac{1}{3}(\frac{1}{g\Delta_{max}})^{\frac{1}{l+f(n)}} - \frac{2}{3}(\frac{1}{g\Delta_{min}})^{\frac{1}{l+f(n)}}] \quad (8)$$

where n is nodes number, l is the current number of loops, $f(n)$ is a function monotonous with the number of nodes, Δ_{max} is the maximum weight difference, and $\Delta_{min} > 0$ is the minimum weight difference. The factor g ensures that the results are within the given accuracy range, and we chose $g = 10^3$ to keep three significant digits. The power correspond to current loops of algorithm, where $f(n)$ is usually taken as a fixed constant. Δ_{max} and Δ_{min} ensure that all edges will not become too small within a certain number of loops. The coefficinet $1/3$ and $2/3$ are obtained by interpolating, and can be adjusted to other numbers. We mention that the above method is not only limited to the Max-Cut problem but also effective for general combinatorial optimization problems.
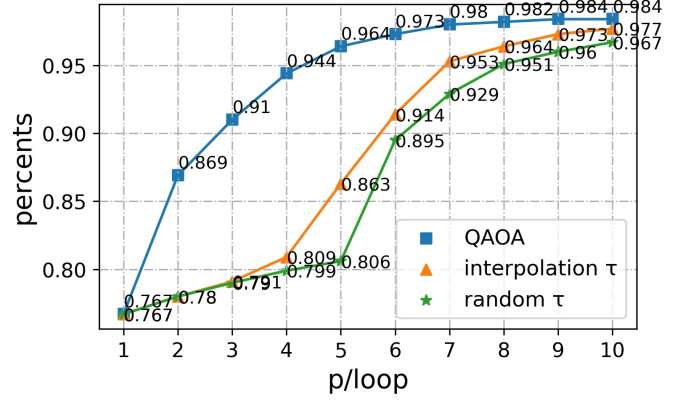
(a)
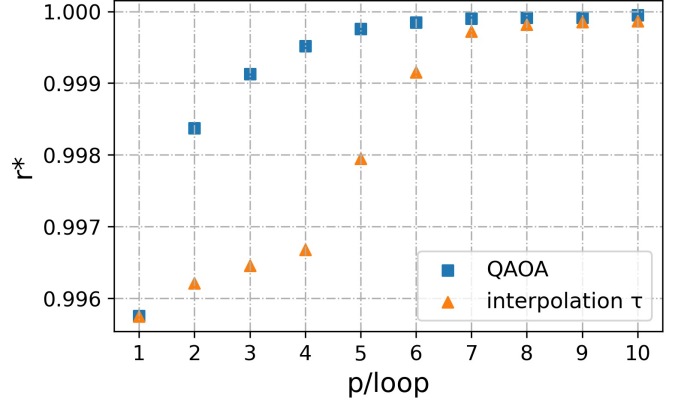
(b)

FIG. 2: **Performances of the conventional QAOA and the loop-QAOA with two different choices of bias strengths.** (a) the average success rates p_s on 10^4 randomly generated 12-vertex w3R graphs. τ indicates the bias strength, and they are chosen by the interpolation in Eq. (8), or randomly choosing form 2.0×10^{-4} to 2.8×10^{-4}. For all cases loop-QAOA gradually approaches the conventional QAOA with the increase of the number of loops. (b) The average approximation ratio r^* of the QAOA and the loop-QAOA with the interpolation method.

IV. SIMULATION RESULTS

In this section, we first give the simulation methods and then present the simulation results. Comparisons between the performances of loop-QAOA and the conventional QAOA are given, both in the presence or absence of noises.

A. Simulation methods

Without losing the difficulty of the combinatorial optimization problem, we are concerned about connected

graphs with weights (such as w3R) w_{ij} chosen uniformly at random from the interval $[0, 1]$. We consider randomly generated instances w3R graphs with vertex number $8 \leq N \leq 14$. For each vertex number, 10^4 w3R graph instances are generated. The loop-QAOA strategy at each loop and the conventional QAOA at each layer p are applied for those generated graphs. As for the optimization method, we apply both the gradient-based optimization routine (BFGS) in our numerical simulation. It is noted that non-gradient-based routines such as Nelder-Mead [41, 42] and Bayesian methods [43] also works for the loop-QAOA.

We also add noise in our numerical simulation to address that the loop-QAOA can be noise-resilience. We choose three quantum noise models to test the algorithm, namely quantum bit flip channel, quantum phase flip channel, and depolarizing channel [44, 45]. The quantum bit flip channel evolves a density matrix via:

$$\rho \rightarrow (1-q)\rho + qX\rho X, \quad (9)$$

where q is the probability of a bit flip (q instead of p is to distinguish it from the previous p-layer). The quantum phase channel evolves a density matrix via:

$$\rho \rightarrow M_0\rho M_0^\dagger + M_1\rho M_1^\dagger \quad (10)$$

with :

$$M_0 = \sqrt{q}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , M_1 = \sqrt{1-q}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad , \quad (11)$$

where q is the probability of a phase flip. The depolarizing channel applies one of 4^n disjoint possibilities: the identity channel or one of the $4^n - 1$ pauli gates. The probability of a non-identity Pauli gates are $q/(4^n - 1)$, and the identity is done with probability $1 - q$, where q is the probability that one of the Pauli gates is applied, n is the number of qubits, P_i are the $4^n - 1$ Pauli gates(excluding the identity). This channel evolves a density matrix via

$$\rho \rightarrow (1-q)\rho + \frac{q}{4^n - 1}\sum_i P_i\rho P_i \quad (12)$$

We apply these models to the whole quantum circuit. We show results for the case of $q = 0.1$ in bit flip channel and phase flip channel, $q = 0.05$.in depolarizing channel in Fig 3.

B. Results

We first compare the performances of the conventional QAOA and the loop-QAOA in the absence of noises. For the loop-QAOA, the bias strengths at different loops are chosen according to the interpolation in Eq. (8) or chosen randomly from a given interval. The results on a set of 12-vertex w3R graphs with 10^4 instances are shown in
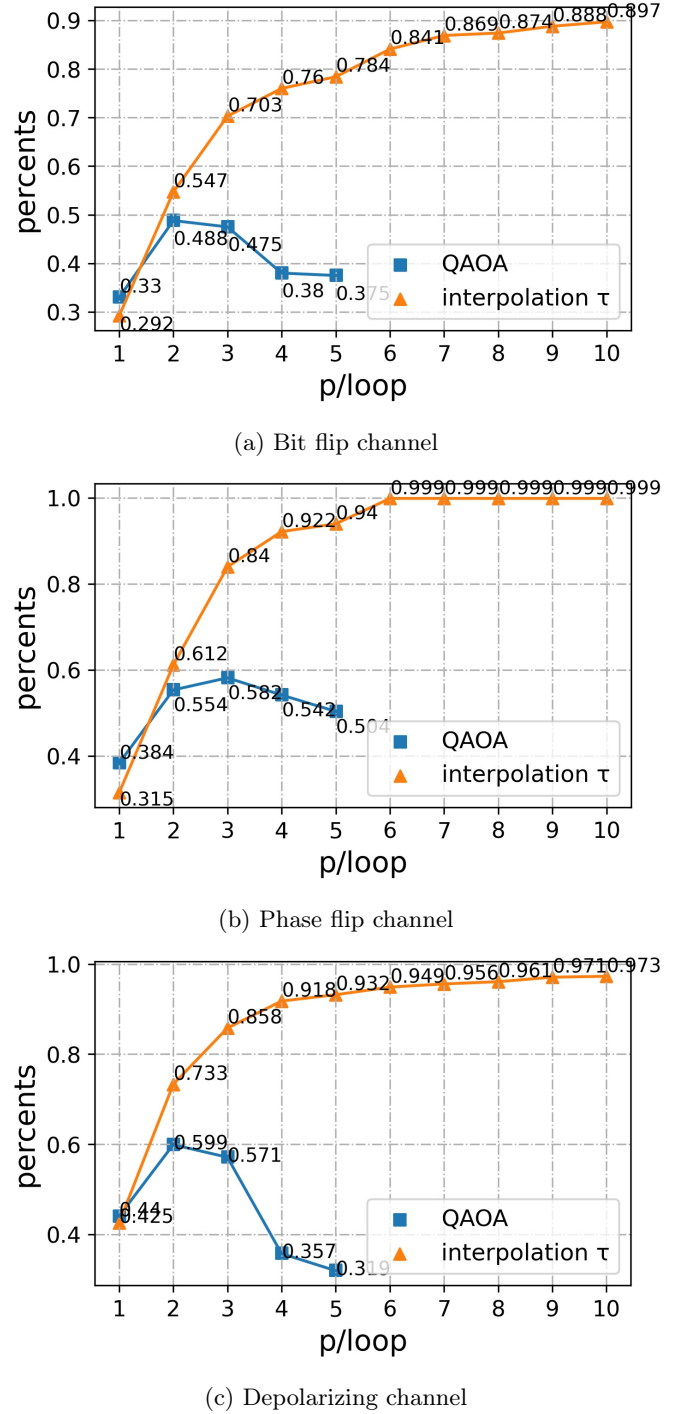

(a) Bit flip channel

(b) Phase flip channel

(c) Depolarizing channel

FIG. 3: **Performances of the conventional QAOA and the loop-QAOA at three different noise model environment.** (a),(b) and (c) shows the average success rates p_s on 10^4 randomly generated 12-vertex w3R graphs with bit flip channel, phase flip channel and depolarizing channel, respectively.

Fig. 2. Since QAOA can be regarded as the case where the bias constant $\tau = 0$, the performance of the two is almost the same here at $p = 1$ and loop $= 1$. At low p or low loops, the loop-QAOA strategy is slightly inferior to the original QAOA method. However, after a certain number of loops, the performance of loop-QAOA can gradually approach the original QAOA. When loop ≥ 8, the performance of the two is very close. For the conventional QAOA, this method yields a success rate of $p_s = 0.983$ at $p = 10$. The interpolation loop-QAOA reaches a success rate of $p_s = 0.979$, which is very close to the former. On the other hand, the loop-QAOA with random choices of bias strengths performs worse than the interpolation one. The above results indicate that the loop-QAOA can be a potential powerful quantum approximate optimizer using only a very shallow quantum circuit as long as appropriate bias field parameters are selected.

Remarkably, the loop-QAOA can have an advantage over the conventional QAOA when noises should be considered. Under three noise models as described in Eq. (9) (10) (12), we simulate the QAOA and the loop-QAOA on randomly generated 10^3 instances of w3R graphs with vertex number $N = 4$, and the results are shown in Fig. 3. We limit p to be less than or equal to 5 for consideration of simulation time and cost. Due to the existence of noise, the performance of QAOA improves slower with the increase of p and even can decrease for larger p. However, the performance of the loop-QAOA still raises with the loops and finally reaches a very high success rate, which significantly outperforms

the best performance of the conventional QAOA.

V. CONCLUSION

To conclude, we have proposed the loop-QAOA as a noise-resilience quantum algorithm for solving combinatorial optimization problems. The loop-QAOA exploits the outputs from a very shallow circuit as a bias to update the problem Hamiltonian and forms a feedback loop between Hamiltonian updating and parameter optimization. We have tested the loop-QAOA and compared it to the conventional QAOA on the Max-Cut problem with and without noises. Remarkably, the performance of the loop-QAOA under noises can still be made better by increasing the loop number, while the performance of the conventional QAOA will cease to increase at a small depth. Our work points out a direction for designing quantum algorithms with practical applications on NISQ devices with very shallow circuits.

[1] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum 2, 79 (2018).

[2] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics 3, 625 (2021).

[3] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New Journal of Physics 18, 023023 (2016).

[4] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature 549, 242 (2017).

[5] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational ansatz-based quantum simulation of imaginary time evolution, npj Quantum Information 5, 75 (2019).

[6] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. MÃŒller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, Quantum optimization using variational algorithms on near-term quantum devices, Quantum Science and Technology 3, 030503 (2018).

[7] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang, Variational quantum eigensolver with fewer qubits, Phys. Rev. Research 1, 023025 (2019).

[8] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller, Self-verifying variational quantum simulation of lattice models, Nature 569, 355 (2019).

[9] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Rev. Mod. Phys. 92, 015003 (2020).

[10] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nature Communications 5, 4213 (2014).

[11] P. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, et al., Scalable quantum simulation of molecular energies, Physical Review X 6, 031007 (2016).

[12] Z.-H. Yuan, T. Yin, and D.-B. Zhang, Hybrid quantum-classical algorithms for solving quantum chemistry in hamiltonian–wave-function space, Phys. Rev. A 103, 012413 (2021).

[13] F. Arute et al., Hartree-fock on a superconducting qubit quantum computer, Science 369, 1084 (2020).

[14] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028 [quant-ph].

[15] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, Phys. Rev. X 10, 021067 (2020).

[16] M. P. Harrigan *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, Nature Physics 17, 332 (2021).

[17] J. R. McClean, M. P. Harrigan, M. Mohseni, N. C. Rubin, Z. Jiang, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Low-depth mechanisms for quantum optimization, PRX Quantum 2, 030312 (2021).

[18] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem (2015), arXiv:1412.6062 [quant-ph].

[19] D. Wecker, M. B. Hastings, and M. Troyer, Training a quantum optimizer, Phys. Rev. A 94, 022309 (2016).

[20] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon, Optimizing variational quantum algorithms using pontryagin's minimum principle, Phys. Rev. X 7, 021027 (2017).

[21] W. W. Ho and T. H. Hsieh, Efficient unitary preparation of non-trivial quantum states, (2018), arXiv:1803.00026.

[22] Z. Jiang, E. G. Rieffel, and Z. Wang, Near-optimal quantum circuit for grover's unstructured search using a transverse field, Phys. Rev. A 95, 062317 (2017).

[23] E. R. Anschuetz, J. P. Olson, A. Aspuru-Guzik, and Y. Cao, Variational Quantum Factoring, (2018), arXiv:1808.08927.

[24] C. Moussa, H. Calandra, and V. Dunjko, To quantum or not to quantum: towards algorithm selection in near-term quantum optimization, Quantum Science and Technology 5, 044009 (2020).

[25] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte, Reachability deficits in quantum approximate optimization, Phys. Rev. Lett. 124, 090504 (2020).

[26] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, Obstacles to variational quantum optimization from symmetry protection, Phys. Rev. Lett. 125, 260505 (2020).

[27] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm (2019), arXiv:1602.07674 [quant-ph].

[28] L. Bittel and M. Kliesch, Training variational quantum algorithms is np-hard, Phys. Rev. Lett. 127, 120502 (2021).

[29] R. Shaydulin and Y. Alexeev, Evaluating quantum approximate optimization algorithm: A case study, 2019 Tenth International Green and Sustainable Computing Conference (IGSC) 10.1109/igsc48788.2019.8957201 (2019).

[30] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nature Communications 9, 4812 (2018).

[31] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, Phys. Rev. Lett. 119, 180509 (2017).

[32] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, Journal of the Physical Society of Japan 90, 032001 (2021), https://doi.org/10.7566/JPSJ.90.032001.

[33] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, Parameter concentrations in quantum approximate optimization, Phys. Rev. A 104, L010401 (2021).

[34] A. Lucas, Ising formulations of many np problems, Frontiers in Physics 2, 10.3389/fphy.2014.00005 (2014).

[35] J. Håstad, Some optimal inapproximability results, J. ACM 48, 798 (2001).

[36] P. Berman and M. Karpinski, On Some Tighter Inapproximability Results (Extended Abstract) (Springer, Berlin, Heidelberg, 1999) pp. 200–209.

[37] C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, Journal of Computer and System Sciences 43, 425 (1991).

[38] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM 42, 1115 (1995).

[39] E. Halperin, D. Livnat, and U. Zwick, MAX CUT in cubic graphs, Journal of Algorithms 53, 169 (2004).

[40] T. Graß, Quantum annealing with longitudinal bias fields, Phys. Rev. Lett. 123, 120501 (2019).

[41] G. Verdon, M. Broughton, and J. Biamonte, A quantum algorithm to train neural networks using low-depth circuits (2019), arXiv:1712.05304 [quant-ph].

[42] G. Giacomo Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms, arXiv e-prints , arXiv:1701.01450 (2017), arXiv:1701.01450 [quant-ph].

[43] J. S. Otterbach *et al.*, Unsupervised machine learning on a hybrid quantum computer (2017), arXiv:1712.05771 [quant-ph].

[44] M. M. Wilde, Preface to the second edition, Quantum Information Theory 10.1017/9781316809976.001.

[45] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).