Sequential minimal optimization for quantum-classical hybrid algorithms

Ken M. Nakanishi,[1,*] Keisuke Fujii,[2,3,†] and Synge Todo[1,4,‡]

[1]*Graduate School of Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*
[2]*Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan*
[3]*JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan*
[4]*Institute for Solid State Physics, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8581, Japan*

(Received 28 February 2020; revised 29 September 2020; accepted 30 September 2020; published 29 October 2020)

We propose a sequential minimal optimization method for quantum-classical hybrid algorithms, which converges faster, robust against statistical error, and hyperparameter-free. Specifically, the optimization problem of the parameterized quantum circuits is divided into solvable subproblems by considering only a subset of the parameters. In fact, if we choose a single parameter, the cost function becomes a simple sine curve with period 2π, and hence we can exactly minimize with respect to the chosen parameter. Furthermore, even in general cases, the cost function is given by a simple sum of trigonometric functions with certain periods and hence can be minimized by using a classical computer. By repeatedly performing this procedure, we can optimize the parameterized quantum circuits so that the cost function becomes as small as possible. We perform numerical simulations and compare the proposed method with existing gradient-free and gradient-based optimization algorithms. We find that the proposed method substantially outperforms the existing optimization algorithms and converges to a solution almost independent of the initial choice of the parameters. This accelerates almost all quantum-classical hybrid algorithms readily and would be a key tool for harnessing near-term quantum devices.

DOI: 10.1103/PhysRevResearch.2.043158

I. INTRODUCTION

Quantum computing devices with almost a hundred qubits are now within reach in the near future [1–3]. Since they have an unignorable amount of error because of a lack of error correction, they are called noisy intermediate-scale quantum (NISQ) devices. While the complexity of NISQ devices is limited because of the noise, still their classical simulation is considered to be intractable on classical computers if the fidelity of gates is sufficiently high [4–6].

In order to exploit NISQ devices in useful ways, quantum-classical hybrid algorithms have been proposed. Quantum-classical hybrid algorithms solve problems with a combination of sampling on low-depth quantum circuits and classical post-processing of the sampled outcomes. The variational quantum eigensolver (VQE) [7–10] is, for example, an attracting algorithm for NISQ devices for finding an approximate ground state of a given Hamiltonian $\mathcal{H}$. The VQE is now extended in various purposes beyond finding the ground state [11,12]. The quantum approximate optimization algorithm (QAOA) [13–15] makes it possible to get approximate solutions for combinatorial optimization problems. The quantum machine learning [16,17], especially for the near-term devices, has been proposed in various machine learning settings such as supervised learning [18–20], unsupervised learning [21], generative model [22], generative adversarial model [23], and so on.

All these major quantum-classical hybrid algorithms have a common structure: a parameterized quantum circuit and its optimization with respect to an observed cost function. More precisely, we repeatedly generate an ansatz state $|\psi(\boldsymbol{\theta})\rangle$ from a parameterized quantum circuit $U(\boldsymbol{\theta})$ on quantum devices and optimize the parameters $\boldsymbol{\theta}$ to minimize the expectation value of the given Hermitian operator, $\langle\mathcal{H}(\boldsymbol{\theta})\rangle = \langle\psi(\boldsymbol{\theta})|\mathcal{H}|\psi(\boldsymbol{\theta})\rangle$, on classical computers. Therefore, the convergence speed of the optimization process is a key factor that determines the overall performance of the algorithms.

Existing works employ either gradient-free [24–26] or gradient-based [27–31] optimization algorithms. For example, Peruzzo *et al.* [7] and Ryabinkin *et al.* [32] use the Nelder-Mead method, which is one of the gradient-free optimization algorithms [24]. Gradient-based methods are generally more efficient if we can get the gradients directly. In the context of the neural networks, the backpropagation method, which is an efficient method to calculate the gradients, is one of the most important ingredients to minimize the cost function of the model. To use gradient-based optimization algorithms for parameterized quantum circuits, Li *et al.* [33] and Mitarai *et al.* [18] propose an analytical way to take a partial derivative of parameterized quantum circuits, which allows us to obtain the gradient directly from the expectation value of a Hermitian operator. However, it is not yet fully explored whether a better optimization algorithm, specially designed for the parameterized quantum circuits, exists, or not. For example, in the optimization procedure in the support vector machine [34], the sequential minimal optimization [35] is widely used, in which the cost function is exactly minimized with respect to two

[*] ken-nakanishi@g.ecc.u-tokyo.ac.jp
[†] fujii@qc.ee.es.osaka-u.ac.jp
[‡] wistaria@phys.s.u-tokyo.ac.jp

parameters at each step by making use of the characteristic structure of the quadratic programming problem.

In this study, we propose an optimization method which is specialized for quantum-classical hybrid algorithms based on parameterized quantum circuits.[1] The idea is similar to the sequential minimal optimization for support vector machine; the cost function is minimized exactly with respect to certain chosen parameters at each step by using the characteristic structure of the parameterized quantum circuits. The proposed optimization method has several good properties: hyperparameter-free, faster convergence, less dependence on the initial choice of the parameters, and robust against the statistical error. To this end, we use the fact that the cost function, as a function of a parameter θ, behaves very simply as a sine curve with period 2π, when the parameterized quantum circuit consists of a unitary gate $\exp(\frac{i\theta}{2}A)$ being subject to $A^2 = I$. By virtue of this special property, we can determine the angle θ that provides the exact minimum of the cost function by evaluating the cost function at three independent points with respect to the parameter. This allows us to update each parameter with fewer measurements and to achieve a smaller value of the cost function. The update is deterministic if the order of the parameters is provided, and hence the proposed optimization method is hyperparameter-free. Independently from our work this periodic property of the cost function has been used as part of proposed methods for optimizing parameters of quantum circuits [36,37].

We perform detailed numerical simulations to compare the proposed method with existing optimization algorithms including both of gradient-free and gradient-based methods. To this end, we develop a benchmark task for which we can purely compare the performance of the optimization algorithms apart from the representation power of the parameterized quantum circuits. The results show that the proposed optimization method converges extremely faster than the existing ones, especially when the statistical error exists because of the finite number of the samples to estimate an expectation value. Furthermore, we find that the proposed method converges to the solution almost irrespective of the choice of the initial parameters, while other methods fail for certain initial parameters. These results mean that the proposed method readily accelerates almost all quantum-classical hybrid algorithms substantially in a practical situation. The proposed optimization method should be an inevitable ingredient in the parameter tuning of NISQ devices.

The rest of the paper is organized as follows. In Sec. II, we first describe the proposed optimization method for a single parameter optimization at each step, where we can find an exact minimum at each step. Then we further extend it for the case where multiple parameters are updated at each step. We also explain the case where multiple gates are parameterized by the same parameters. In Sec. III, we present numerical simulations to compare the proposed method with the existing optimization algorithms. We perform two tasks. One is a benchmark of the optimization of the parameterized quantum circuits, where we can achieve the exact solution if an optimization algorithm successfully finds the best parameters regardless of the representation power of the ansatz state. The other is VQE of the lithium hydride (LiH) molecule. Section IV is devoted to the conclusion.

II. METHODS

A. Preconditions

Our optimization method requires the following three conditions of the quantum-classical hybrid algorithms with parameterized quantum circuits. (1) The parameters of the parameterized quantum circuit are independent of each other. (This condition can be relaxed by extending the proposed method as we will see in Sec. II D.)

(2) The parameterized quantum circuit with J parameters $U(\boldsymbol{\theta})$ $(\boldsymbol{\theta} := \{\theta_j\}_{j=1}^J)$ is composed only of the two types of gates: fixed unitary gates (e.g., the Hadamard gate and the control-Z gate) and rotation gates

$$R_j(\theta_j) = \exp\left(-\frac{i\theta_j}{2}A_j\right) \quad (j = 1, 2, \ldots, J), \tag{1}$$

where A_j $(j = 1, 2, \ldots, J)$ satisfies the condition

$$A_j^2 = I \tag{2}$$

(e.g., the Z-rotation gate and the X-rotation gate).

(3) The cost function which we are going to minimize is written by the (weighted) sum of K expectation values:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \langle\varphi_k|U^\dagger(\boldsymbol{\theta})\mathcal{H}_k U(\boldsymbol{\theta})|\varphi_k\rangle\,, \tag{3}$$

where $\mathcal{H}_k$ $(k = 1, \ldots, K)$ are Hermitian operators such as Hamiltonian, $\{|\varphi_k\rangle\}_{k=1}^K$ are the input states, and w_k is the weight of the kth term. Hereinafter, we refer to $\mathcal{L}$ as the "cost function."

Most quantum-classical hybrid algorithms with parameterized quantum circuits, such as hardware efficient ansatz [10,20], satisfy these requirements.

B. Our method

Here we describe how parameters are updated in the proposed optimization method. Let $\boldsymbol{\theta}^{(n)}$ be the parameters of the circuit after n steps of update. Then, let $U_j^{(n)}(\theta_j)$ be the parameterized quantum circuit $U(\boldsymbol{\theta})$ in which the parameters are fixed to be $\boldsymbol{\theta}^{(n)}$ except for the jth parameter θ_j:

$$U_j^{(n)}(\theta_j) := U(\boldsymbol{\theta})|_{\theta_{j'}=\theta_{j'}^{(n)}\ (\text{for } j'\neq j)}. \tag{4}$$

Similarly, let $\mathcal{L}_j^{(n)}(\theta_j)$ be the cost function $\mathcal{L}(\boldsymbol{\theta})$ with the fixed parameters $\boldsymbol{\theta}^{(n)}$ except for the jth parameter θ_j:

$$\mathcal{L}_j^{(n)}(\theta_j) := \mathcal{L}(\boldsymbol{\theta})|_{\theta_{j'}=\theta_{j'}^{(n)}\ (\text{for } j'\neq j)}. \tag{5}$$

This cost function $\mathcal{L}_j^{(n)}(\theta_j)$ can be rewritten as a function of θ_j:

$$\begin{aligned}\mathcal{L}_j^{(n)}(\theta_j) &= \sum_{k=1}^{K} w_k \langle\varphi_k|U_j^{(n)\dagger}(\theta_j)\mathcal{H}_k U_j^{(n)}(\theta_j)|\varphi_k\rangle \\ &= a_{1j}^{(n)}\cos\left(\theta_j - a_{2j}^{(n)}\right) + a_{3j}^{(n)},\end{aligned} \tag{6}$$

where $a_{\ell j}^{(n)}$ $(\ell = 1, 2, 3)$ denote constants independent of θ_j. (See Appendix A for the derivation.) Equation (6) tells us that

[1] A reference implementation of the proposed method as an optimization module of scipy.optimize is available at https://github.com/ken-nakanishi/nftopt.

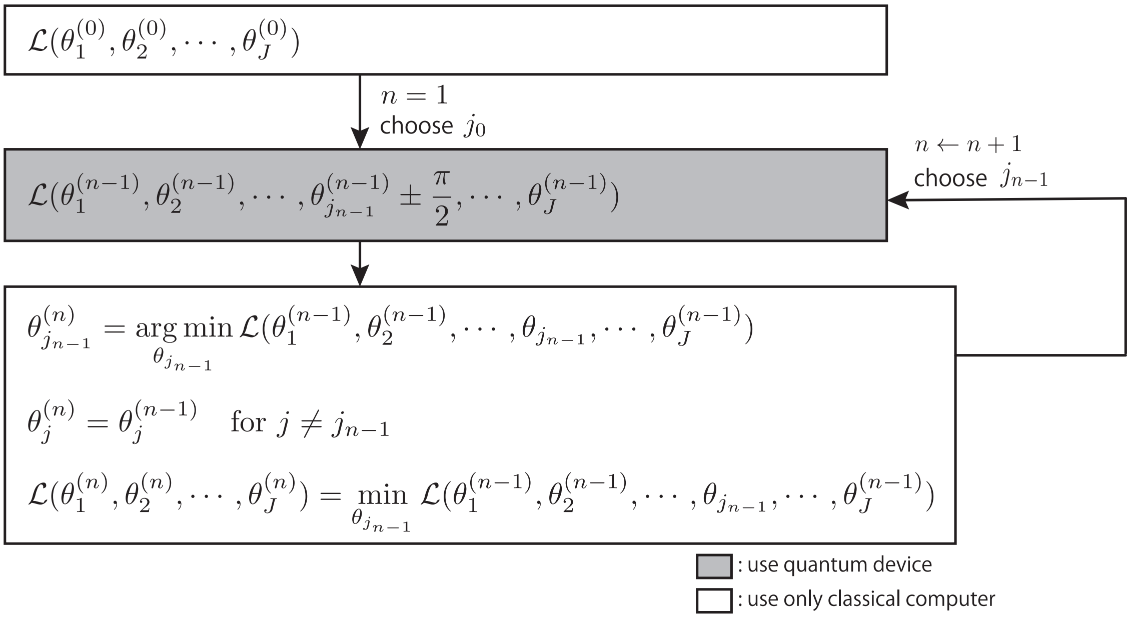
$\mathcal{L}(\theta_1^{(0)}, \theta_2^{(0)}, \cdots, \theta_J^{(0)})$
$n=1$
choose j_0
$n \leftarrow n+1$
choose j_{n-1}
$\mathcal{L}(\theta_1^{(n-1)}, \theta_2^{(n-1)}, \cdots, \theta_{j_{n-1}}^{(n-1)} \pm \frac{\pi}{2}, \cdots, \theta_J^{(n-1)})$
$\theta_{j_{n-1}}^{(n)} = \arg\min_{\theta_{j_{n-1}}} \mathcal{L}(\theta_1^{(n-1)}, \theta_2^{(n-1)}, \cdots, \theta_{j_{n-1}}, \cdots, \theta_J^{(n-1)})$
$\theta_j^{(n)} = \theta_j^{(n-1)}$ for $j \neq j_{n-1}$
$\mathcal{L}(\theta_1^{(n)}, \theta_2^{(n)}, \cdots, \theta_J^{(n)}) = \min_{\theta_{j_{n-1}}} \mathcal{L}(\theta_1^{(n-1)}, \theta_2^{(n-1)}, \cdots, \theta_{j_{n-1}}, \cdots, \theta_J^{(n-1)})$
▨ : use quantum device
□ : use only classical computer

FIG. 1. The flow chart of the method shown in Sec. II B. The gray box is the part of using quantum device, and the white boxes are the part of using only classical computer. Note that, at the top white box, the first and every few times we use a quantum device to estimate the cost function.

the relation of $\mathcal{L}_j^{(n)}(\theta_j)$ to θ_j is just a sine curve with period 2π. The three constants $a_{1j}^{(n)}$, $a_{2j}^{(n)}$, and $a_{3j}^{(n)}$ can be determined from the values of the cost function $\mathcal{L}_j^{(n)}(\theta_j)$ evaluated at three independent points of θ_j. Then we can find the argument θ_j that minimizes the cost function $\mathcal{L}_j^{(n)}(\theta_j)$, i.e., $\arg\min_{\theta_j} \mathcal{L}_j^{(n)}(\theta_j)$.

Using the above feature, we update the parameters as follows. The flowchart in Fig. 1 also shows the procedure for updating the parameters. (1) Estimate $\mathcal{L}(\boldsymbol{\theta}^{(0)})$ from a quantum device. (2) Choose an index $j_n \in \{1, 2, \cdots, J\}$ of the parameters sequentially or randomly. (3) Estimate $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}^{(n-1)} \pm \frac{\pi}{2})$ from a quantum device. Although the arguments do not have to be $\theta_{j_n}^{(n-1)} \pm \frac{\pi}{2}$, the present choice simplifies the optimization at step 3 greatly. (4) Determine θ_{j_n} minimizing the cost function $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n})$ given by Eq. (6) using $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}^{(n-1)})$, which was obtained in the previous update, and $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}^{(n-1)} \pm \frac{\pi}{2})$. (5) Update as follows:

$$\theta_{j_n}^{(n)} = \arg\min_{\theta_{j_n}} \mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}), \tag{7}$$

$$\theta_j^{(n)} = \theta_j^{(n-1)} \quad (\text{for } j \neq j_n), \tag{8}$$

$$\mathcal{L}_{j_{n+1}}^{(n)}\big(\theta_{j_{n+1}}^{(n)}\big) = \min_{\theta_{j_n}} \mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}). \tag{9}$$

Note that, the minimum is not estimated directly but calculated from the sine curve. Therefore the statistical error would accumulate. To avoid this, the minimum $\mathcal{L}_{j_{n+1}}^{(n)}(\theta_{j_{n+1}}^{(n)})$ should be estimated directly in a certain period. Under the circumstances in which there is no statistical error, on the other hand, one can further accelerate the convergence by performing a line search, such as the golden section search, along on the line connecting the initial and final points after each iteration over the whole parameters. (6) Repeat steps 2, 3, 4, and 5 until $\mathcal{L}(\boldsymbol{\theta})$ converges.

C. Generalization of our method

In the previous section, the minimization is performed with respect to one parameter at each update. Here we generalize it for a multiparameter case. Let $\boldsymbol{\theta}^{(n)}$ be the values of the parameters after n times of update as in Sec. II B. Then, let $U_M^{(n)}(\{\theta_j\}_{j\in M})$ be the parameterized quantum circuit $U(\boldsymbol{\theta})$ in which the parameters are chosen to be $\boldsymbol{\theta}^{(n)}$ except for a set of parameters $\{\theta_j\}_{j\in M}$ $(M \subset \{1, \cdots, J\})$:

$$U_M^{(n)}(\{\theta_j\}_{j\in M}) := U(\boldsymbol{\theta})|_{\theta_{j'}=\theta_{j'}^{(n)}\ (\text{for } j'\notin M)}, \tag{10}$$

where M is a subset of indices for which the minimization is performed. Similarly, let $\mathcal{L}_M^{(n)}(\{\theta_j\}_{j\in M})$ be the cost function $\mathcal{L}(\boldsymbol{\theta})$ with the fixed parameters $\boldsymbol{\theta}^{(n)}$ except for the parameters $\{\theta_j\}_{j\in M}$ $(M \subset \{1, \dots, J\})$:

$$\mathcal{L}_M^{(n)}(\{\theta_j\}_{j\in M}) := \mathcal{L}(\boldsymbol{\theta})|_{\theta_{j'}=\theta_{j'}^{(n)}\ (\text{for } j'\notin M)}. \tag{11}$$

This cost function $\mathcal{L}_M^{(n)}(\{\theta_j\}_{j\in M})$ can be rewritten as

$$\begin{aligned}
&\mathcal{L}_M^{(n)}\big(\{\theta_j\}_{j\in M}\big) \\
&= \sum_{k=1}^{K} w_k \, \langle\varphi_k| U_M^{(n)\dagger}\big(\{\theta_j\}_{j\in M}\big) \mathcal{H}_k U_M^{(n)}\big(\{\theta_j\}_{j\in M}\big) |\varphi_k\rangle \\
&= \boldsymbol{b}_M^{(n)} \cdot \left[\bigotimes_{j\in M} \begin{pmatrix} \cos\theta_j \\ \sin\theta_j \\ 1 \end{pmatrix}\right],
\end{aligned} \tag{12}$$

where $\bigotimes$ denotes the Kronecker product, and $\boldsymbol{b}_M^{(n)}$ denotes a $3^{|M|}$-dimensional coefficient vector with $|M|$ being the number of the elements of M. [See Appendix B for the derivation of Eq. (12).] Since Eq. (12) has $3^{|M|}$ parameters $\boldsymbol{b}_M^{(n)}$, we can determine $\boldsymbol{b}_M^{(n)}$ using the values of cost function $\mathcal{L}_M^{(n)}(\{\theta_j\}_{j\in M})$ evaluated at $3^{|M|}$ independent points of $\{\theta_j\}_{j\in M}$. Then we can find the values of $\{\theta_j\}_{j\in M}$ that minimize the cost function $\mathcal{L}_M^{(n)}(\{\theta_j\}_{j\in M})$. Unlike the method in Sec. II B, the true minimum cannot be found analytically. In the experiments in Sec. III, we numerically determined this with the BFGS method. Note that the minimum does not necessarily equal to the true minimum.

The optimization for the multiparameter update runs as follows. (1) Estimate $\mathcal{L}(\boldsymbol{\theta}^{(0)})$ from a quantum device. (2) Choose a subset $M_n \subset \{1, 2, \ldots, J\}$ of indices. (3) Estimate $\mathcal{L}_{M_n}^{(n-1)}(\{\theta_j^{(n-1)} + \frac{2\pi}{3}\alpha_j\}_{j\in M_n})$ for all $\boldsymbol{\alpha} \in \{0, \pm 1\}^{\otimes |M_n|} \setminus \{0\}^{\otimes |M_n|}$ using a quantum device, where α_j denotes the jth element of $\boldsymbol{\alpha}$. Although the above choices of the parameters are not necessary, they have the advantage that the coefficients in Eq. (12) can be easily determined by using the discrete Fourier transformation. (4) Determine $\{\theta_j\}_{j\in M_n}$ that minimizes the cost function $\mathcal{L}_{M_n}^{(n-1)}(\{\theta_j\}_{j\in M_n})$ from Eq. (12) using $\mathcal{L}_{M_n}^{(n-1)}(\{\theta_j^{(n-1)} + \frac{2\pi}{3}\alpha_j\}_{j\in M_n})$ for $\boldsymbol{\alpha} \in \{0, \pm 1\}^{\otimes |M_n|}$.

(5) Update as follows:

$$\left\{\theta_j^{(n)}\right\}_{j\in M_n} = \arg\min_{\{\theta_j\}_{j\in M_n}} \mathcal{L}_{M_n}^{(n-1)}\left(\{\theta_j\}_{j\in M_n}\right) \tag{13}$$

$$\theta_j^{(n)} = \theta_j^{(n-1)} \quad (\text{for } j \notin M_n), \tag{14}$$

$$\mathcal{L}_{M_{n+1}}^{(n)}\left(\left\{\theta_j^{(n)}\right\}_{j\in M_n}\right) = \min_{\{\theta_j\}_{j\in M_n}} \mathcal{L}_{M_n}^{(n-1)}\left(\{\theta_j\}_{j\in M_n}\right). \tag{15}$$

Similarly to the previous case, $\mathcal{L}_{M_{n+1}}^{(n)}(\{\theta_j^{(n)}\}_{j\in M_n})$ should be estimated directly in a certain period to avoid the error accumulation. (6) Repeat steps 2, 3, 4, and 5 until $\mathcal{L}(\boldsymbol{\theta})$ converges.

D. Special case of our method

In this section, we consider the case where the several rotation gates in the parameterized quantum circuit share the same parameter. This is the case when the target state has a symmetry, like translation invariance, and hence the ansatz state is also subject to it. Assume that the parameterized quantum circuit has J parameters $\boldsymbol{\theta} = \{\theta_j\}_{j=1}^J$, which are independent of each other. Unlike Secs. II B and II C, each θ_j is used at S_j times in the circuit. $\boldsymbol{\theta}^{(n)}$, $U_j^{(n)}(\theta_j)$, and $\mathcal{L}_j^{(n)}(\theta_j)$ is defined in the same way as Sec. II B. The cost function $\mathcal{L}_j^{(n)}(\theta_j)$ can be transformed as

$$\begin{aligned}\mathcal{L}_j^{(n)}(\theta_j) &= \sum_{k=1}^{K} w_k \langle\varphi_k| U_j^{(n)\dagger}(\theta_j)\mathcal{H}_k U_j^{(n)}(\theta_j)|\varphi_k\rangle \\ &= \sum_{s=1}^{S_j} a_{sj}^{(n)}\cos(s\theta) + \sum_{s=1}^{S_j} b_{sj}^{(n)}\sin(s\theta) + c_j^{(n)},\end{aligned} \tag{16}$$

where $a_{sj}^{(n)}, b_{sj}^{(n)}$ $(s = 1, \ldots, S_j)$ and c_j denote some constants. Note that these constants are independent of θ_j. (See Appendix C for the derivation.) Since Eq. (16) has $2S_j + 1$ parameters, $a_{sj}^{(n)}, b_{sj}^{(n)}$ $(s = 1, \ldots, S_j)$ and $c_j^{(n)}$, we can determine these constants using the values of cost function $\mathcal{L}_j^{(n)}(\theta_j)$ at $2S_j + 1$ different values of θ_j, and then find θ_j minimizing the cost function $\mathcal{L}_j^{(n)}(\theta_j)$.

Using this feature, we propose an updating method which is the same method of Sec. II B except for the following points. (1) Estimate $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}^{(n-1)} + \frac{2\pi s}{2S_j+1})$ $(s = 1, \ldots, 2S_j)$ using a quantum device. Although the above choices of the parameters are not necessary, they have the advantage that the coefficients can be determined easily by using the discrete Fourier transformation. (2) Determine θ_{j_n} minimizing the cost function $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n})$ [Eq. (16)] by using $\mathcal{L}_{j_n}^{(n-1)}(\theta_{j_n}^{(n-1)} + \frac{2\pi s}{2S_j+1})$ $(s = 0, \ldots, 2S_j)$.

III. NUMERICAL SIMULATION

A. Numerical setups

In this section, we numerically demonstrate the performance of the proposed methods via two types of optimization tasks. One (task 1) is a benchmark task of the parameter optimization problems, which we introduced here to compare the performance of different optimization algorithms. In this task, we minimize

$$\mathcal{L}(\boldsymbol{\theta}) = -\left|\langle 0|^{\otimes r}\, U^\dagger(\boldsymbol{\theta}^*) U(\boldsymbol{\theta})\, |0\rangle^{\otimes r}\right|^2, \tag{17}$$

where r is the number of qubits, and $\boldsymbol{\theta}^*$ is randomly chosen from a uniform distribution $[0, 2\pi)$ *a priori*. This task is equivalent to VQE whose Hamiltonian is given by $\mathcal{H} = U(\boldsymbol{\theta}^*)|0\rangle^{\otimes r}\langle 0|^{\otimes r} U^\dagger(\boldsymbol{\theta}^*)$. However, in our numerical simulations, we sample the output from the quantum circuit $U^\dagger(\boldsymbol{\theta}^*)U(\boldsymbol{\theta})|0\rangle^{\otimes r}$ in the $\{|0\rangle, |1\rangle\}$ basis for each qubit, which can be done even on an actual quantum device. Then we calculate the probability to obtain the outcome zero for all qubits similarly to the estimation of the inner product in Higgott *et al.* [38]. The cost function multiplied by -1 corresponds to the fidelity between two quantum states $U(\boldsymbol{\theta})|0\rangle^{\otimes r}$ and $U(\boldsymbol{\theta}^*)|0\rangle^{\otimes r}$. Since the minimum of the $\mathcal{L}(\boldsymbol{\theta})$ is exactly -1 and achieved at least when $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, we can purely compare the performance of the optimization algorithms apart from the representation power of the parameterized quantum circuits.

The other (task 2) is VQE for the lithium hydride (LiH) molecule at bond distance with four qubits. The molecular Hamiltonian $\mathcal{H}_{\text{LiH}}$ is the same as the one given in the supplemental material of Kandala *et al.* [10].

We use the method with the single-parameter minimization proposed in Sec. II B at each update (method II B). We sequentially choose a parameter to be optimized at each iteration, and re-estimate the cost function $\mathcal{L}_{j_{n+1}}^{(n)}(\theta_{j_{n+1}}^{(n)})$ once in 32 iterations. In addition, we show the results of the two-parameter simultaneous minimization version of the method II C (method II C 2). We randomly choose two parameters to be optimized at each iteration, and re-estimate the cost function $\mathcal{L}_{M_{n+1}}^{(n)}(\{\theta_j^{(n)}\}_{j\in M_n})$ once in 32 iterations. We compare the proposed methods with five existing optimization algorithms, Powell, Nelder-Mead, conjugate gradient (CG), BFGS, and SPSA methods, implemented in the SCIPY library [39] except for the SPSA method.[2] The hyperparameters of the SPSA

[2] Actually, we use a slightly modified code of them not to finish the optimization process in the middle. This modified code use for our experiment is available at https://github.com/ken-nakanishi/scipy.

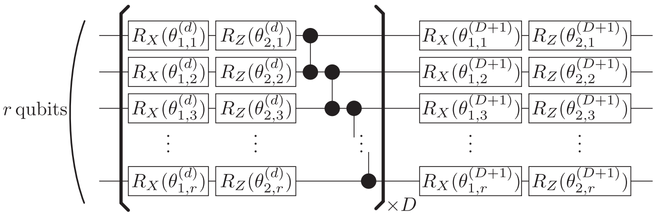
r qubits
$R_X(\theta^{(d)}_{1,1})$
$R_Z(\theta^{(d)}_{2,1})$
$R_X(\theta^{(d)}_{1,2})$
$R_Z(\theta^{(d)}_{2,2})$
$R_X(\theta^{(d)}_{1,3})$
$R_Z(\theta^{(d)}_{2,3})$
$R_X(\theta^{(d)}_{1,r})$
$R_Z(\theta^{(d)}_{2,r})$
$\times D$
$R_X(\theta^{(D+1)}_{1,1})$
$R_Z(\theta^{(D+1)}_{2,1})$
$R_X(\theta^{(D+1)}_{1,2})$
$R_Z(\theta^{(D+1)}_{2,2})$
$R_X(\theta^{(D+1)}_{1,3})$
$R_Z(\theta^{(D+1)}_{2,3})$
$R_X(\theta^{(D+1)}_{1,r})$
$R_Z(\theta^{(D+1)}_{2,r})$

FIG. 2. The parameterized quantum circuit used in the simulations of Sec. III. These parameters $\boldsymbol{\theta}$ are optimized to to minimize $\mathcal{L}$. D denotes the number of repetition of a circuit in the bracket.

method we used are the same as the default values in QISKIT v0.8.0 [40].

The initial values of the circuit parameters were randomly sampled from a uniform distribution $[0, 2\pi)$. In the present numerical simulations, we sampled the outcome 1024 times for the estimation of the cost function (except for the results shown in Fig. 4), which determine the amount of the statistical error. For each simulation, the optimization was run 100 times starting from different initial values of the parameters. In the following, *step* counts the number of the estimation of the expectation value of the cost function $\mathcal{L}(\boldsymbol{\theta})$.

B. Numerical results

In task 1, we set the number of qubits and the depth of the circuit in Fig. 2 to be $r = 5$ and $D = 9$, respectively. The total number of the parameters is thus 100. In Fig. 3, the horizontal axes represent the fidelity, and the vertical axes represent the number of samples whose fidelity is under the value of x axis for each of 1024, 2048, 4096, and 8192 steps from left to right. From Fig. 3, one can see that our methods, shown as solid lines, converge extremely faster than the other methods. Comparing the two proposed methods, method II B converges faster than method II C-2. Specifically, our method II B achieved fidelity higher than 0.98 after 8192 steps being independent of the initial set of parameters, while the conventional methods only result in far low fidelity for certain choices of the initial parameters. Figure 3 also shows that the gradient-based methods converge much faster than the gradient-free methods in general. Regarding the gradient-based methods, BFGS seems to be the best, while SPSA exhibits a better fidelity for some initial parameters. Furthermore, the Powell method outperforms the Nelder-Mead method when the number of steps is sufficiently large.

To investigate the statistical-error tolerance of each method, the number of samples to estimate the cost function is changed. In Fig. 4, we show the results after 8192 steps, for each of 256, 1024, 16 384, and ∞ outcomes from left to right. Here ∞ means that the cost function is directly calculated from the inner product. In the limit of the larger number of accumulations, Nelder-Mead, CG, BFGS, and the proposed methods achieve high fidelity with almost no dependence on the initial choice of the parameters. Specifically, BFGS and the proposed methods both result in fidelity close to unity. However, if the statistical error becomes larger with fewer accumulations, the advantage of the proposed methods gets larger. Notably, the proposed methods can achieve fidelity above 0.9 with only 256 accumulations for almost all choices of the initial parameters. Thus we conclude that the proposed methods are robust against the statistical error. We present the additional experiments of the method proposed in Sec. II D in Appendix D.

In task 2, the depth D in Fig. 2 is set to 4, and hence the circuit has 40 parameters. We show the results of VQE of the lithium hydride molecule in Fig. 5. The horizontal axes represent the energy difference (for top four figures) and fidelity (for bottom four figures) between the solution of VQE and the true ground state. The vertical axes represent the choices of the initial parameters, which are sorted by the values of each of energy difference and fidelity. From Fig. 5, one can see that Method II B gets closer to the true ground state much faster than the conventional methods we compared. Especially, method II B achieved fidelity higher than 0.95, which is considered the close-to-limit of the representation power of the prepared parameterized quantum circuits, after only 512 steps against almost all initial set of parameters, while the conventional methods only result in far low fidelity for certain choices of the initial parameters.

Let us finally argue why the proposed methods outperform the other existing optimization algorithms. The possible weak point of BFGS and CG methods is that these methods require the gradients of all the parameters. They are good methods

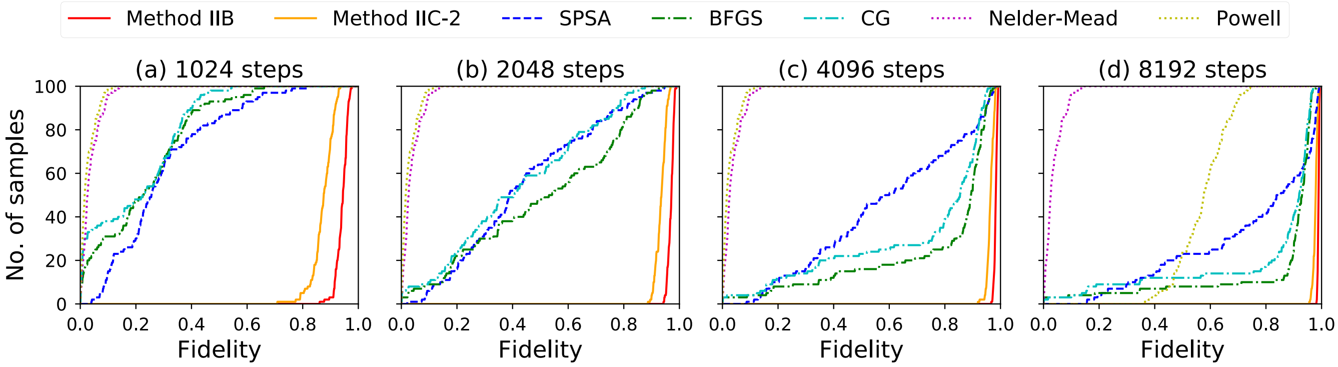

FIG. 3. Cumulative distribution function of fidelity after 1024, 2048, 4096, and 8192 steps in task 1. Method II B is denoted by the solid red (dark gray) lines. Method II C-2 is denoted by the solid orange (light gray) lines. The horizontal axes are the fidelity $|\langle 0|^{\otimes r} U^\dagger(\boldsymbol{\theta}^*) U(\boldsymbol{\theta}) |0\rangle^{\otimes r}|^2$. The vertical axis shows the number of samples whose fidelity is under the value of x axis at particular steps.

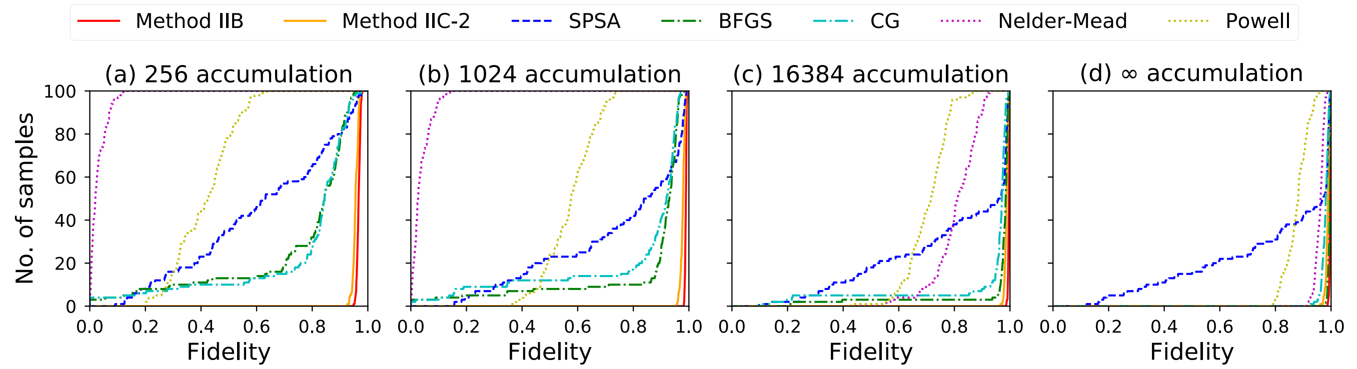

FIG. 4. Dependence of cumulative distribution function of fidelity on the strength of statistical error. Method II B is denoted by the solid red (dark gray) lines. Method II C-2 is denoted by the solid orange (light gray) lines. The horizontal axes are the fidelity $|\langle 0|^{\otimes r} U^{\dagger}(\boldsymbol{\theta}^{*})U(\boldsymbol{\theta})\,|0\rangle^{\otimes r}|^{2}$. The vertical axis shows the number of samples whose fidelity is under the value of x axis after 8192 steps.

when we can calculate all gradients at once, such as neural-network models. In the quantum-classical hybrid algorithm, however, we need to estimate the cost function twice as many as the number of the parameters to get all gradients of the parameter. This could be a disadvantage of these methods, since the step counts not the number of updates but the number of the estimation of the cost function in total. In SPSA method, we use the difference of the two cost functions $\frac{\mathcal{L}(\boldsymbol{\theta}+\delta\boldsymbol{\theta})-\mathcal{L}(\boldsymbol{\theta}-\delta\boldsymbol{\theta})}{2|\delta\boldsymbol{\theta}|}$ instead of the gradient. This method cannot avoid the additional noise on the gradient, and this can interrupt the fast convergence. In task 2, the SPSA method is better results than in task 1. It is considered that the gradients of the parameters in task 2 is farther apart of zero than ones in task 1 and then can be estimated better.

IV. CONCLUSION

In this work, we proposed an efficient optimization method for quantum-classical hybrid algorithms using parameterized quantum circuits. In most quantum-classical hybrid algorithms with parameterized quantum circuits, the relation of the cost function to each parameter of the circuit is just a sine curve with period 2π, on which our proposed method is based. To make a good use of the above property, we divide the optimization problem of the parameterized quantum circuits into solvable subproblems by considering only a subset of the parameters. By numerical simulations, we demonstrate that our method converges to a better solution much faster than the existing ones, especially in the presence of a large statistical

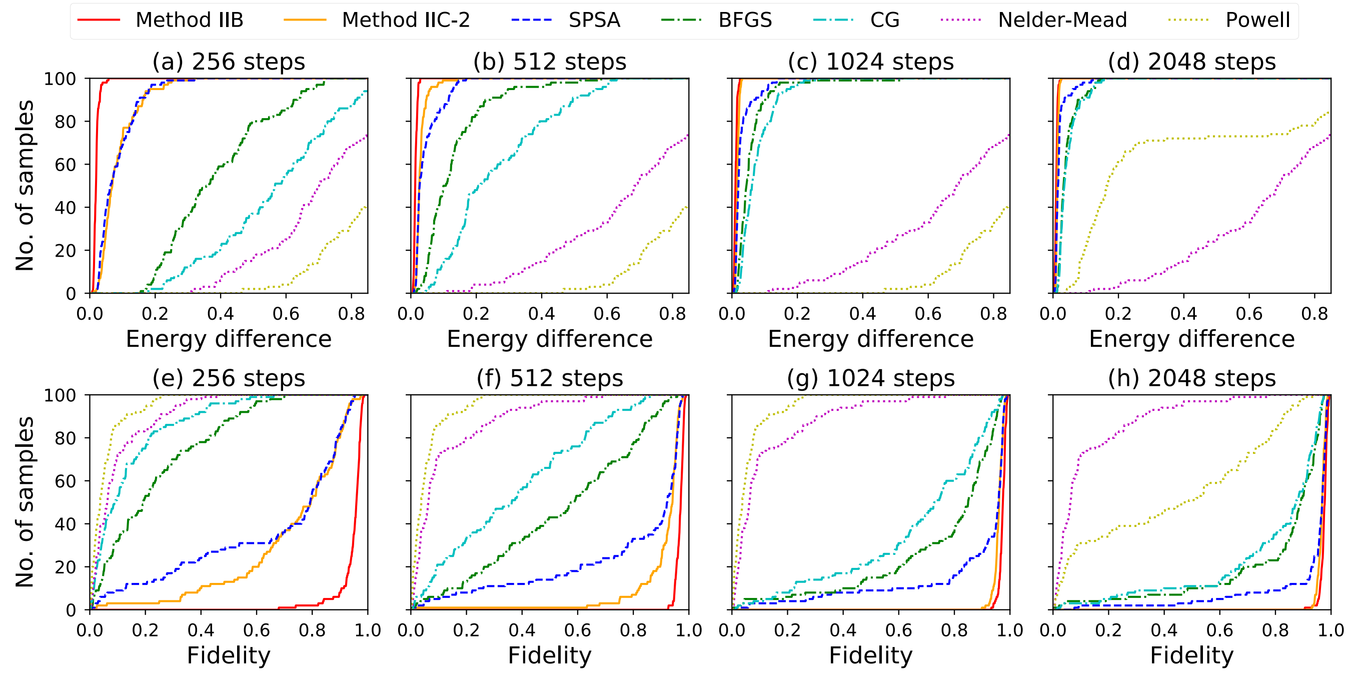

FIG. 5. Cumulative distribution function of energy difference and fidelity in VQE for the ground state of the Hamiltonian of lithium hydride molecule LiH (task 2). Method II B is denoted by the solid red (dark gray) lines. Method II C-2 is denoted by the solid orange (light gray) lines. The horizontal axes are the energy difference/fidelity between the calculated ground state and the true ground state. The vertical axes show the number of samples whose energy difference/fidelity is under the value of x axis at particular steps.

error. The proposed method is expected to have robustness not only to the statistical error but also to the noise in the NISQ devices due to the lack of the error correction, because the relation of the cost function to each parameter of the circuit would be robust against noise. This property of the present method would enable us to conduct parameterized-quantum-circuit-based variational algorithms on real quantum devices. We believe that this work drastically accelerates the quantum-classical hybrid algorithms and makes them practical in a realistic situation. Furthermore, the proposed method itself can be applied not only for optimizing parameterized quantum circuits but also for classical variational method. For example, the proposed method might be applied for the optimization of multi-scale entanglement renormalization ansatz (MERA) [41] by using variational unitary matrices inspired by parameterized quantum circuits. Neural networks composed of rotation matrices might also be considered as the another candidate.

ACKNOWLEDGMENTS

K.M.N. thanks IPA for its support through MITOU Target program. The computation in this work has been partly done using the facility of the Supercomputer Center, Institute for Solid State Physics, The University of Tokyo. K.F. is supported by KAKENHI No.16H02211, JST PRESTO JPMJPR1668, JST ERATO JPMJER1601, and JST CREST JPMJCR1673. This work is supported by MEXT, Q-LEAP.

APPENDIX A: DERIVATION OF EQ. (6)

Using Eq. (2), the jth rotation gate $R_j(\theta_j)$ can be rewritten as

$$R_j(\theta_j) = I\cos\frac{\theta_j}{2} - iA_j\sin\frac{\theta_j}{2}. \tag{A1}$$

The circuit $U_j^{(n)}(\theta_j)$ can be divided into three parts as follows:

$$U_j^{(n)}(\theta_j) = V_{1j}^{(n)}R_j(\theta_j)V_{2j}^{(n)}, \tag{A2}$$

where $V_{1j}^{(n)}$ and $V_{2j}^{(n)}$ are unitary operators which are independent of θ_j. The cost function $\mathcal{L}_j^{(n)}(\theta_j)$ can then be transformed as

$$\begin{aligned}
\mathcal{L}_j^{(n)}(\theta_j) &= \sum_{k=1}^{K} w_k \langle\varphi_k|U_j^{(n)\dagger}(\theta_j)\mathcal{H}_k U_j^{(n)}(\theta_j)|\varphi_k\rangle\\
&= \sum_{k=1}^{K} w_k \langle\,\varphi_k|V_{2j}^{(n)\dagger}R_j(-\theta_j)V_{1j}^{(n)\dagger}\mathcal{H}_k V_{1j}^{(n)}\\
&\quad\times R_j(\theta_j)V_{2j}^{(n)}|\varphi_k\rangle\\
&= \sum_{k=1}^{K} w_k \left\langle\varphi_{kj}^{\prime(n)}\right|\left(I\cos\frac{\theta_j}{2} + iA_j\sin\frac{\theta_j}{2}\right)\\
&\quad\times \mathcal{H}_{kj}^{\prime(n)}\left(I\cos\frac{\theta_j}{2} - iA_j\sin\frac{\theta_j}{2}\right)\left|\varphi_{kj}^{\prime(n)}\right\rangle\\
&\quad\times\left(\left|\varphi_{kj}^{\prime(n)}\right\rangle := V_{2j}^{(n)}|\varphi_k\rangle\,,\quad \mathcal{H}_{kj}^{\prime(n)} := V_{1j}^{(n)\dagger}\mathcal{H}_k V_{1j}^{(n)}\right)\\
&= \cos^2\frac{\theta_j}{2}\sum_{k=1}^{K} w_k\left\langle\varphi_{kj}^{\prime(n)}\middle|\mathcal{H}_{kj}^{\prime(n)}\middle|\varphi_{kj}^{\prime(n)}\right\rangle + \sin^2\frac{\theta_j}{2}\\
&\quad\times\sum_{k=1}^{K} w_k\left\langle\varphi_{kj}^{\prime(n)}\middle|A_j\mathcal{H}_{kj}^{\prime(n)}A_j\middle|\varphi_{kj}^{\prime(n)}\right\rangle + \sin\frac{\theta_j}{2}\cos\frac{\theta_j}{2}\\
&\quad\times\sum_{k=1}^{K} w_k\left\langle\varphi_{kj}^{\prime(n)}\middle|i\left(A_j\mathcal{H}_{kj}^{\prime(n)} - \mathcal{H}_{kj}^{\prime(n)}A_j\right)\middle|\varphi_{kj}^{\prime(n)}\right\rangle\\
&= c_{1j}^{(n)}\cos^2\frac{\theta_j}{2} + c_{2j}^{(n)}\sin^2\frac{\theta_j}{2} + c_{3j}^{(n)}\sin\frac{\theta_j}{2}\cos\frac{\theta_j}{2}\\
&\quad\times\left(c_{1j}^{(n)}, c_{2j}^{(n)}, c_{3j}^{(n)} \in \mathbb{R}\right)\\
&= c_{4j}^{(n)}\cos\theta_j + c_{5j}^{(n)}\sin\theta_j + c_{6j}^{(n)}\ \left(c_{4j}^{(n)}, c_{5j}^{(n)}, c_{6j}^{(n)} \in \mathbb{R}\right)\\
&= a_{1j}^{(n)}\cos\left(\theta_j - a_{2j}^{(n)}\right) + a_{3j}^{(n)}\quad\left(a_{1j}^{(n)}, a_{2j}^{(n)}, a_{3j}^{(n)} \in \mathbb{R}\right),
\end{aligned} \tag{A3}$$

where $c_{\ell j}^{(n)}$ $(\ell = 1, 2, \ldots, 6)$ and $a_{\ell j}^{(n)}$ $(\ell = 1, 2, 3)$ denote real constants independent of θ_j.

APPENDIX B: DERIVATION OF EQ. (12)

Mathematical induction can be used to prove that the following statement, $P(m)$, holds for all natural numbers m.

$$\begin{aligned}
P(m) :\ \exists\boldsymbol{b}\quad &\langle\varphi|U_m^{\dagger}\left(\{\theta_j\}_{j=1}^{m}\right)\mathcal{H}U_m\left(\{\theta_j\}_{j=1}^{m}\right)|\varphi\rangle\\
&= \boldsymbol{b}\cdot\left[\bigotimes_{j=1}^{m}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right],
\end{aligned} \tag{B1}$$

where $\bigotimes$ denotes Kronecker product, $U_m(\{\theta_j\}_{j=1}^{m})$ is a given parameterized quantum circuit which has m parameters and which satisfies the three conditions of Sec. II A, $\mathcal{H}$ is a given Hamiltonian, and $|\varphi\rangle$ is a quantum state. The order of the parameters $\{\theta_j\}_{j=1}^{m}$ is the same as the order of operating corresponding rotation gates.

When $m = 1$, the statement can prove in the similar manner as Eq. (A3).

Show that if $P(t)$ holds, then also $P(t+1)$ holds. This can be done as follows.

Assume $P(t)$ holds for some unspecified value of t. It must then be shown that $P(t+1)$ holds as follows.

The given circuit $U_{t+1}(\{\theta_j\}_{j=1}^{t+1})$ can be divided into three parts as follows:

$$U_{t+1}\left(\{\theta_j\}_{j=1}^{t+1}\right) = U_0R_{t+1}(\theta_{t+1})U_t\left(\{\theta_j\}_{j=1}^{t}\right), \tag{B2}$$

where U_0 denotes a unitary with no parameter and $U_t(\{\theta_j\}_{j=1}^{t})$ denotes a unitary with t parameter.

Therefore

$$\begin{aligned}
&\langle\varphi|U_{t+1}^{\dagger}\left(\{\theta_j\}_{j=1}^{t+1}\right)\mathcal{H}U_{t+1}\left(\{\theta_j\}_{j=1}^{t+1}\right)|\varphi\rangle\\
&\quad= \langle\,\varphi|U_t^{\dagger}\left(\{\theta_j\}_{j=1}^{t}\right)R_{t+1}(-\theta_{t+1})U_0^{\dagger}\mathcal{H}U_0R_{t+1}(\theta_{t+1})\\
&\qquad\times U_t\left(\{\theta_j\}_{j=1}^{t}\right)|\varphi\rangle
\end{aligned}$$

$$
\begin{aligned}
&= \langle \varphi|U_t^{\dagger}\big(\{\theta_j\}_{j=1}^{t}\big)\Big(I\cos\frac{\theta_{t+1}}{2} + iA_{t+1}\sin\frac{\theta_{t+1}}{2}\Big)\\
&\quad\times \mathcal{H}'\Big(I\cos\frac{\theta_{t+1}}{2} - iA_{t+1}\sin\frac{\theta_{t+1}}{2}\Big)U_t\big(\{\theta_j\}_{j=1}^{t}\big)|\varphi\rangle\\
&\quad\times \big(\mathcal{H}' := U_0^{\dagger}\mathcal{H}U_0\big)\\
&= \cos^2\frac{\theta_{t+1}}{2}\,\langle\varphi|U_t^{\dagger}\big(\{\theta_j\}_{j=1}^{t}\big)\mathcal{H}'U_t\big(\{\theta_j\}_{j=1}^{t}\big)|\varphi\rangle\\
&\quad+\ \sin^2\frac{\theta_{t+1}}{2}\,\langle\varphi|U_t^{\dagger}\big(\{\theta_j\}_{j=1}^{t}\big)A_{t+1}\mathcal{H}'A_{t+1}U_t\big(\{\theta_j\}_{j=1}^{t}\big)|\varphi\rangle\\
&\quad+\ \sin\frac{\theta_{t+1}}{2}\cos\frac{\theta_{t+1}}{2}\,\langle\varphi|U_2^{\dagger}\big(\{\theta_j\}_{j=1}^{t}\big)i(A_{t+1}\mathcal{H}'-\mathcal{H}'A_{t+1})\\
&\quad\times U_2\big(\{\theta_j\}_{j=1}^{t}\big)|\varphi\rangle\\
&= \boldsymbol{b}_1\cdot\left[\bigotimes_{j=1}^{t}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\cos^2\frac{\theta_{t+1}}{2}\\
&\quad+\boldsymbol{b}_2\cdot\left[\bigotimes_{j=1}^{t}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\sin^2\frac{\theta_{t+1}}{2}\\
&\quad+\boldsymbol{b}_3\cdot\left[\bigotimes_{j=1}^{t}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\sin\frac{\theta_{t+1}}{2}\cos\frac{\theta_{t+1}}{2}\\
&\quad\times\big(\boldsymbol{b}_1,\boldsymbol{b}_2,\boldsymbol{b}_3\in\mathbb{R}^{3^t}\big)\\
&= \boldsymbol{b}_4\cdot\left[\bigotimes_{j=1}^{t}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\cos\theta_{t+1}\\
&\quad+\boldsymbol{b}_5\cdot\left[\bigotimes_{j=1}^{t}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\sin\theta_{t+1}\\
&\quad+\boldsymbol{b}_6\cdot\left[\bigotimes_{j=1}^{t}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\quad\big(\boldsymbol{b}_4,\boldsymbol{b}_5,\boldsymbol{b}_6\in\mathbb{R}^{3^t}\big)\\
&= \boldsymbol{b}\cdot\left[\bigotimes_{j=1}^{t+1}\begin{pmatrix}\cos\theta_j\\ \sin\theta_j\\ 1\end{pmatrix}\right]\quad\big(\boldsymbol{b}\in\mathbb{R}^{3^{t+1}}\big).
\end{aligned}
\tag{B3}
$$

Thereby showing that indeed $P(t+1)$ holds.

Since both the base case and the inductive step have been performed, by mathematical induction the statement $P(m)$ holds for all natural numbers m.

APPENDIX C: DERIVATION OF EQ. (16)

Assume that the parameterized quantum circuit has S rotation gates which has the same parameter θ. Because of Eq. (B3), then cost function can written by

$$
\boldsymbol{b}\cdot\begin{pmatrix}\cos\theta\\ \sin\theta\\ 1\end{pmatrix}^{\otimes S}. \tag{C1}
$$

This equation is written as follows:

$$
\begin{aligned}
&\boldsymbol{b}\cdot\begin{pmatrix}\cos\theta\\ \sin\theta\\ 1\end{pmatrix}^{\otimes S}\\
&\quad= \sum_{\substack{p,\,q\,\in\,\mathbb{N}\\ p+q\,\leqslant\, S}} a_{p,q}\cos^p\theta\sin^q\theta\quad(a_{p,q}\in\mathrm{R})\\
&\quad= \sum_{s=0}^{S}\eta_s\cos^s\theta + \sum_{s=0}^{S-1}\xi_s\cos^s\theta\sin\theta\quad(\eta_s,\xi_s\in\mathbb{R})\\
&\quad= \sum_{s=0}^{S}\eta'_s\cos(s\theta) + \sum_{s=0}^{S-1}\xi'_s\cos(s\theta)\sin\theta\quad(\eta'_s,\xi'_s\in\mathbb{R})\\
&\quad= \sum_{s=0}^{S}\eta'_s\cos(s\theta) + \sum_{s=0}^{S-1}\frac{\xi'_s}{2}(\sin((s+1)\theta)-\sin((s-1)\theta))\\
&\quad= \sum_{s=0}^{S}\eta'_s\cos(s\theta) + \sum_{s=0}^{S}\xi''_s\sin(s\theta)\quad(\xi''_s\in\mathbb{R})\\
&\quad= \sum_{s=1}^{S}a_s\cos(s\theta) + \sum_{s=1}^{S}b_s\sin(s\theta) + c\quad(a_s,b_s,c\in\mathbb{R}).
\end{aligned}
\tag{C2}
$$

Using Eq. (C2), we can derive Eq. (16).

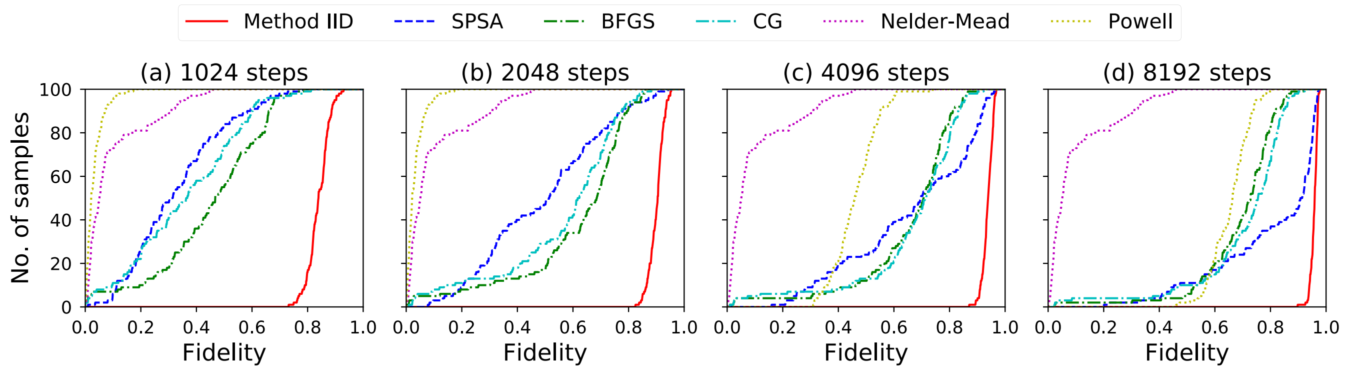

FIG. 6. Cumulative distribution function of fidelity after 1024, 2048, 4096, and 8192 steps in task 1. Method II D is denoted by the solid red lines. The horizontal axes are the fidelity $|\langle 0|^{\otimes r}U^{\dagger}(\boldsymbol{\theta}^{*})U(\boldsymbol{\theta})|0\rangle^{\otimes r}|^{2}$. The vertical axis shows the number of samples whose fidelity is under the value of x axis at particular steps.

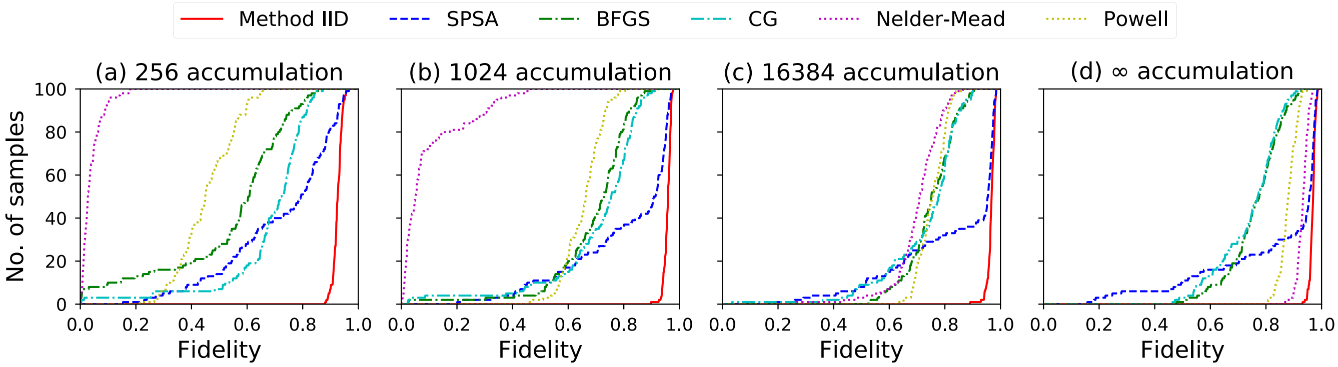

FIG. 7. Dependence of cumulative distribution function of fidelity on the strength of statistical error. Method II D is denoted by the solid red lines. The horizontal axes are the fidelity $|\langle 0|^{\otimes r} U^{\dagger}(\boldsymbol{\theta}^{*})U(\boldsymbol{\theta})|0\rangle^{\otimes r}|^{2}$. The vertical axis shows the number of samples whose fidelity is under the value of x axis after 8192 steps.

APPENDIX D: NUMERICAL SIMULATION OF SEC. II D

In this Appendix, we show the additional results of the method proposed in Sec. II D (method II D) comparing with the conventional methods. In task 1, we set the number of qubits and the depth of the circuit in Fig. 2 to be $r = 5$ and $D = 9$, respectively. The total number of the parameters is thus 100. Here, we divide them into 50 pairs, each of which takes the same value. In Fig. 6, the horizontal axes represent the fidelity, and the vertical axes represent the number of samples whose fidelity is under the value of x axis for each of 1024, 2048, 4096, and 8192 steps from left to right. The other experimental settings are the same as the task 1 in Sec. III.

To investigate the statistical-error tolerance of each method, the number of samples to estimate the cost function is changed. In Fig. 7, we show the results after 8192 steps, for each of 256, 1024, 16 384, and ∞ outcomes from left to right.

[1] J. Kelly, A Preview of Bristlecone, Googles New Quantum Processor (2018), https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html.

[2] W. Knight, IBM Raises the Bar with a 50-Qubit Quantum Computer (2017), https://www.technologyreview.com/2017/11/10/147728/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/.

[3] The Future of Quantum Computing is Counted in Qubits (2019), https://newsroom.intel.com/news/future-quantum-computing-counted-qubits/.

[4] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Nat. Phys. **14**, 595 (2018).

[5] A. Bouland, B. Fefferman, C. Nirkhe, and U. Vazirani, arXiv:1803.04402.

[6] J. Chen, F. Zhang, C. Huang, M. Newman, and Y. Shi, arXiv:1805.01450.

[7] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nat. Commun. **5**, 4213 (2014).

[8] J. R. McClean, I. D. Kivlichan, K. J. Sung, D. S. Steiger, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, P. Gokhale, T. Häner, T. Hardikar, V. Havlíček, C. Huang, J. Izaac, Z. Jiang, X. Liu, M. Neeley, T. O'Brien, I. Ozfidan, M. D. Radin, J. Romero, N. Rubin, N. P. D. Sawaya, K. Setia, S. Sim, M. Steudtner, Q. Sun, W. Sun, F. Zhang, and R. Babbush, arXiv:1710.07629.

[9] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, Phys. Rev. X **6**, 031045 (2016).

[10] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature (London) **549**, 242 (2017).

[11] K. M. Nakanishi, K. Mitarai, and K. Fujii, Phys. Rev. Res. **1**, 033062 (2019).

[12] K. Heya, Y. Suzuki, Y. Nakamura, and K. Fujii, arXiv:1810.12745.

[13] E. Farhi, J. Goldstone, and S. Gutmann, arXiv:1411.4028.

[14] E. Farhi and A. W. Harrow, arXiv:1602.07674.

[15] J. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong *et al.*, arXiv:1712.05771.

[16] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature (London) **549**, 195 (2017).

[17] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Quantum Sci. Technol. **4**, 043001 (2019).

[18] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018).

[19] N. Wiebe, A. Kapoor, and K. M. Svore, arXiv:1412.3489.

[20] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Nature (London) **567**, 209 (2019).

[21] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv:1307.0411.

[22] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, and M. H. Amin, Quantum Sci. Technol. **4**, 014001 (2018).

[23] P.-L. Dallaire-Demers and N. Killoran, Phys. Rev. A **98**, 012324 (2018).

[24] J. A. Nelder and R. Mead, Comput. J. **7**, 308 (1965).

[25] M. J. D. Powell, Comput. J. **7**, 155 (1964).

[26] A. R. Conn, K. Scheinberg, and P. L. Toint, Approximation theory and optimization: Tributes to MJD Powell, 83 (1997).

[27] R. Fletcher, *Practical Methods of Optimization* (Wiley, 2013), Chap. 3-4, pp. 44–94.

[28] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, SIAM J. Sci. Comput. **16**, 1190 (1995).

[29] D. P. Kingma and J. Ba, arXiv:1412.6980.

[30] M. R. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems* (NBS Washington, DC, 1952), Vol. 49.

[31] J. C. Spall *et al.*, IEEE transactions on automatic control **37**, 332 (1992).

[32] I. G. Ryabinkin, S. N. Genin, and A. F. Izmaylov, J. Chem. Theory Comput. **15**, 249 (2019).

[33] J. Li, X. Yang, X. Peng, and C.-P. Sun, Phys. Rev. Lett. **118**, 150503 (2017).

[34] V. Vapnik, Autom. Remote Control **24**, 774 (1963).

[35] J. Platt, Advances in Kernel Methods-Support Vector Learning **208** (1998).

[36] R. M. Parrish, J. T. Iosue, A. Ozaeta, and P. L. McMahon, arXiv:1904.03206.

[37] M. Ostaszewski, E. Grant, and M. Benedetti, arXiv:1905.09692.

[38] O. Higgott, D. Wang, and S. Brierley, Quantum **3**, 156 (2019).

[39] E. Jones, T. Oliphant, P. Peterson *et al.*, SciPy: Open source scientific tools for Python (2001).

[40] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernádez, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow, A. D. Córcoles-Gonzales, A. J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. D. L. P. González, E. D. L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I. F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B. G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Ł. Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krsulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F. J. Martín-Fernández, D. T. McClure, D. McKay, S. Meesala, A. Mezzacapo, N. Moll, D. M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L. J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyanov, M. Reuter, J. Rice, A. R. Davila, R. H. P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J. A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I. Y. Akhalwaya, and C. Zoufal, QISKIT: An Open-source Framework for Quantum Computing, Version 0.7.2 (2019), https://zenodo.org/record/2562111#.X5e6Tbm6b5o.

[41] G. Vidal, Phys. Rev. Lett. **101**, 110501 (2008).