Solving constrained optimization problems via the variational quantum eigensolver with constraints

Thinh Viet Le[*] and Vassilis Kekatos[†]
Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47906, USA

(Received 17 April 2024; accepted 31 July 2024; published 20 August 2024)

Variational quantum approaches have shown great promise in finding near-optimal solutions to computationally challenging tasks, including solving optimization problems. Nonetheless, optimization problems with constraints may not have been handled in a disciplined fashion thus far. To address this gap, this work proposes a hybrid quantum-classical algorithmic paradigm termed the variational quantum eigensolver with constraints (VQEC) that extends the celebrated VQE to handle optimization with constraints. As with the standard VQE, the vector of optimization variables is captured by the state of a variational quantum circuit (VQC). To deal with constraints, VQEC optimizes a Lagrangian function classically over both the VQC parameters as well as the dual variables associated with constraints. To comply with the quantum setup, variables are updated via a perturbed primal-dual method leveraging the parameter shift rule. Among a wide gamut of potential applications, we showcase how VQEC can approximately solve quadratically constrained binary optimization problems, find stochastic binary policies satisfying quadratic constraints on the average and in probability, and solve large-scale linear programs over the probability simplex. Under an assumption on the error for the VQC to approximate an arbitrary probability mass function, we provide bounds on the optimality gap attained by a VQC. Numerical tests on a quantum simulator investigate the effect of various parameters and corroborate that VQEC can generate high-quality solutions.

DOI: 10.1103/PhysRevA.110.022430

I. INTRODUCTION

Quantum computing could be a disruptive technology in dealing with challenging computational tasks. Seminal works have developed quantum computing algorithms to tackle problems, such as integer factorization [1], searching in databases [2], solving systems of linear equations [3,4], and various machine learning tasks [5,6], with polynomial or exponential speed-ups over their classical computing alternatives. However, these algorithms are assumed to operate on fault-tolerant quantum computers, which are projected not to be available in the near future. Recent research and development efforts focus on devising algorithms that are of relevance to practical applications on contemporary, qubit-limited, low-circuit depth, and noisy quantum hardware, often referred to as noisy intermediate-scale quantum (NISQ) [7]. Variational quantum approaches (VQAs) exploit parameterized quantum circuits (VQCs) of limited depth and reduced number of qubits, and become the leading candidates to showcase quantum advantage in the NISQ context [8,9].

The variational quantum eigensolver (VQE) is one of the most well-studied variational quantum approaches. Given a sequence of parameterized quantum gates, VQE aims at seeking the eigenvector corresponding to the minimum eigenvalue (energy) of an exponentially large Hermitian matrix representing a quantum observable [10]. While VQE has been successfully utilized as a heuristic to find near-optimal solutions for quadratic unconstrained binary optimization (QUBO) problems [11–15], quadratic problems oftentimes come with constraints. A constrained binary problem can be converted to an unconstrained one upon penalizing constraint violations by adding them to the objective function. Nonetheless, the weights associated with each penalty term are nontrivial to select unless the constraints are of specific forms, such as Boolean functions or linear equalities [16,17]. Another VQA that has been particularly successful for binary optimization is the quantum approximate optimization algorithm (QAOA) [18]. The QAOA is a special case of VQE that uses a problem-dependent VQC or ansatz.

To incorporate constraints, Refs. [19–21] adapt the mixer Hamiltonian of the QAOA's ansatz to ensure that the target quantum state remains within the feasible subspace. This strategy is also studied on quantum annealing, the analog counterpart of QAOA [22,23]. However, confining the mixer Hamiltonian applies only to a single linear equality constraint and requires a larger number of additional gates [21]. Reference [24] considers binary quadratic programs with linear constraints again, by combining the quantum adiabatic approach with the classical branch-and-bound method. Reference [14] proposes a VQA to minimize an objective expressed as a sum of a quadratic function over binary variables and a convex function over continuous variables. Binary and quadratic variables are set to be equal through linear equality constraints. The problem is solved using the alternating direction method of multipliers (ADMM). Each ADMM iteration entails solving the convex subproblem over the continuous variables using standard convex optimization techniques on a classical computer, and the QUBO subproblem

[*] Contact author: le272@purdue.edu
[†] Contact author: kekatos@purdue.edu

via VQE/QAOA. Nonetheless, solving a VQE or QAOA to optimality for each ADMM iteration may be computationally demanding. In [25] a VQA method is proposed to improve the chance of acquiring feasible solutions for constrained combinatorial problems through a greedy postprocessing method, which nonetheless is applicable only to linear constraints. To deal with linear and semidefinite programs by VQA, Ref. [26] transforms inequality constraints into equality constraints by means of introducing slack variables. Again, the penalty method is employed to modify the equality-constrained optimization to the unconstrained one. Solving the resultant unconstrained problem offers a lower bound on the optimal value of the original constrained problem. In practice, however, the penalty parameter should be chosen carefully, as unreasonably large values could lead to ill-conditioning. To find a suitable value that avoids numerical issues and gives a strict bound at the same time, the unconstrained problem has to be solved for a sequence of penalty parameters. Such penalty-based methods have been proposed in the context of quantum many-body problems with constraints as well [27–29].

Lagrangian duality offers a more systematic way of handling optimization problems with constraints. In the context of VQA, the vector of VQC parameters $\boldsymbol{\theta}$ constitutes the primal optimization variables, and the vector of Lagrange multipliers $\boldsymbol{\lambda}$ corresponds to dual variables. Albeit the primal problem is nonconvex, the dual problem is known to be always convex [30]. References [31,32] deal with quadratic constrained binary optimization (QCBO) using *dual decomposition*, a variation of subgradient descent that aims at solving the dual problem. Reference [33] engages dual decomposition too to solve semidefinite programs using variational quantum optimization. However, each update of dual variables involves a complete run of quantum annealing [31], or solving a VQE to optimality [32,33], either of which can be computationally formidable.

In contrast, the primal-dual method used in [34,35] to train neural networks under constrained optimization problems is more suitable, since each primal step requires only updating the primal variable inexactly. The convergence of the primal-dual method is guaranteed under strict settings, such as the Lagrangian function being strictly convex and strictly concave [36]. However, by updating the primal and dual variables at appropriate perturbed points, Refs. [37,38] ensures the sequence of the primal-dual pairs converging to the optimal point without putting strict assumptions on the Lagrangian function. Another variant of the primal-dual method with perturbations is used in [39], which is often referred to as the extragradient method (EGM). EGM has been expanded recently to the stochastic setting [40]. Nonetheless, in the VQC context, EGM increases substantially the number of VQC compilations.

Although our approach is presented for solving optimization problems with constraints, it can be expanded to other setups, such as when a VQC is used as a machine learning (ML) model. This idea has attracted sizable research interest recently; see, e.g., [41,42]. A popular choice of the loss function for quantum ML is the minimization of expectations of quantum observables concerning a quantum state prepared by the VQC. Although there is no quantum equivalent to automatic differentiation, Refs. [43,44] derive an analytical formula for the gradient of typical loss functions. However, estimating the gradient through quantum measurements is always subject to noise. Studies [45,46] show that estimating gradients using a finite number of measurements engenders an unbiased estimator and can facilitate stochastic gradient descent. In light of interpreting VQCs as ML models, it is also of crucial importance to explore VQCs from the perspective of learning models for constrained setups.

A. Contributions

Acknowledging the gap in incorporating constraints into VQAs and its relevance to optimization and ML tasks, the contribution of this work is on four fronts.

Contribution 1. Section II puts forth an algorithmic framework for handling constrained optimization problems in a principled manner via a VQA, termed the *variational quantum eigensolver with constraints* (VQEC). This framework applies to problems where cost and constraint functions can be captured as quantum observables over general, exponentially large, Hermitian matrices. As with VQE, rather than solving the problem over the original, exponentially large decision variable, VQEC adopts a hybrid quantum-classical approach. A quantum circuit parameterized over fewer parameters stored in vector $\boldsymbol{\theta}$ measures the observables, and a classical computer updates $\boldsymbol{\theta}$ iteratively. To incorporate constraints in a disciplined fashion, VQEC implements a primal-dual method, wherein the classical computer updates not only $\boldsymbol{\theta}$, but also the vector of Lagrange multipliers associated with the constraints. For improved convergence properties, a perturbed variant termed the perturbed primal-dual (PPD) method is adapted to the quantum setup by capitalizing on the parameter shift rule. Interestingly, if cost and constraint observables can be measured simultaneously, VQEC requires approximately the same quantum computations as VQE.

Contribution 2. Section III subsequently exemplifies how VQEC applies to problems with *diagonal observables* (observables defined over diagonal Hermitian matrices), and shows that a wide gamut of optimization tasks can be formulated as such. The list includes problems such as constrained quadratic binary optimization (QCBO), designing stochastic policies over binary-valued vectors that satisfy constraints on the average or in probability as chance constraints, learning large-scale PMFs, and solving large-scale LPs over the probability simplex. Such problems abound in diverse application domains, including reinforcement and machine learning in general, wireless communications, portfolio optimization, and optimal resource allocation. Although the focus of this work is on problems with diagonal observables, in principle, VQEC can be also applied to problems with nondiagonal observables. The implications arising when working with nondiagonal observables are the topic of ongoing research.

Contribution 3. Section IV provides analytical bounds on the optimality gap experienced when a problem with diagonal observables is solved in its variational form over $\boldsymbol{\theta}$ rather than its original form over an exponentially large variable. Under an assumption resembling the universal approximation theorem for deep neural networks, the optimality gap is shown to depend on the approximation within which a VQC can approximate any PMF as well as the sensitivity of the original problem to perturbations in the constraints.

Contribution 4. Section V numerically investigates the performance of VQEC on problems with diagonal observables. The tests demonstrate its convergence, reasonable performance even with finitely many quantum measurements, sensitivity to VQC depth, and ability to provide high-quality solutions to binary programs with constraints and large-scale LPs over the probability simplex.

Section VI concludes this work, and sketches exciting ongoing and future research directions for VQEC.

This Introduction closes with a quick note on notation: The symbol $(.)^\top$ denotes transposition; $(.)^{\mathcal{H}}$ stands for Hermitian transposition. Matrices (column vectors) are denoted by upper- (lower-) case boldface letters; scalars are denoted by lower-case letters. The operator $\mathrm{dg}(\mathbf{x})$ defines a diagonal matrix with vector $\mathbf{x}$ on its main diagonal. $\mathbb{E}$ is the expectation operator. Calligraphic symbols are reserved for sets.

II. PROPOSED ALGORITHM

This section develops a general algorithmic framework for incorporating constraints into variational quantum optimization. This broadens the applicability of VQE to cope with constrained programs over binary variables or large-scale continuous variables. We set the stage and introduce notation with a brief summary of VQE.

Given a $2^n \times 2^n$ Hermitian matrix $\mathbf{H}_0$, VQE aims at finding the eigenvector associated with the smallest eigenvalue of $\mathbf{H}_0$ [10]. From Rayleigh's quotient, this eigenvector coincides with the minimizer of

$$\min_{\mathbf{x}:\mathbf{x}^{\mathcal{H}}\mathbf{x}=1} \mathbf{x}^{\mathcal{H}}\mathbf{H}_0\mathbf{x}. \tag{1}$$

Because the dimension $N = 2^n$ of $\mathbf{x}$ is exponentially large, VQE models $\mathbf{x}$ via the state $|\mathbf{x}\rangle$ of a variational quantum circuit operating on n qubits. Given a prespecified ansatz, this quantum circuit is parameterized by a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^P$, and its output state is denoted by $|\mathbf{x}(\boldsymbol{\theta})\rangle$. Rather than solving (1), VQE solves the ensuing parameterized eigenproblem over $\boldsymbol{\theta}$:

$$\min_{\boldsymbol{\theta}} \langle \mathbf{x}(\boldsymbol{\theta})|\mathbf{H}_0|\mathbf{x}(\boldsymbol{\theta})\rangle . \tag{2}$$

The VQE operates in a hybrid quantum/classical computing fashion. A quantum computer samples from $|\mathbf{x}(\boldsymbol{\theta})\rangle$, and measures the observable

$$F_0(\boldsymbol{\theta}) := \langle \mathbf{x}(\boldsymbol{\theta})|\mathbf{H}_0|\mathbf{x}(\boldsymbol{\theta})\rangle \tag{3}$$

and possibly its gradient $\nabla_{\boldsymbol{\theta}}F_0(\boldsymbol{\theta})$. A classical computer subsequently optimizes $F_0(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ using standard optimization techniques, such as coordinate descent and stochastic gradient descent [42,46].

The cardinal question is how to extend VQE to handle optimization problems with constraints. We consider the prototypical constrained problem

$$\begin{aligned} P_\theta^* : = &\min_{\boldsymbol{\theta}} \ F_0(\boldsymbol{\theta}) \\ &\text{subject to (s. to) } F_m(\boldsymbol{\theta}) \leqslant 0, \quad m = 1 : M. \end{aligned} \tag{4}$$

Similarly to the cost function in (2), constraint functions should be expressible as quantum observables

$$F_m(\boldsymbol{\theta}) := \langle \mathbf{x}(\boldsymbol{\theta})|\mathbf{H}_m|\mathbf{x}(\boldsymbol{\theta})\rangle , \quad m = 1 : M$$

defined by Hermitian matrices $\mathbf{H}_m$. Problem (4) is nonconvex as quantum observables are known to be trigonometric functions of $\boldsymbol{\theta}$ [47].

Lagrange duality constitutes a systematic way of dealing with constrained optimization. Towards solving (4), let λ_m be the Lagrange multiplier or dual variable associated with inequality constraint $F_m(\boldsymbol{\theta}) \leqslant 0$. Stack all Lagrange multipliers in vector $\boldsymbol{\lambda} \in \mathbb{R}^M$. To simplify notation, let us also define the constant $\lambda_0 = 1$. Then the Lagrangian function of (4) can be expressed as

$$\mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}) := \sum_{m=0}^{M} \lambda_m F_m(\boldsymbol{\theta}). \tag{5}$$

The associated dual function is defined as

$$D_\theta(\boldsymbol{\lambda}) := \min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}),$$

and the corresponding dual problem aims at maximizing

$$D_\theta^* := \max_{\boldsymbol{\lambda}\geq\mathbf{0}} D_\theta(\boldsymbol{\lambda}). \tag{6}$$

The dual problem is convex regardless if the primal problem is convex or not. Moreover, weak duality asserts that $D_\theta^* \leqslant P_\theta^*$.

In [32] we attempted solving (6) using the iterative method of *dual decomposition*. Given an estimate $\boldsymbol{\lambda}^t$ of the optimal $\boldsymbol{\lambda}^*$ at the beginning of iteration t, iteration t updates the dual variables in two steps:

$$\boldsymbol{\theta}^t \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}^t), \tag{7a}$$

$$\lambda_m^{t+1} : = \left[\lambda_m^t + \mu_\lambda^t F_m(\boldsymbol{\theta}^t)\right]_+, \quad m = 1 : M, \tag{7b}$$

where $\mu_\lambda > 0$ is a step size and $[x]_+ = \max\{x, 0\}$ projects dual variables to the non-negative reals. Dual decomposition is known to be a projected subgradient ascent method to maximize $D_\theta(\boldsymbol{\lambda})$. This is because the constraint function value $F_m(\boldsymbol{\theta}^t)$ belongs to the subdifferential of $D_\theta(\boldsymbol{\lambda})$ with respect to λ_m evaluated at $\boldsymbol{\lambda}^t$. Obviously, the primal update step in (7a) constitutes a standard (unconstrained) VQE task using the observable $\mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}^t) = F_0(\boldsymbol{\theta}) + \sum_{m=1}^M \lambda_m^t F_m(\boldsymbol{\theta})$ rather than $F_0(\boldsymbol{\theta})$ alone. Unfortunately, solving a VQE task to optimality per iteration of the dual decomposition method is computationally impractical.

To alleviate this limitation, we propose switching from dual decomposition to the so-termed *primal-dual (PD) method*; see [36,48]. The latter seeks a *saddle point* of the Lagrangian function over primal/dual variables as

$$\max_{\boldsymbol{\lambda}\geq\mathbf{0}} \min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}). \tag{8}$$

A saddle point is a pair $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*) \in \mathbb{R}^P \times \mathbb{R}_+^M$ of primal-dual vectors satisfying

$$\mathcal{L}_\theta(\boldsymbol{\theta}^*; \boldsymbol{\lambda}) \leqslant \mathcal{L}_\theta(\boldsymbol{\theta}^*; \boldsymbol{\lambda}^*) \leqslant \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}^*)$$

for all $(\boldsymbol{\theta}, \boldsymbol{\lambda}) \in \mathbb{R}^P \times \mathbb{R}_+^M$. For problems where strong duality holds, a pair of primal-dual vectors is optimal if and only if it is a saddle point of the Lagrangian [30, p. 239]. Moreover, the optimal value of the Lagrangian function equals the optimal values of the primal and dual problems. How about the nonconvex problem over $\boldsymbol{\theta}$ in (4)? Although the latter may not feature strong duality, Sec. IV bounds the distance between $\mathcal{L}_\theta(\boldsymbol{\theta}^*;\boldsymbol{\lambda}^*)$ and the optimal cost of the original, nonparameterized constrained optimization problem. This motivates solving (4) using saddle point-seeking algorithms, such as the primal-dual method proposed next.

Instead of a fully fledged VQE, the PD method updates primal variables by taking a gradient descent step on $\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda})$ with respect to $\boldsymbol{\theta}$, evaluated at $(\boldsymbol{\theta}^t,\boldsymbol{\lambda}^t)$, that is,

$$\boldsymbol{\theta}^{t+1} := \boldsymbol{\theta}^t - \mu_\theta^t \nabla_{\boldsymbol{\theta}}\mathcal{L}_\theta(\boldsymbol{\theta}^t;\boldsymbol{\lambda}^t).$$

The required gradient follows from (5) as

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_\theta(\boldsymbol{\theta}^t;\boldsymbol{\lambda}^t) = \sum_{m=0}^{M}\lambda_m^t\nabla_{\boldsymbol{\theta}}F_m(\boldsymbol{\theta}^t) \tag{9}$$

with $\lambda_0^t = 1$ for all t. Analogously, the PD method updates $\boldsymbol{\lambda}$ by taking a gradient ascent step on $\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$, evaluated again at $(\boldsymbol{\theta}^t,\boldsymbol{\lambda}^t)$. The partial derivative of $\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda})$ with respect to λ_m is $F_m(\boldsymbol{\theta})$.

Putting the two steps together yields a single iteration of the PD method:

$$\boldsymbol{\theta}^{t+1} := \boldsymbol{\theta}^t - \mu_\theta^t\sum_{m=0}^{M}\lambda_m^t\nabla_{\boldsymbol{\theta}}F_m(\boldsymbol{\theta}^t), \tag{10a}$$

$$\lambda_m^{t+1} := \left[\lambda_m^t + \mu_\lambda^t F_m(\boldsymbol{\theta}^t)\right]_+,\ \ m = 1:M, \tag{10b}$$

where μ_θ^t and μ_λ^t are positive step sizes. We next elaborate on the practical implementation of (10) and comment on the related convergence guarantees.

Dual update step (10b). To update the vector of multipliers, we need to compute the quantum observables $\{F_m(\boldsymbol{\theta}^t)\}_{m=1}^M$. As with VQE and other VQAs, quantum observables cannot be computed exactly, but only estimated through measurements. More specifically, the quantum circuit is parameterized by $\boldsymbol{\theta}^t$ and is run S times. The number S will be referred to as the number of *measurement shots*. Given these S independent runs, the quantum observable can be estimated classically or quantumly, within accuracy ϵ if S scales as $O(\epsilon^{-2})$; see [42, p. 181]. For the optimization programs considered in this work and delineated later in Sec. III, all $M+1$ quantum observables involved in (4) are estimated classically and in parallel using the same S measurement shots.

Primal update step (10a). The gradients $\{\nabla_{\boldsymbol{\theta}}F_m(\boldsymbol{\theta}^t)\}_{m=0}^M$ are first estimated with the aid of the quantum computer. The primal variables are then updated using the classical computer. The required gradients can be computed using the *parameter shift rule* [43,44]. If the ansatz takes the form $\mathbf{U}(\boldsymbol{\theta}) = \prod_{i=p}^{P}\exp(-j\theta_p\mathbf{G}_p)$, where each $\mathbf{G}_p$ is a single-qubit Hermitian generator with two distinctive eigenvalues $\pm r$, each partial derivative of $F_m(\boldsymbol{\theta})$ with respect to θ_p can be computed exactly as the difference of two function evaluations at shifted values of $\boldsymbol{\theta}$:

$$\frac{\partial F_m(\boldsymbol{\theta}^t)}{\partial\theta_p} = r\left[F_m\left(\boldsymbol{\theta}^t + \tfrac{\pi}{4r}\mathbf{e}_p\right) - F_m\left(\boldsymbol{\theta}^t - \tfrac{\pi}{4r}\mathbf{e}_p\right)\right],$$

where $\mathbf{e}_p$ is the pth column of the identity $\mathbf{I}_P$. For example, if $\mathbf{G}_i$ are the generator matrices corresponding to Pauli rotations $\frac{1}{2}\{\rho_x,\rho_y,\rho_z\}$, then $r=\frac{1}{2}$ and the angle shift is $\frac{\pi}{2}$. As with $F_m(\boldsymbol{\theta}^t)$ in the dual update step, the quantum observables $F_m(\boldsymbol{\theta}^t \pm \frac{\pi}{4r}\mathbf{e}_p)$ can only be estimated through measurements. Overall, the quantum circuit has to be compiled for $2P$ different values of $\boldsymbol{\theta}$, two for each partial derivative. For each one of these $2P$ compilations, the quantum circuit is executed S times to estimate $F_m(\boldsymbol{\theta}^t \pm \frac{\pi}{4r}\mathbf{e}_p)$. For the problems considered later in Sec. III, quantum observables are estimated classically and in parallel for $m=0,\dots,M$. Heed that if $\lambda_m^t = 0$, there is no need to measure $\nabla_{\boldsymbol{\theta}}F_m(\boldsymbol{\theta}^t)$.

Overall, each PD iteration entails compiling the quantum circuit for $2P+1$ different values of $\boldsymbol{\theta}$ and executing the circuit $(2P+1)S$ times.

The convergence of the PD method has been widely studied for convex problems. For example, under a constant step-size rule a subsequence of the sequence $\{(\boldsymbol{\theta}_t,\boldsymbol{\lambda}_t)\}$ generated by the PD method is known to converge within a neighborhood of a saddle point of the Lagrangian [36]. For a diminishing step-size rule, a subsequence of the iterates has been shown to converge to a saddle point [48]. To guarantee that the entire sequence of PD updates converges to a saddle point, the function defining the saddle point should be strictly convex in $\boldsymbol{\theta}$ and strictly concave in $\boldsymbol{\lambda}$ [36], which unfortunately does not hold for the Lagrangian function in general.

These limitations of the PD method can be resolved using a slight modification, which enjoys convergence of the entire sequence without heavy assumptions on the Lagrangian function [37,38,49]. The so-termed *PPD method* updates primal/dual variables upon evaluating gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda})$ and $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda})$ not at $(\boldsymbol{\theta}^t,\boldsymbol{\lambda}^t)$, but at a pair of *perturbed* primal/dual variables denoted by $(\tilde{\boldsymbol{\theta}}^t,\tilde{\boldsymbol{\lambda}}^t)$. In particular, the primal-dual variables are updated as

$$\boldsymbol{\theta}^{t+1} := \boldsymbol{\theta}^t - \mu_\theta^t\nabla_{\boldsymbol{\theta}}\mathcal{L}_\theta(\boldsymbol{\theta}^t;\tilde{\boldsymbol{\lambda}}^t), \tag{11a}$$

$$\lambda_m^{t+1} := \left[\lambda_m^t + \mu_\lambda^t F_m(\tilde{\boldsymbol{\theta}}^t)\right]_+,\ \ m = 1:M. \tag{11b}$$

The perturbed variables are updated according to the standard primal-dual method, that is,

$$\tilde{\boldsymbol{\theta}}^t := \boldsymbol{\theta}^t - \nu_\theta\nabla_{\boldsymbol{\theta}}\mathcal{L}_\theta(\boldsymbol{\theta}^t;\boldsymbol{\lambda}^t), \tag{12a}$$

$$\tilde{\lambda}_m^t := \left[\lambda_m^t + \nu_\lambda F_m(\boldsymbol{\theta}^t)\right]_+,\ \ m = 1:M, \tag{12b}$$

for positive step sizes ν_θ and ν_λ.

Substituting (9) and putting the updates in order, each iteration of PPD involves the next four steps:

$$\tilde{\boldsymbol{\theta}}^t := \boldsymbol{\theta}^t - \nu_\theta\sum_{m=0}^{M}\lambda_m^t\nabla_{\boldsymbol{\theta}}F_m(\boldsymbol{\theta}^t), \tag{13a}$$

$$\tilde{\lambda}_m^t := \left[\lambda_m^t + \nu_\lambda F_m(\boldsymbol{\theta}^t)\right]_+,\ \ m = 1:M, \tag{13b}$$

$$\boldsymbol{\theta}^{t+1} := \boldsymbol{\theta}^t - \mu_\theta^t\sum_{m=0}^{M}\tilde{\lambda}_m^t\nabla_{\boldsymbol{\theta}}F_m(\boldsymbol{\theta}^t), \tag{13c}$$

$$\lambda_m^{t+1} := \left[\lambda_m^t + \mu_\lambda^t F_m(\tilde{\boldsymbol{\theta}}^t)\right]_+,\ \ m = 1:M \tag{13d}$$

with $\lambda_0^t = \tilde{\lambda}_0^t = 1$ for all t. The iterations in (13) constitute the proposed algorithm, termed *variational quantum eigensolver with constraints VQEC.*

Although the PPD method may seem complicated compared to PD, the additional computations do not incur

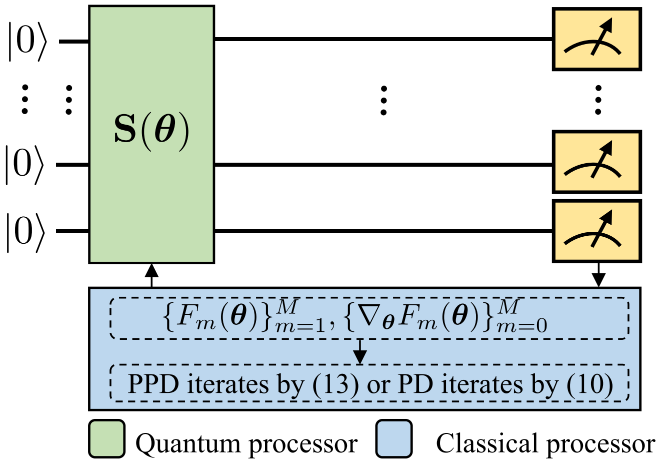
$|0\rangle$
$|0\rangle$
$|0\rangle$
$\mathbf{S}(\boldsymbol{\theta})$
$\{F_m(\boldsymbol{\theta})\}_{m=1}^M, \{\nabla_{\boldsymbol{\theta}} F_m(\boldsymbol{\theta})\}_{m=0}^M$
PPD iterates by (13) or PD iterates by (10)
Quantum processor
Classical processor

FIG. 1. Coordination between a quantum and a classical computer while running the PD or PPD method to solve a constrained optimization problem through a variational quantum approach. The VQC is encoded by unitary matrix $\mathbf{S}(\boldsymbol{\theta})$. The proposed method features minimal computational overhead over the standard, unconstrained VQE. If all involved quantum observables $\{F_m(\boldsymbol{\theta})\}_{m=0}^M$ can be measured simultaneously, the overhead is insignificant and is confined only to the classical computer.

significant overhead and are primarily run on the classical computer. A simple count of quantum observables to be measured reveals minimal quantum computation overhead. Fortunately, steps (13a) and (13c) share the exact same gradients $\nabla_{\boldsymbol{\theta}} F_m(\boldsymbol{\theta}^t)$, which can be evaluated using $2P$ compilations of the quantum circuit as in (10a). Step (13b) requires compiling the quantum circuit only for $\boldsymbol{\theta}^t$ as in (10b). Step (13d) introduces the sole extra quantum computation as it requires compiling the quantum circuit for $\tilde{\boldsymbol{\theta}}^t$. Overall, the PPD method requires $2P + 2$ compilations of the quantum circuit, whereas the PD method needs $2P + 1$ compilations. The workflow for implementing the PPD method is illustrated in Fig. 1. This completes the presentation of VQEC. Additional implementation details are provided in Sec. III, as such details pertain to the particular applications considered in this work. A salient feature of VQEC is highlighted next.

Remark 1. The VQEC framework is presented here in the context of optimization problems associated with diagonal observables. Nonetheless, VQEC can handle problems with nondiagonal observables as well. It should be emphasized that so long as cost and constraint observables are compatible (i.e., they can be measured simultaneously), the quantum computations of VQEC do not differ substantially from those of VQE, while the complexity of its classical computations increases only by the number of constraints M. Diagonal observables are obviously compatible, in which case the previous statement holds trivially. For problems with noncompatible nondiagonal observables, challenges related to measuring observables may arise. Dealing with these challenges is a topic of our ongoing work.

Some comments on the convergence of PPD are in order. As discussed earlier, the assumptions for the convergence of PD iterates are quite restrictive. PPD iterates enjoy more favorable convergence guarantees. If functions $\{F_m(\boldsymbol{\theta})\}_{m=0}^M$ are convex, continuously differentiable, and have Lipschitz continuous gradients, the PPD method with decreasing step sizes μ_θ^t and μ_λ^t generates a sequence of $\{(\boldsymbol{\theta}_t, \boldsymbol{\lambda}_t)\}$ converging to an optimal primal/dual pair [49]. Unfortunately, the parameterized variational problem in (4) is nonconvex, and hence, the mentioned guarantees may not carry over. The convergence analysis of VQEC becomes even more challenging due to its stochastic nature as quantum observables and their gradients are only measured in noise.

In the context of unconstrained optimization, albeit the convergence of stochastic gradient descent has not been fully analyzed, the method constitutes the spearhead for scaling up deep learning in classical computing. For variational quantum optimization without constraints, preliminary studies on the effect of stochastic gradients can be found in [45,46]. Toward handling constraints, stochastic PD methods have been utilized before in the context of training deep neural networks to satisfy stochastic constraints [34,35]. Nevertheless, the convergence of stochastic PD/PPD methods has not been fully established, even in the convex setting. In the convex setting, only saddle point problems featuring particular forms such as bilinear structure, have been analyzed so far. Despite the lack of convergence guarantees, the numerical tests of Sec. V study the effect of S and indicate that VQEC iterates do converge to meaningful points and exhibit superior performance over the stochastic PD iterates.

It is worth adding a quick note on another PPD variant, termed the extragradient method [39]. The perturbed points in EGM are computed as in (13a)–(13b), yet steps (13c)–(13d) are altered as

$$\boldsymbol{\theta}^{t+1} := \tilde{\boldsymbol{\theta}}^t - \mu_\theta^t \sum_{m=0}^{M} \tilde{\lambda}_m^t \nabla_{\boldsymbol{\theta}} F_m(\tilde{\boldsymbol{\theta}}^t),$$

$$\lambda_m^{t+1} := \left[\tilde{\lambda}_m^t + \mu_\lambda^t F_m(\tilde{\boldsymbol{\theta}}^t)\right]_+, \quad m = 1 : M.$$

Although seemingly minor, this modification incurs substantial computational overhead in the quantum setting. This is because measuring $\nabla_{\boldsymbol{\theta}} F_m(\tilde{\boldsymbol{\theta}}^t)$ requires $2P$ additional compilations of the quantum circuit.

III. APPLICATIONS

This section presents prototypical examples of optimization problems with constraints that can be handled by VQEC. Although VQEC applies to quantum observables of the general form $F_m(\boldsymbol{\theta}) = \langle \mathbf{x}(\boldsymbol{\theta})|\mathbf{H}_m|\mathbf{x}(\boldsymbol{\theta})\rangle$, we hereafter focus on observables defined by *diagonal* Hermitian matrices $\mathbf{H}_m$ for all m. This restriction simplifies the process of measuring cost and constraint observables and facilitates the performance analysis of Sec. IV. Despite the restriction, quantum observables with diagonal Hermitian matrices can handle a wide gamut of optimization problems, including binary problems with constraints and large-scale LPs over the probability simplex. Quantum observables with nondiagonal Hermitian matrices comprise the subject of our ongoing research.

Consider a diagonal Hermitian matrix $\mathbf{H}_m = \mathrm{dg}(\mathbf{f}_m)$, whose diagonal is defined by vector $\mathbf{f}_m \in \mathbb{R}^N$. It can be trivially verified that the corresponding quantum observable evaluates as

$$F_m(\boldsymbol{\theta}) = \langle \mathbf{x}(\boldsymbol{\theta})|\mathbf{H}_m|\mathbf{x}(\boldsymbol{\theta})\rangle = \mathbf{f}_m^\top \mathbf{p}(\boldsymbol{\theta}), \tag{14}$$

where the kth entry of vector $\mathbf{p}(\boldsymbol{\theta})$ is defined as

$$p_k(\boldsymbol{\theta}) := |\langle k|\mathbf{x}(\boldsymbol{\theta})\rangle|^2, \quad k = 0, \dots, N-1. \tag{15}$$

Apparently, observables defined by diagonal Hermitian matrices correspond to inner products. These observables can be measured on the computational basis. They will henceforth referred to as *diagonal observables*. For such observables, the general variational problem in (4) can be expressed as

$$\min_{\boldsymbol{\theta}} \ \mathbf{f}_0^\top \mathbf{p}(\boldsymbol{\theta}),$$
$$\text{s.to } \mathbf{f}_m^\top \mathbf{p}(\boldsymbol{\theta}) \leqslant 0, \quad m = 1:M. \tag{16}$$

Recall that diagonal observables admit an additional neat interpretation. By the properties of $|\mathbf{x}(\boldsymbol{\theta})\rangle$, vector $\mathbf{p}(\boldsymbol{\theta})$ defines a *probability mass function* (PMF). If vector $\mathbf{f}_m$ carries the N possible values of a discrete random variable distributed according to $\mathbf{p}(\boldsymbol{\theta})$, then $\mathbf{f}_0^\top \mathbf{p}(\boldsymbol{\theta})$ yields the mean value of this random variable.

Suppose we measure the VQC state $|\mathbf{x}(\boldsymbol{\theta})\rangle$ in the computational basis. The outcome of this measurement would be binary vectors $\mathbf{b} \in \{0,1\}^n$ distributed per $\mathbf{p}(\boldsymbol{\theta})$. For each $\mathbf{b}$, suppose we evaluate a quadratic or other function $f_m(\mathbf{b})$. If vector $\mathbf{f}_m$ carries the evaluations of f_m for all possible values of $\mathbf{b}$ [i.e., the kth entry of $\mathbf{f}_m$ is $f_m(|k\rangle)$], the diagonal observable provides

$$F_m(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}[f_m(\mathbf{b})], \tag{17}$$

where $\mathbb{E}_{\boldsymbol{\theta}}$ denotes the expectation operator over the PMF $\mathbf{p}(\boldsymbol{\theta})$ of random variable $\mathbf{b}$.

Given the previous interpretations, we next elaborate on what types of problems can be posed as (16). Before doing so, let us recall how VQE handles a famous unconstrained binary optimization problem.

A. QUBO and MaxCut

The *quadratic unconstrained binary optimization* (QUBO) is one of the problems tackled by VQE. It is originally posed as

$$\min_{\mathbf{b}\in\{0,1\}^n} f_0(\mathbf{b}) := \mathbf{b}^\top \mathbf{A}_0 \mathbf{b} + \mathbf{b}^\top \mathbf{c}_0 + d_0. \tag{QUBO}$$

Without harming generality, parameters $(\mathbf{A}_0, \mathbf{c}_0, d_0)$ are assumed to be real-valued, and matrix $\mathbf{A}_0$ is symmetric. QUBO is known to be NP-hard in general, yet VQE-based heuristics have been particularly successful in finding near-optimal solutions [11–15].

We briefly review how VQE is utilized for solving (QUBO). If we introduce the N-long vector $\mathbf{f}_0$ whose kth entry is $f_0(|k\rangle)$ for $k = 0, \dots, N-1$, problem (QUBO) can be equivalently reformulated as

$$\min_{\mathbf{p}\in\mathcal{P}_c} \mathbf{f}_0^\top \mathbf{p}, \tag{18}$$

where $\mathcal{P}_c$ is the set of all N canonical vectors in $\mathbb{R}^N$. The minimizer of (18) is the canonical vector corresponding to the smallest entry of $\mathbf{f}_0$. If (QUBO) has multiple minimizers, problem (18) will identify one of them. Problem (18) is as hard as QUBO.

In pursuit of a computationally more tractable solution, the feasible set of (18) can be relaxed from $\mathcal{P}_c$ to its convex hull, that is, the *probability simplex* in $\mathbb{R}^N$:

$$\mathcal{P} := \text{conv}(\mathcal{P}_c) = \{\mathbf{p} \in \mathbb{R}^N : \mathbf{p} \geqslant \mathbf{0}, \mathbf{p}^\top \mathbf{1} = 1\}. \tag{19}$$

This relaxation yields rise to a linear program (LP)

$$\min_{\mathbf{p}\in\mathcal{P}} \mathbf{f}_0^\top \mathbf{p}. \tag{20}$$

Problem (20) is equivalent to (QUBO) despite the relaxation. This is easy to see since the minimizer of the LP coincides with one of the corners of $\mathcal{P}$, or convex combinations thereof.

Despite being an LP, problem (20) is still computationally challenging as $\mathbf{p}$ is exponentially large. Rather than solving (20), VQE parameterizes $\mathbf{p}$ through $|\mathbf{x}(\boldsymbol{\theta})\rangle$ as in (15), and solves QUBO in the variational form

$$\min_{\boldsymbol{\theta}} \mathbf{f}_0^\top \mathbf{p}(\boldsymbol{\theta}). \tag{21}$$

Therefore, QUBO can be posed as an instance of the VQE task in (2) with a diagonal observable.

MaxCut is an instance of QUBO and an NP-hard problem [50], for which VQE/QAOA have been successful in finding candidate solutions [15,51,52]. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over vertex set $\mathcal{V} = \{1, 2, \dots, n\}$ and edge set $\mathcal{E} = \{(i,j) : i, j \in \mathcal{V}\}$ of edges weighted by non-negative w_{ij}, MaxCut aims at partitioning $\mathcal{V}$ into two subsets so that the size of the *cut* between the subsets is maximized. A cut is a set of edges spanning across two subsets of vertices. The size of a cut is the sum of its edge weights. For instance, in VLSI circuit design, each circuit component is modeled by a vertex. The preference to connect two components in the same or different layers of the circuit is captured by the indicated weight of the edge between them [53].

Let spin variable $s_i \in \{\pm 1\}$ indicate the partition vertex i is assigned to. If vector $\mathbf{s} \in \{\pm 1\}^n$ collects all spin variables, the cut size defined by assignment $\mathbf{s}$ is

$$\frac{1}{4}\sum_{(i,j)\in\mathcal{E}} w_{ij}(1 - s_i s_j).$$

Obviously, edge (i, j) contributes unity to the cut only when s_i and s_j have different signs. If symmetric matrix $\mathbf{W}$ stores the edge weights as $\mathbf{W}_{ij} = w_{ij}$, MaxCut is equivalent to minimizing $\mathbf{s}^\top \mathbf{W}\mathbf{s}$. The latter can be written as a QUBO upon converting spin to binary variables via the transformation $b_i = (1 - s_i)/2$ for $i = 1, \dots, n$. Having reviewed QUBO and MaxCut, we next embark on incorporating constraints into them.

B. Stochastic QCBO and learning PMFs

Consider the *quadratic constrained binary optimization* (QCBO) problem:

$$\min_{\mathbf{b}\in\{0,1\}^n} f_0(\mathbf{b})$$
$$\text{s. to } f_m(\mathbf{b}) \leqslant 0, \quad m = 1:M \tag{22}$$

where $f_m(\mathbf{b}) := \mathbf{b}^\top \mathbf{A}_m \mathbf{b} + \mathbf{b}^\top \mathbf{c}_m + d_m$ for $m = 0, \dots, M$. As with (QUBO), parameters $(\mathbf{A}_m, \mathbf{c}_m, d_m)$ are real-valued, and matrices $\mathbf{A}_m$ are symmetric for all m. Being a generalization of QUBO, QCBO is also NP-hard. A motivating example of a QCBO is discussed next and more can be found in [54–56].

Albeit MaxCut has been widely studied as an unconstrained problem, constrained versions are of relevance too. In the VLSI design example, the designer may know in advance that specific pairs of components must be on the same or different layers. In that setting, a MaxCut with constraints can minimize the cost of wires used to connect components within and between layers while respecting prior connectivity specifications. Specifications can be encoded in matrix $\mathbf{C}$ with entries [55]

$$\mathbf{C}_{ij} = \begin{cases} +1, & \text{if } (i, j) \text{ are in the same partition} \\ -1, & \text{if } (i, j) \text{ are in different partitions} \\ 0, & \text{no prior information on } (i, j) \text{ or } i = j. \end{cases}$$

We can now define a constrained version of MaxCut [55]:

$$\min_{\mathbf{s}\in\{\pm1\}^n} \quad \mathbf{s}^\top \mathbf{W}\mathbf{s}, \tag{23a}$$

$$\text{s.to } \mathbf{s}^\top \mathbf{C}\mathbf{s} \geqslant \sum_{i=1}^{n}\sum_{j=1}^{n} |\mathbf{C}_{ij}|. \tag{23b}$$

Obviously, if a pair (i, j) is correctly assigned to partitions, it contributes $2\mathbf{C}_{ij}s_is_j = +2$ in the left-hand side of (23b). Otherwise, it contributes -2 and the constraint is violated. Upon converting spin to binary variables, problem (23) can be posed as a QCBO.

Each specification can also be expressed as a linear constraint $s_i = \mathbf{C}_{ij}s_j$. However, the formulation in (23) has a single quadratic constraint instead of multiple linear ones. In fact, the single quadratic constraint can be obtained by squaring and summing up the linear constraints because $\sum_{(i,j):\mathbf{C}_{ij}\neq 0}(s_i - \mathbf{C}_{ij}s_j)^2 \leqslant 0$ is equivalent to (23b). In general, linear equality constraints such as $\mathbf{E}\mathbf{b} = \mathbf{g}$ can be handled by a single quadratic constraint $\|\mathbf{E}\mathbf{b} - \mathbf{g}\|_2^2 = 0$ or $\|\mathbf{E}\mathbf{b} - \mathbf{g}\|_2^2 \leqslant 0$. MaxCut may also come with the balance constraint $-B \leqslant \mathbf{s}^\top \mathbf{1} \leqslant B$, which ensures that the cardinalities of the two partitions do not differ more than a given constant B from each other.

Given the maturity of mixed-integer linear program (MILP) solvers, quadratic binary programs are oftentimes converted to MILPs. This is possible by introducing an auxiliary variable z_{ij} for each product b_ib_j of binary variables. The constraint $z_{ij} = b_ib_j$ is then handled using McCormick linearization, which requires a few linear constraints involving (z_{ij}, b_i, b_j). Nonetheless, this approach can increase the number of constraints and variables by $O(n^2)$. Hence, it may be meaningful to solve quadratic binary problems directly and use quantum computing approaches in particular. Having motivated the need for QCBOs, we next resume with variational quantum approaches for solving them.

Mimicking QUBO, QCBO can be recast as a minimization over the canonical vectors of $\mathbb{R}^N$ as

$$\begin{aligned} &\min_{\mathbf{p}\in\mathcal{P}_c} \mathbf{f}_0^\top \mathbf{p} \\ &\text{s. to } \mathbf{f}_m^\top \mathbf{p} \leqslant 0, \quad m = 1 : M. \end{aligned} \tag{24}$$

As with $\mathbf{f}_0$, each N-long vector $\mathbf{f}_m$ evaluates the mth quadratic function over all possible values of the binary vector $\mathbf{b}$. The feasible set of (24) can be subsequently relaxed from $\mathcal{P}_c$ to the probability simplex $\mathcal{P}$, to yield the large-scale LP

$$\begin{aligned} &\min_{\mathbf{p}\in\mathcal{P}} \mathbf{f}_0^\top \mathbf{p} \\ &\text{s. to } \mathbf{f}_m^\top \mathbf{p} \leqslant 0, \quad m = 1 : M. \end{aligned} \tag{25}$$

Contrary to QUBO, the minimizer of (25) may not be at a vertex of $\mathcal{P}$. Hence, the problem in (25) is *not* equivalent to (22). Nonetheless, problem (25) is of interest in its own right as will be explicated shortly. We first explain how (25) can be solved variationally and then discuss possible applications.

Solving (25) classically is technically challenging. Computing the values of vectors $\{\mathbf{f}_m\}_{m=0}^M$ alone requires $O(NMn^2)$ operations. The size of $\mathbf{p}$ precludes interior-point methods, while first-order methods would require at least $O(MN)$ operations per iteration only to evaluate the constraint functions. In contrast, a quantum approach could offer a more practical solution as delineated next. Again, variable $\mathbf{p}$ is substituted by its parameterized form $\mathbf{p}(\boldsymbol{\theta})$, and problem (25) is surrogated by the variational problem in (16) with diagonal observables having $\mathbf{H}_m = \text{dg}(\mathbf{f}_m)$ for $m = 0, \ldots, M$. Consequently, it can be handled by VQEC.

We coin (25) and its variational form as the *average QCBO* for the following reason. The VQC state $|\mathbf{x}(\boldsymbol{\theta})\rangle$ can be used as a *sampler* of binary vectors $\mathbf{b} \in \{0, 1\}^n$ drawn from PMF $\mathbf{p}(\boldsymbol{\theta})$. From the viewpoint of (17), the sampled binary vectors minimize the average cost $\mathbb{E}_{\boldsymbol{\theta}}[f_0(\mathbf{b})]$ and satisfy constraints in the average sense $\mathbb{E}_{\boldsymbol{\theta}}[f_m(\mathbf{b})] \leqslant 0$ for $m = 1, \ldots, M$.

The average QCBO can be alternatively interpreted as the task of *learning a PMF*. The PMF $\mathbf{p}(\boldsymbol{\theta})$ is designed to satisfy specifications when applied to given functions. This could be of relevance to machine learning tasks over exponentially large PMFs. Such PMFs arise when dealing with *joint* PMFs over discrete-valued random variables (categorical), and/or probability density functions (PDFs) over continuous random variables that have been finely quantized. Moreover, in certain applications (e.g., reinforcement learning, wireless communications, optimal scheduling), it may be of interest to find a *stochastic policy* to draw binary vectors from, that solve the average QCBO of (4). The optimized quantum circuit $|\mathbf{x}(\boldsymbol{\theta})\rangle$ can serve as such policy. The policy can also be used to sample candidate solutions for the deterministic QCBO in (22), yet more elaborate solutions for that follow.

C. QCBO and chance-constrained QCBO

So far, vectors $\mathbf{f}_m$ were assumed to evaluate quadratic functions $f_m(\mathbf{b})$ of binary vectors. Nevertheless, functions $f_m(\mathbf{b})$ do not have to be quadratic necessarily. Let us see an interesting example. Consider again the QCBO in (22) and define functions

$$g_m(\mathbf{b}) = \begin{cases} 1, & f_m(\mathbf{b}) \leqslant 0 \\ 0, & f_m(\mathbf{b}) > 0 \end{cases}, \quad m = 1, \ldots, M. \tag{26}$$

We next let vectors $\mathbf{f}_m$ evaluate functions $g_m(\mathbf{b})$ rather than $f_m(\mathbf{b})$ for $m = 1, \ldots, M$, and evaluate the corresponding diagonal observables with Hermitian matrices $\mathbf{H}_m = \text{dg}(\mathbf{f}_m)$. It is not hard to verify that the new constraint observables compute the probability

$$F_m(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}[g_m(\mathbf{b})] = \Pr(f_m(\mathbf{b}) \leqslant 0).$$

Vector $\mathbf{f}_0$ still evaluates the original quadratic $f_0(\mathbf{b})$, and so observable $F_0(\boldsymbol{\theta})$ remains unchanged.

Due to the previous modeling, the constraint

$$F_m(\boldsymbol{\theta}) = \mathbf{f}_m^\top \mathbf{p}(\boldsymbol{\theta}) \geqslant 1 - \beta \tag{27}$$

guarantees that when drawing binary vectors from $\mathbf{p}(\boldsymbol{\theta})$, they satisfy $f_m(\mathbf{b}) \leqslant 0$ with probability larger than $1 - \beta$. Here β is a small positive constant capturing the violation probability. Measuring these observables entails counting the frequency at which each one of the logical statements $(f_m(\mathbf{b}) \leqslant 0)$ evaluates as true. Constraint (27) can be made to comply with (16) because

$$\mathbf{f}_m^\top \mathbf{p}(\boldsymbol{\theta}) \geqslant 1 - \beta = (1 - \beta)\mathbf{1}^\top \mathbf{p}(\boldsymbol{\theta})$$

is equivalent to

$$[(1 - \beta)\mathbf{1} - \mathbf{f}_m]^\top \mathbf{p}(\boldsymbol{\theta}) \leqslant 0.$$

The latter constraint can take the form $\mathbf{f}_m^\top \mathbf{p}(\boldsymbol{\theta}) \leqslant 0$ needed in (16) with yet another simple change in $\mathbf{f}_m$'s.

The previous discussion shows that the variational form in (16) allows for dealing with *chance-constrained QCBOs*. This allows us to design stochastic policies from which we can draw binary vectors satisfying quadratic constraints with a prescribed probability. Joint chance constraints can be captured too if we define an observable counting the frequency of multiple logical statements being satisfied simultaneously. Designing policies satisfying (joint) chance constraints may be of interest to application domains such as wireless communications. Clearly, setting $\beta = 0$ provides a heuristic for dealing with the original *deterministic QCBO* in (22).

D. Large-scale LPs on the probability simplex

Lastly, the constrained variational problem in (16) can be used to deal with large-scale LPs over the probability simplex. In this case, vectors $\mathbf{f}_m$'s may bear arbitrary entries, not provided by a particular function. Such large-scale LPs could appear in different application domains, including optimal resource allocation, portfolio optimization, or learning large-scale PMFs from data. In this case, measuring an observable entails VQEC reading out a particular entry of $\{\mathbf{f}_m\}_{m=0}^M$. The entries of these vectors may not correspond to evaluations of quadratic or other functions. Dealing with such LPs is still computationally challenging on a classical computer, and hence, variational quantum solutions could be welcome.

E. Implementation details

Some remarks are now due on some implementation details of VQEC. We start with measuring observables. For the average QCBO setting, observables can be computed by running the VQC for a particular $\boldsymbol{\theta}^t$, measuring its state to get a binary vector $\mathbf{b}_s$, and evaluating the quadratic function $f_m(\mathbf{b}_s)$. The process is repeated S times for the same $\boldsymbol{\theta}$ and observable $F_m(\boldsymbol{\theta}^t)$ is estimated as

$$\hat{F}_m(\boldsymbol{\theta}^t) = \frac{1}{S}\sum_{s=1}^{S} f_m(\mathbf{b}_s). \tag{28}$$

For the deterministic and chance-constrained QCBOs, a similar process estimates the frequency at which each constraint is violated across all measurement shots. For the case of large-scale LPs, measuring $F_m(\boldsymbol{\theta}^t)$ entails reading out the entries of $\mathbf{f}_m$'s indexed by the sampled $\mathbf{b}_s$'s and computing the sample mean of these entries.

Gradients $\{\nabla_{\boldsymbol{\theta}} F_m(\boldsymbol{\theta}^t)\}_{m=0}^M$ can be measured similarly thanks to the parameter shift rule. For a particular $\boldsymbol{\theta}^t$, we need to compute $\hat{F}_m(\boldsymbol{\theta}^t \pm \frac{\pi}{4r}\mathbf{e}_p)$ as discussed earlier, for $p = 1, \ldots, P$. Hence, the complete measuring process has to be repeated $2P$ times.

Interestingly, the parameter shift rule applies to *all* problem types identified in this section, i.e., regardless of whether observables evaluate quadratic, binary, or other functions, or simply read out the coefficient vectors of an LP. It is worth stressing that because all observables are diagonal, they can be measured *simultaneously*. In other words, the number of constraints M does not affect the number of VQC compilations or runs of VQEC as all cost/constraint functions and gradients are estimated classically using the same measurements:

P1) The average QCBO of (4).

P2) The deterministic QCBO in (22) expressed as (4) with constraints $F_m(\boldsymbol{\theta}) \geqslant 1$. Here observables evaluate binary functions by counting the probability of satisfying quadratic constraints.

P3) Chance-constrained QCBOs expressed as (4) with constraints $F_m(\boldsymbol{\theta}) \geqslant 1 - \epsilon$, for small $\epsilon \geqslant 0$. *P2)* is a special case of *P3)* for $\epsilon = 0$.

P4) Binary optimization problems with cost and constraint functions more general than quadratic, which are nonetheless, easy to compute or measure.

P5) Large-scale LPs over the probability simplex as in (25).

F. Discussion

The previous discussion suggests that depending on how vectors $\mathbf{f}_m$'s are defined, the formulation in (16) can handle a wide variety of optimization problems. Vectors $\mathbf{f}_m$ essentially define the objective and constraint observables. Vector $\mathbf{f}_m$ may not necessarily evaluate quadratic functions on the computational basis. Instead, it may evaluate a binary-valued, polynomial, or other analytic function on the computational basis. The presumption is that this function can be efficiently evaluated or measured on a classical (or possibly quantum) computer. Is there any price paid when observables do not correspond to quadratic functions? We do not have an answer to this question but only provide some thoughts. When diagonal observables originate from quadratic functions, the VQC can be tailored to the problem at hand as in QAOA. Nonquadratic diagonal observables, on the other hand, are not amenable to efficient implementations of problem Hamiltonians in the ansatz. Among the applications presented, only the average QCBO seems to be amenable to a QAOA implementation. One may also argue that compared to evaluating logical expressions as in $g_m(\mathbf{b})$, quadratic functions $f_m(\mathbf{b})$ exhibit continuity, which may help VQEC in more accurately estimating observables or optimizing over the landscape induced by the VQC.

It is worth noting that VQC optimization could suffer from the so-termed *barren plateaus*, where gradients vanish exponentially fast in terms of the VQC size (number of qubits, number of gates, and/or circuit depth) [57]. Vanishing gradients may cause gradient-type algorithms to stall. Although such issues may not be easy to encounter in small-sized examples under which current VQCs are currently tested, they could inhibit VQC's adoption to larger real-world-sized problems. It should be stressed that barren plateaus are a challenge for all VQC-based approaches, not VQEC alone. It is expected that approaches for handling barren plateaus would carry over to VQEC. For example, recent results show that judiciously initializing VQC's parameters could help in avoiding barren plateaus [58,59]. By transforming VQC into a circuit without fixed entangling gates, Ref. [59] proves there exist landscapes of VQC parameters, where gradients have large magnitudes regardless of the circuit depth. This result is examined on the hardware efficient ansatz, which is prevalently used in VQC studies due to its suitability for NISQ devices. Albeit finding an initial large gradient does not ensure the algorithm is free from barren plateaus during the training procedure, numerical tests evince that such initialization strategies indeed aid training VQC efficiently [58,59].

IV. PERFORMANCE ANALYSIS

This section analyzes the degradation in performance when the optimization problems of Sec. III are solved in their quantum variational form of (16) instead of the original form in (24), which is an exponentially large LP over the probability simplex. Collecting the linear inequality coefficient vectors $\{\mathbf{f}_m\}_{m=1}^M$ as columns of an $N \times M$ matrix $\mathbf{F}$, the LP in (25) can be written as

$$P^* = \min_{\mathbf{p}\in\mathcal{P}} \mathbf{f}_0^\top \mathbf{p}$$

$$\text{s. to } \mathbf{F}^\top \mathbf{p} \leqslant \mathbf{0}: \quad \boldsymbol{\lambda}. \tag{29}$$

Its dual function can be expressed as

$$D(\boldsymbol{\lambda}) = \min_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(\mathbf{p};\boldsymbol{\lambda}), \tag{30}$$

where the related Lagrangian function is defined as

$$\mathcal{L}(\mathbf{p};\boldsymbol{\lambda}) := (\mathbf{f}_0 + \mathbf{F}\boldsymbol{\lambda})^\top \mathbf{p}. \tag{31}$$

The associated dual problem maximizes the dual function over the dual variables as

$$D^* = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} D(\boldsymbol{\lambda}). \tag{32}$$

If the primal problem is feasible, strong duality holds and yields that $D^* = P^*$.

Let us next consider (29) in its variational quantum form, where variable $\mathbf{p}$ is parameterized by $\boldsymbol{\theta}$. Rephrasing (16), the parameterized primal problem reads as

$$P_\theta^* = \min_{\boldsymbol{\theta}} \mathbf{f}_0^\top \mathbf{p}(\boldsymbol{\theta})$$

$$\text{s. to } \mathbf{F}^\top \mathbf{p}(\boldsymbol{\theta}) \leqslant \mathbf{0}: \quad \boldsymbol{\lambda}. \tag{33}$$

The corresponding dual function is

$$D_\theta(\boldsymbol{\lambda}) := \min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda}) \tag{34}$$

with the related Lagrangian function defined as

$$\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda}) := (\mathbf{f}_0 + \mathbf{F}\boldsymbol{\lambda})^\top \mathbf{p}(\boldsymbol{\theta}). \tag{35}$$

The parameterized dual problem is expressed as

$$D_\theta^* = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} D_\theta(\boldsymbol{\lambda}). \tag{36}$$

The original problem is convex, yet exponentially large. The variational problem is over $\boldsymbol{\theta} \in \mathbb{R}^P$ with $P \ll N$, yet nonconvex. Migrating from the former to the latter lurks two risks. The first risk is that VQEC may not converge, or converge to a local optimum or stationary point. Studying the convergence of (stochastic) PD methods for nonconvex problems is challenging, and is not addressed here. Given the success of stochastic gradient-based methods for nonconvex problems in deep learning, we only investigate the convergence issues of VQEC numerically in Sec. V. The second risk is the possible degradation in optimality, which constitutes the topic of this section.

Although (33) is nonconvex, standard results from duality theory assert that (36) is a convex problem and that weak duality holds so that $D_\theta^* \leqslant P_\theta^*$. The goal of the ensuing analysis is dual: (1) Characterize the duality gap $P_\theta^* - D_\theta^*$. If the duality gap of (33) is zero, the saddle point of the Lagrangian function $\mathcal{L}_\theta$ corresponds to optimal primal/dual solutions for the parameterized problem. Hence, aiming for a saddle point of the Lagrangian is equivalent to seeking the optimal primal/dual variables; and (2) compare D_θ^* with D^*. The parameterized problem is apparently a restriction of the original problem, which entails that $P_\theta^* \geqslant P^*$. Because bounding the difference $P_\theta^* - P^*$ is challenging, we study the difference $D_\theta^* - D^*$ instead. The methods proposed in Sec. II to deal with (25) target at optimizing $\mathcal{L}_\theta$ anyway, which at optimality equals D_θ^*, that is, $D_\theta^* = \mathcal{L}_\theta^*$ by definition of the latter.

Toward the first goal, consider two assumptions.

Assumption 1. Consider a VQC whose state $|\mathbf{x}(\boldsymbol{\theta})\rangle$ induces the parameterized PMF vector $\mathbf{p}(\boldsymbol{\theta})$. The set of PMF vectors that can be produced by all admissible $\boldsymbol{\theta}$'s

$$\mathcal{P}_\theta := \{\mathbf{p} : \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}) \text{ for some} \boldsymbol{\theta}\}$$

is a convex set.

Note that $\mathcal{P}_\theta$ is a set over $\mathbf{p} \in \mathbb{R}^N$, and more specifically $\mathcal{P}_\theta \subseteq \mathcal{P}$. It is not a set over $\boldsymbol{\theta}$.

Assumption 2. The parameterized primal problem in (33) is strictly feasible, i.e., there exist $\boldsymbol{\theta}$ and $\mathbf{s}_\theta > \mathbf{0}$ for which $\mathbf{F}^\top \mathbf{p}(\boldsymbol{\theta}) = -\mathbf{s}_\theta < \mathbf{0}$.

Theorem 1. Under Assumptions 1 and 2, the parameterized problem in (33) has zero duality gap, that is, $D_\theta^* = P_\theta^*$.

Theorem 1 provides two sufficient conditions under which the parameterized problem features zero duality gap; the proof of this theorem as well as all other claims can be found in the Appendixes. Unfortunately, Assumption 1 may not hold for practical VQCs. If it does hold, then not only $D_\theta^* = P_\theta^*$, but also $P_\theta^* = P^*$. To show this, the next lemma is needed.

Lemma 1. The variational PMF $\mathbf{p}(\boldsymbol{\theta})$ induced by the two-local VQC with full entanglement depicted in Fig. 2 can capture all corner points of the probability simplex $\mathcal{P}$. In other words, for $k = 0, \dots, N-1$, there exists a $\boldsymbol{\theta}_k$ for which $\mathbf{p}(\boldsymbol{\theta}_k) = |k\rangle$.

According to Lemma 1, the domain $\mathcal{P}_\theta$ of $\mathbf{p}(\boldsymbol{\theta})$ includes the corners of $\mathcal{P}$. If additionally, set $\mathcal{P}_\theta$ is convex per Assumption 1, then $\mathcal{P}_\theta = \mathcal{P}$ since $\mathcal{P}$ is the convex hull of its corners. Consequently, the variational problem is not a restriction anymore,

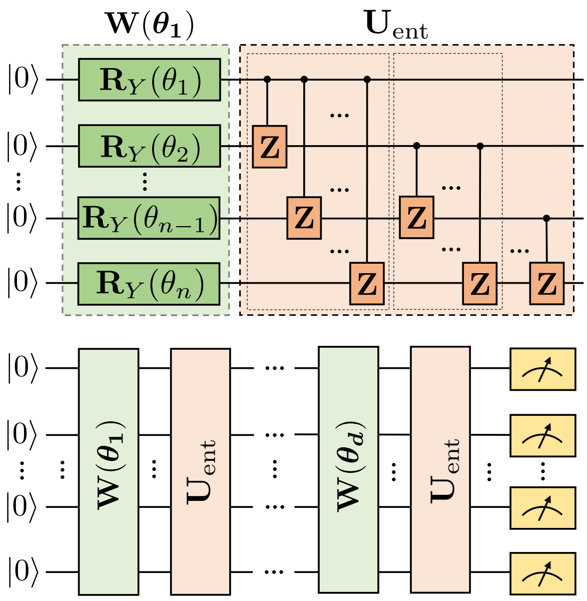
$\mathbf{W}(\boldsymbol{\theta}_1)$
$\mathbf{U}_{\text{ent}}$
$|0\rangle$
$\mathbf{R}_Y(\theta_1)$
$\mathbf{R}_Y(\theta_2)$
$\mathbf{R}_Y(\theta_{n-1})$
$\mathbf{R}_Y(\theta_n)$
Z
$\mathbf{W}(\boldsymbol{\theta}_1)$
$\mathbf{U}_{\text{ent}}$
$\mathbf{W}(\boldsymbol{\theta}_d)$
$\mathbf{U}_{\text{ent}}$

FIG. 2. The two-local VQC operating on an n-qubit system. Top: One layer of the two-local VQC consists of the parameterized block $\mathbf{W}(\boldsymbol{\theta}_1)$ followed by the full entanglement block $\mathbf{U}_{\text{ent}}$. Bottom: A d-layered two-local VQC.

and $P_\theta^* = P^*$. Because Assumption 1 is unlikely to be satisfied by practical VQCs, we aim at characterizing the optimality gap under a milder assumption on the VQC.

Assumption 3. For every $\mathbf{p} \in \mathcal{P}$, there exists a $\boldsymbol{\theta}$ for which $\|\mathbf{p} - \mathbf{p}(\boldsymbol{\theta})\|_1 \leqslant \epsilon$ for some $\epsilon > 0$.

Assumption 3 assumes that the domain of $\mathbf{p}(\boldsymbol{\theta})$ is sufficiently dense so that any PMF vector can be approximated under the l_1 norm within accuracy ϵ. This assumption resembles the *universal approximation property* established for and widely used in deep neural networks; see, e.g., [60, Sec. 6.4.1]. Although the observables defined over asymptotically rich VQCs have been shown to be universal function approximators [61], the argument may not be trivially extendable to the PMF setup, where a VQC aims to approximate an N-long vector rather than a univariate function. Because $\|\mathbf{p} - \mathbf{p}(\boldsymbol{\theta})\|_1 \leqslant \|\mathbf{p}\|_1 + \|\mathbf{p}(\boldsymbol{\theta})\|_1 \leqslant 2$ due to the triangle inequality and PMF vectors lying on $\mathcal{P}$, the accuracy is obviously upper bounded by 2.

An additional assumption on a perturbed version of the original primal LP of (29) will be also needed.

Assumption 4. Consider the linear program

$$\begin{aligned}\tilde{P}^* = \min_{\mathbf{p}\in\mathcal{P}} \ & \mathbf{f}_0^\top \mathbf{p} \\ \text{s. to } & \mathbf{F}^\top \mathbf{p} \leqslant -\epsilon L \mathbf{1} : \quad \boldsymbol{\lambda},\end{aligned} \tag{37}$$

where ϵ has been defined under Assumption 3. Let us extend $\mathbf{F}$ to an $N \times (M+1)$ matrix $\bar{\mathbf{F}}$ by adding $\mathbf{f}_0$ as $\bar{\mathbf{F}}$'s first column and define

$$L := \max_{i=1:N} \sum_{j=1}^{M+1} |\bar{\mathbf{F}}_{ij}|, \tag{38}$$

where $\bar{\mathbf{F}}_{ij}$ is the (i, j) entry of $\bar{\mathbf{F}}$. The assumption is that (37) is strictly feasible. In other words, there exist $\hat{\mathbf{p}} \in \mathcal{P}$ and $s_0 > 0$ satisfying

$$\mathbf{F}^\top \hat{\mathbf{p}} \leqslant -\epsilon L \mathbf{1} - s_0 \mathbf{1}. \tag{39}$$

Using Assumption 2 and inspired by the proofing procedure of [34], we next bound the degradation in performance when surrogating the original exponentially large LP by its variational form.

Theorem 2. Under Assumptions 2, 3, and 4, the optimal dual value of the variational quantum problem satisfies

$$D^* \leqslant D_\theta^* \leqslant D^* + \epsilon L \max\{1, \|\tilde{\boldsymbol{\lambda}}\|_1\}, \tag{40}$$

where $\tilde{\boldsymbol{\lambda}}$ is the vector of optimal Lagrange multipliers for the perturbed primal problem in (37). The norm $\|\tilde{\boldsymbol{\lambda}}\|_1$ is upper bounded as

$$\|\tilde{\boldsymbol{\lambda}}\|_1 \leqslant \frac{\mathbf{f}_0^\top \hat{\mathbf{p}} - P^*}{s_0}. \tag{41}$$

Theorem 2 predicates that the optimality gap in the dual domain $D_\theta^* - D^*$ is affected by two factors: (1) the accuracy ϵ within which the VQC can approximate PMF vectors; and (2) the sensitivity of the primal LP to ϵ perturbations in the cost and constraints.

Remark 2. Assumptions 2 and 4 presume that both the original and the perturbed problems are strictly feasible. Such a requirement automatically excludes problems with equality constraints. Variational problems with equality constraints can be handled by Theorem 2 only if equality constraints are relaxed to double-side inequality constraints. Despite this limitation in performance analysis, the algorithms of Sec. II remain directly applicable to equality-constrained variational problems.

V. NUMERICAL TESTS

VQEC was numerically evaluated under three setups, which are representative of the application examples presented in Sec. III.

S1) This setup solves the average QCBO of Sec. III B on the constrained MaxCut problem. A weighted $n = 14$-vertex graph was randomly generated. To include constraints, we randomly sampled seven pairs of vertices and added connection specifications so the problem was feasible. The number of qubits used is $n = 14$.

S2) This setup deals with the deterministic QCBO of (22) using the formulation of Sec. III C, applied to the same instance as in S1 using $n = 14$ qubits.

S3) This setup solves an LP over the probability simplex like the one in (25) as described in Sec. III D. Given the restrictions of quantum simulators, the problem dimension was $N = 256$ with $M = 3$ constraints. Vectors $\{\mathbf{f}_m\}_{m=0}^3$ were stacked into a 256×4 matrix $\mathbf{F}$, whose entries were randomly generated from the standard normal distribution. The number of qubits needed is $n = 8$.

All numerical tests were performed on quantum simulators from IBM's Qiskit [62]. Simulation scripts were written in Python. The two-local VQC from Qiskit was used across tests. As illustrated in Fig. 2, each layer of the VQC consists of a parameterized block implemented via single-qubit $\mathbf{R}_Y(\theta_p)$ gates applied to each qubit and a full entanglement block implemented via CZ gates between all pairs of qubits. The

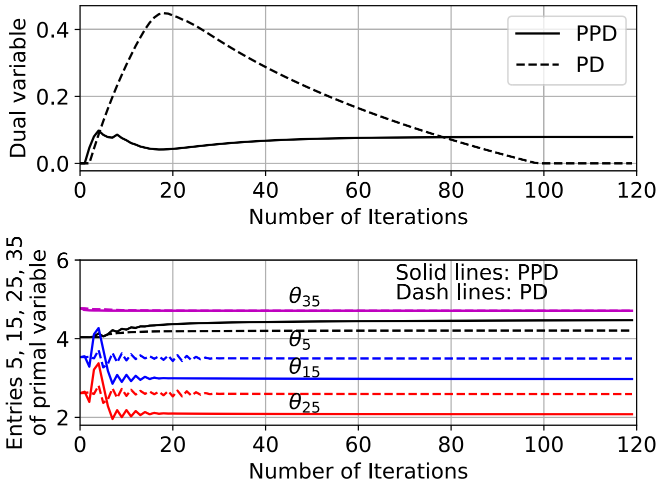

FIG. 3. Convergence of primal/dual variables for PD and PPD methods over the number of primal/dual iterations under setup S1 using a quantum state simulator (no measurement noise). Top: Convergence of dual variable λ. Because the constraint is active, the optimal λ is nonzero. Bottom: Convergence of entries 5, 15, 25, 35 of the primal variable $\boldsymbol{\theta}$.

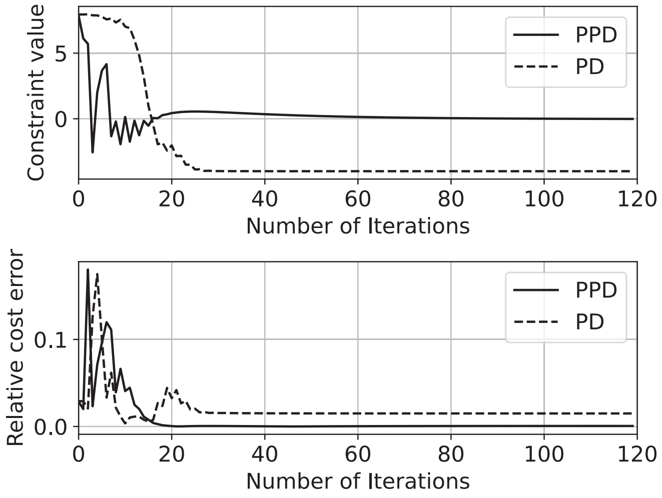

FIG. 4. Convergence of constraint function value $F_1(\boldsymbol{\theta}^t)$ (top) and relative cost error $|(P_\theta^t - P^*)/P^*|$ (bottom) for the PD and PPD methods over the number of primal-dual iterations under setup S1 using a quantum state simulator (no measurement noise).

number of parameterized blocks of the VQC is referred to as the *circuit depth* d. Clearly, the length P of $\boldsymbol{\theta}$ relates to the problem size n and the circuit depth d as $P = dn$. The PPD/PD iterations were deemed to have converged when $\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t-1}\|_2/\|\boldsymbol{\theta}^{t-1}\|_2 \leqslant \epsilon$ for $\epsilon = 1 \times 10^{-5}$. The initial primal vector $\boldsymbol{\theta}^0$ was drawn uniformly within $[0, 2\pi]$ using a fixed seed across tests, while $\boldsymbol{\lambda}^0 = \mathbf{0}$.

Regarding setup S1, we first compared the convergence properties of VQEC. As noted in Sec. II, PPD updates with decreasing step size converge to optimal solutions for convex problems [37,38,49]. Given no analogous result for nonconvex problems, we investigated the convergence of VQEC numerically. To eliminate stochasticity due to measurement noise, the initial test utilized the `statevector_simulator` quantum simulator from IBM's Qiskit. The two-local VQC used in this test has two full layers and an additional parameterized block yielding a depth of $d = 3$. In this case, the test implements the exact PPD/PD methods rather than their stochastic variants. Step sizes followed a time-decreasing rule as $\mu_\theta^t = 1.5/t$, $\mu_\lambda^t = 0.1/(t + 15)$, while the additional ones for PPD were set to $\nu_\theta = \nu_\lambda = 0.05$.

For comparison, we also solved the original average QCBO in (25) to optimality using the Gurobi solver under the YALMIP environment [63,64]. The optimal cost and constraint function values were used as references to verify the feasibility and optimality of the variational solutions found by VQEC. Due to sign invariance, MaxCut solutions come in pairs. The deterministic MaxCut problem has a unique pair of solutions. Nonetheless, the average QCBO returns a PMF vector with four nonzero entries. If we convert these entries to indices of canonical vectors, two of these canonical vectors are optimal and two are infeasible for the original deterministic MaxCut with constraints. Hence, this particular problem instance is nonideal as the average QCBO does not solve the deterministic QCBO.

Figure 3 depicts the convergence of primal/dual variables using PD and PPD (VQEC). The constraint is active (i.e., satisfied with equality) for this studied instance. Both PD and PPD converged after roughly 500 iterations. However, as shown in Fig. 3, the dual variable of PD converged to 0, whereas the one of PPD converged near 0.1. The VQC parameters for the two methods converged to different values too. Regardless of the converged $\boldsymbol{\theta}$, how did the trained VQCs perform in terms of feasibility and optimality? Figure 4 depicts the convergence of constraint function $F_1(\boldsymbol{\theta}^t)$ and relative cost error $|(P_\theta^t - P^*)/P^*|$, where P^* is the optimal cost found by Gurobi. Evidently, both the constraint function and the cost error converged to zero using PPD, whereas the ones obtained by PD have greatly deviated from zero. Although PD found a feasible solution, that solution yielded suboptimal cost. On the other hand, PPD found the optimal solution in this case. This test evinces that the employed VQC is capable of finding a $\mathbf{p}(\boldsymbol{\theta})$ that coincides with the optimal solution of (25), and also that PPD can converge to the saddle point of the Lagrangian function of (35). It should be emphasized that PPD is not guaranteed to converge to a global optimum of the nonconvex variational problem in general. Nonetheless, given the advantage demonstrated by the previous and similar tests, all subsequent tests use the PPD (rather than the PD) method.

The previous test utilized `statevector_simulator`, which is equivalent to measuring observables using an infinite number of measurement shots S in (28). In practice, observables are measured using a finite S giving rise to *stochastic* PPD updates. To assess the effect of using a finite number of measurement shots, we ran PPD iterates under S1 with S taking values in $\{1, 25, 50, \infty\}$. The `aer_simulator` was used for finite values of S. The circuit depth was set to $d = 3$. Figure 5 shows the convergence of the constraint value, and relative cost error of the proposed method compared to the exact cost value solved by the Gurobi solver. For each finite S, PPD iterations were repeated eight times to account for stochasticity. As can be seen from Fig. 5, the constraint

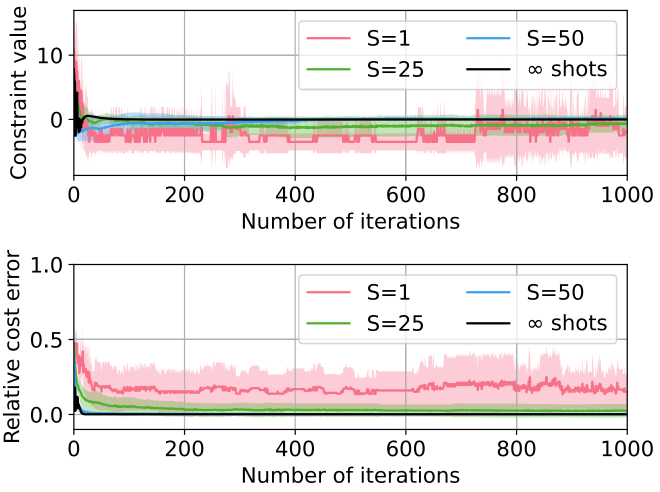

FIG. 5. Convergence of constraint function value $F_1(\boldsymbol{\theta}^t)$ (top) and relative cost error $|(P_\theta^t - P^*)/P^*|$ (bottom) for PPD over the number of primal-dual iterations under setup S1 using different values of measurement shots S. For $S \in \{1, 25, 50\}$, PPD was repeated eight times to account for the randomness in iterations. The plots display confidence intervals within one standard deviation around the mean per iteration.

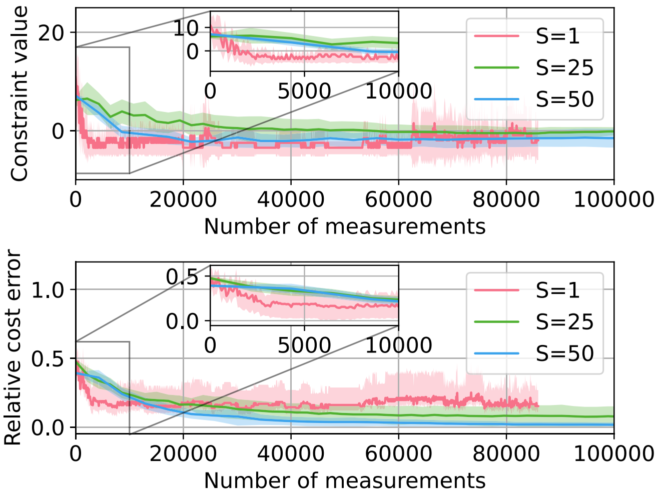

FIG. 6. Convergence of constraint function value $F_1(\boldsymbol{\theta}^t)$ (top) and relative cost error $|(P_\theta^t - P^*)/P^*|$ (bottom) for PPD over the total number of measurement shots while solving setup S1 for different S. For $S \in \{1, 25, 50\}$, PPD was repeated eight times. The plots display confidence intervals within one standard deviation around the mean per iteration.

values and cost errors given by $S = 1$ are highly variant and away from zero. For increasing S, the constraint value and cost error not only move closer to zero but also exhibit less variance across iteration instantiations. Compared to the state estimator (infinite S), PPD with $S = 50$ is competent to sample near-optimal solutions under S1. This test demonstrates the improvement in convergence of VQEC for increasing S. Nevertheless, the comparison across S may not be fair as a PPD iteration with $S = 50$ requires running the VQC 50 times more than a PPD iteration with $S = 1$. Recall also that each PPD iteration requires $(2P + 2)S$ measurement shots.

To study VQEC's convergence over the total VQC runs, Fig. 6 plots the convergence of the constraint value and relative cost error over the total number of measurement shots. As highlighted in the two insets, for the first 10 000 shots, the constraint value and cost error of the one-shot method moved to the zero line faster than the ones of the 25-shot and 50-shot methods did. Eventually, however, the constraint value and the cost error of the one-shot method remained away from zero, even after 86 000 shots. On the other hand, the 25- and 50-shot iterations approached zero. The solution generated by the 50-shot method in particular is feasible and its cost error is near-optimal after 100 000 measurement shots. Given these observations, it may be meaningful to use $S = 1$ at early iterations and deliberately increase S as time goes by, to save the total number of measurement shots without compromising optimality. This strategy was also suggested for standard VQE without constraints in [46,65]. Since $S = 50$ attained a good trade-off between optimality and total number of shots, it was used for all subsequent tests.

Moving on to setup S2, the goal here is to solve a deterministic QCBO (constrained MaxCut) via VQEC. We do so using the average QCBO and the deterministic QCBO, both solved repeatedly eight times for different S. The circuit depth was kept to $d = 3$. Step sizes of VQEC while solving the average QCBO were kept the same as in setup S1. For the deterministic QCBO, step sizes of primal/dual updates were selected as $\mu_\theta^t = 12/(t + 10)$ and $\mu_\lambda^t = 4/(t + 15)$, while two additional step sizes of perturbed updates were set to $\nu_{\boldsymbol{\theta}} = 1$ and $\nu_{\boldsymbol{\lambda}} = 1.5$. Recall that the two problems differ in how vectors $\mathbf{f}_m$ are computed. The question is whether the binary vectors $\mathbf{b}$ drawn from the obtained $\mathbf{p}(\boldsymbol{\theta})$ solve the original deterministic QCBO. The constrained MaxCut has two optimal solutions. We measure the probability of success by summing up the two entries of $\mathbf{p}(\boldsymbol{\theta})$ associated with the two optimal solutions. For each S, we define as the final probability of success the worst-case probability of success across the eight independent runs. After training the VQC using $S \in \{1, 25, 50\}$ to get $\boldsymbol{\theta}$, the obtained $\mathbf{p}(\boldsymbol{\theta})$ was read out using the `statevector_simulator`. Table I reports these probabilities. For $S = 1$, the probability of success of both problems is 0, which is justifiable as the method has not converged. For $S = 25$ and 50, the variational deterministic QCBO achieved a very good probability of success compared to the variational stochastic QCBO. In other words, the former can serve as an excellent heuristic to provide solutions to QCBOs with high probability.

For a more thorough evaluation of VQEC in solving the deterministic QCBO, we solved S2 over 10 different constrained MaxCut instances. The optimal value of the Lagrangian function in (35) found by VQEC was compared to the exact one in

TABLE I. Worst-case probability of obtaining optimal solutions when solving the original QCBO of (22) through the variational average QCBO and the variational deterministic QCBO.

Variational problem	Probability of success		
	$S = 1$	$S = 25$	$S = 50$
Average QCBO	0.0000	0.5240	0.5899
Deterministic QCBO	0.0000	0.9940	0.9704

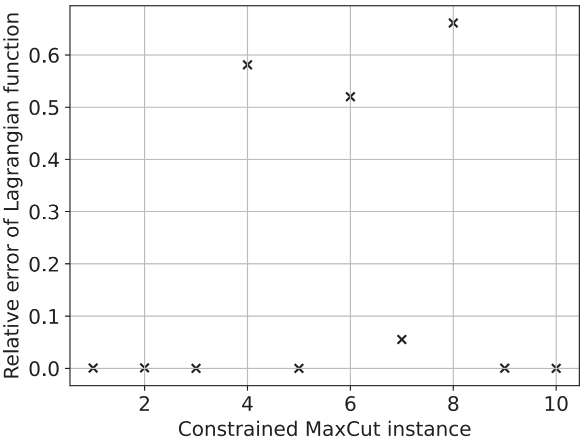

FIG. 7. Relative error of the Lagrangian function after convergence while solving the deterministic QCBO for $S = 50$ over 10 constrained MaxCut instances. The step sizes of VQEC and the simulator seed were fixed across instances.

(31) obtained by Gurobi. The number of measurement shots was fixed to $S = 50$, and the circuit depth was set to $d = 3$. Step sizes of VQEC were set as in the previous test. Figure 7 shows the relative error in the Lagrangian function value $|(\mathcal{L}(\boldsymbol{\theta}^t;\boldsymbol{\lambda}^t) - \mathcal{L}^*)/\mathcal{L}^*|$ over 10 different constrained MaxCut instances, where $\mathcal{L}^*$ is the exact value of the Lagrangian function found by Gurobi. Although VQEC managed to find optimal or near-optimal solutions for several instances, there exist instances $\{4, 6, 8\}$ with large relative errors.

As a sanity check, we repeated the following numerical tests over instances of the constrained MaxCut problem. We solved the related deterministic QCBO by Gurobi and found the minimizer. We then initialized $\boldsymbol{\theta}$ so that the VQC output coincides with the minimizer. Such initialization can be performed using the procedure described in the proof of Lemma 1 found in the Appendixes. We consequently ran PPD and observed that VQEC did not drift away from the optimal $\boldsymbol{\theta}$.

To assess the effect of the circuit depth (dimension of $\boldsymbol{\theta}$) on the performance of VQEC, we solved the variational deterministic QCBO under S2 using circuit depths $d \in \{2, 3, 4\}$. The number of measurement shots was fixed to $S = 50$ and each test was repeated eight times. Figure 8 illustrates the convergence of the constraint value and the relative cost error obtained by PPD for different values of d. Clearly, the constraint value and the cost error obtained with $d = 2$ deviate substantially from the ideal zero lines. On the contrary, the VQCs with $d = 3$ and $d = 4$ were able to solve the problem to optimality by increasing the expressibility of the VQC. This observation agrees with recent theoretical findings on the expressibility of VQCs [61]. It is important to remark that $d = 4$ seems to be an over-parameterization of the problem and takes longer to converge. Similar observations were made in [66]. It is also worth stressing that even for $d = 4$, the total number of parameters $P = dn = 56$ is much smaller than the size of the original primal LP, that is, $N = 2^{14} = 16\,384$. This test signifies that low-depth VQCs might be suboptimal, whereas deeper VQCs may take longer to converge.

Under setup S3, we evaluated VQEC in solving the large-scale LP in (25). The step sizes of VQEC were set as $\mu_{\theta}^t = \mu_{\lambda}^t = 0.02 \times 0.999^t$ and $\nu_{\boldsymbol{\theta}} = \nu_{\boldsymbol{\lambda}} = 3$. The number of measurement shots S was set to 150, and the circuit depth d was set to 3. The need to increase S for this setup might be caused by the randomly generated values of $\{\mathbf{f}_m\}_{m=0}^{M}$, which did not correspond to quadratic functions anymore. VQEC was run eight times with different values of the simulator seed. The confidence intervals within one standard deviation around the mean per iteration are displayed in Fig. 9. Dual variables converged after 2000 iterations. While λ_2 converged exactly to 0 for all runs, λ_1 and λ_3 varied slightly for different runs. Five randomly chosen entries of the primal variable are shown in Fig. 10. All entries converged after 2000 iterations. While entries θ_1, θ_5, and θ_{12} converged to roughly the same values

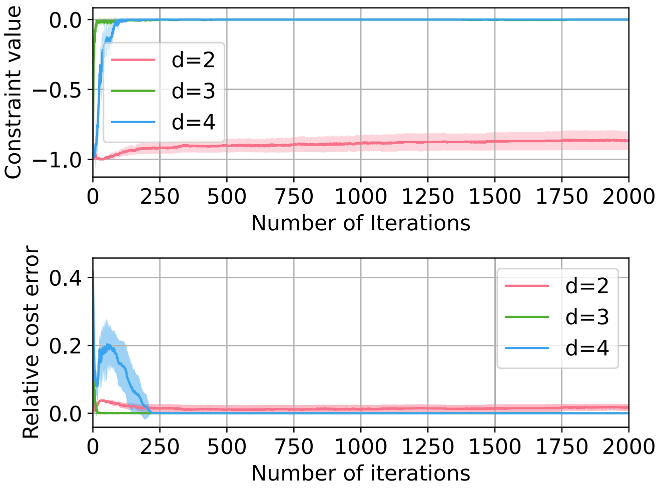

FIG. 8. Convergence of constraint function value $F_1(\boldsymbol{\theta}^t)$ (top) and relative cost error $|(P_\theta^t - P^*)/P^*|$ (bottom) for PPD over the number of PPD iterations while solving setup S2 for $S = 50$ and varying values of circuit depths d. The plots display confidence intervals within one standard deviation around the mean per iteration, computed over eight runs.

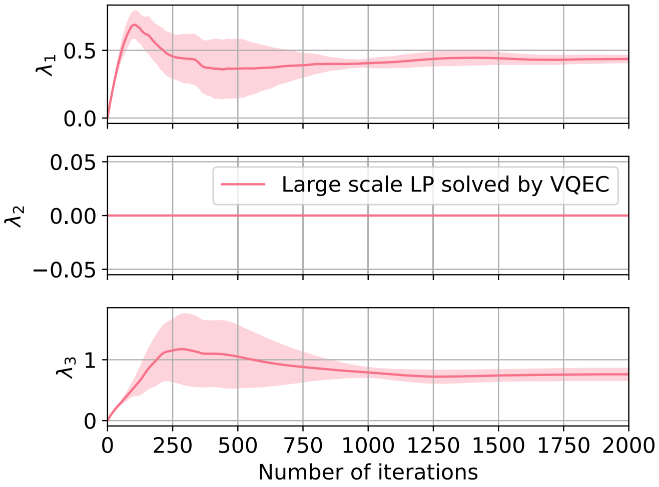

FIG. 9. Convergence of entries of the dual variable $\boldsymbol{\lambda}$ over the number of primal/dual iterations under setup S3. The VQEC was repeated eight times. The plots display confidence intervals within one standard deviation around the mean per iteration, computed over eight runs.

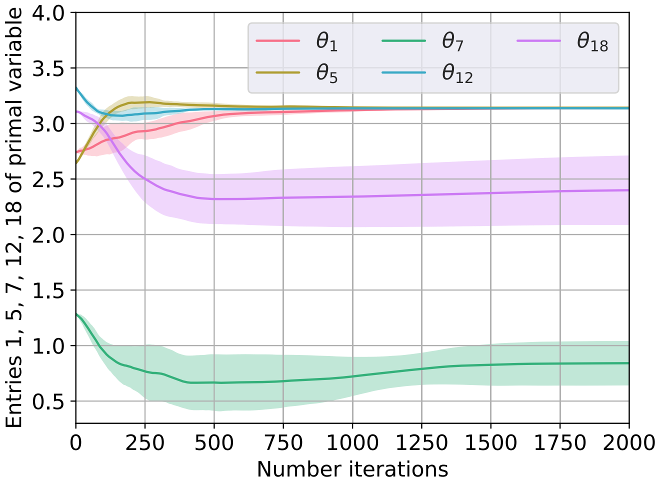

FIG. 10. Convergence of entries 1, 5, 7, 12, 18 of the primal variable $\boldsymbol{\theta}$ over the number of primal/dual iterations under setup S3. The VQEC was repeated eight times. The plots display confidence intervals within one standard deviation around the mean per iteration, computed over eight runs.

over eight runs, entries θ_7 and θ_{18} varied quite significantly for different runs. To validate the feasibility of the obtained solution, the large-scale LP in (25) was also solved exactly over the exponentially large vector $\mathbf{p}$ by Gurobi. Figure 11 compares the constraint values found by the proposed method and the reference values by Gurobi. As shown in Fig. 11, the solution obtained by VQEC satisfied all three constraints. The relative cost error $|(P_\theta^t - P^*)/P^*|$ is shown in Fig. 12, where P^* is the exact cost value obtained by Gurobi. As illustrated in Fig. 12, the relative cost error attained by VQEC converged to around 10% after 2000 iterations. The variance of the relative cost error over eight runs is very low after convergence. This corroborates that VQEC can generate near-optimal solutions for the large-scale LP in (25).

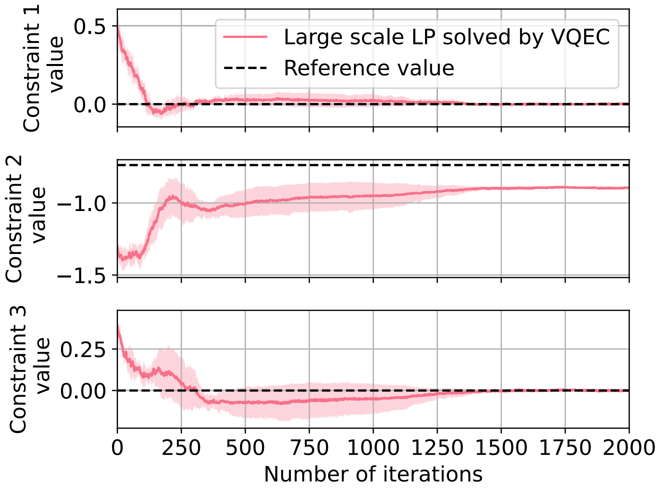

FIG. 11. Comparing the constraint values attained by the VQEC and the reference values solved exactly by Gurobi under setup S3. The VQEC was repeated eight times. The plots display confidence intervals within one standard deviation around the mean per iteration, computed over eight runs.

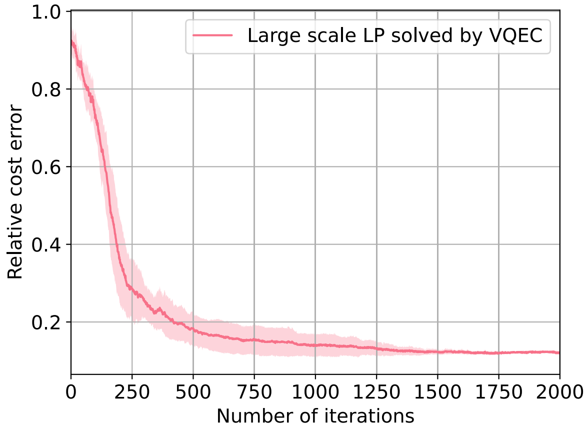

FIG. 12. Convergence of the relative cost error under setup S3. The VQEC was repeated eight times. The plots display confidence intervals within one standard deviation around the mean per iteration, computed over eight runs.

The optimality gap observed in the previous test reveals that the frequency spectrum of the employed VQC might not be rich enough [61]. To expand the frequency spectrum, we repeated each VQC parameter of the previous test three times and investigated its effect on the performance of VQEC. Setup S3 was solved over 10 different large-scale LP instances. The circuit depth d and number of measurement shots S were kept as in the previous test. The step sizes of primal/dual updates were chosen as $\mu_\theta^t = 0.05 \times 0.999^t$, $\mu_\lambda^t = 0.002 \times 0.999^t$, and $\nu_{\boldsymbol{\theta}} = \nu_{\boldsymbol{\lambda}} = 1$. These step sizes and the simulator seed were fixed across the ten instances. Figure 13 compares the relative errors of the Lagrangian function of VQEC with $L = 3$ and $L = 1$, where L is the number of repeated VQC. The method was deemed to have converged after 500 iterations. As shown in Fig. 13, for nine instances, VQEC with $L = 3$ performed much better than the one with $L1$. Among those nine cases, eight instances attained relative errors below 20%. This test shows that repeating parameters enhances the performance of VQEC.

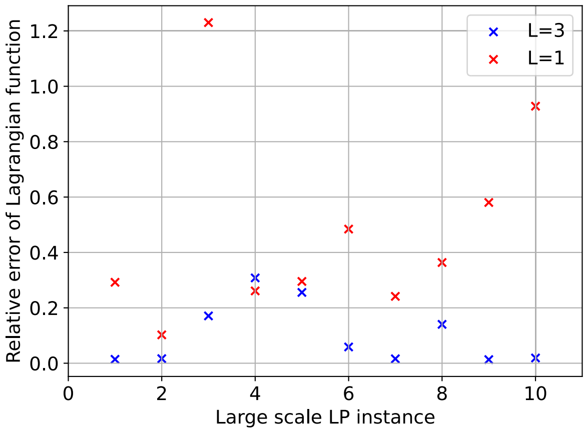

FIG. 13. Relative error of the Lagrangian function after 500 iterations while solving setup S3 for $S = 150$ over ten large-scale LP instances. The step sizes of VQEC and the simulator seed were fixed across instances.

VI. CONCLUSIONS

This work has developed, analyzed, and evaluated an algorithm for handling optimization problems with constraints using a variational quantum approach. The proposed VQEC deals with constraints in the dual domain of the parameterized variational quantum problem. Primal and dual variables are iteratively updated on a classical computer via stochastic PPD iterates. Compared to VQE and for problems with diagonal observables, VQEC entails insignificant computational overhead as the cost and constraint observables are measured simultaneously. VQEC has been applied to diverse problems defined over diagonal observables, including binary programs with constraints, finding optimal stochastic policies to draw binary vectors satisfying average and chance constraints, learning probability mass functions, and solving large-scale LPs over the probability simplex. The possible performance degradation by solving a problem in its quantum variational form *vis-à-vis* its original form has been characterized. Extensive numerical tests using IBM's quantum simulator have corroborated that (1) the PPD method performs remarkably better than the PD method with only one additional measurement shot per primal/dual iteration; (2) VQEC with finite numbers of measurement shots can yield meaningful solutions to QCBOs, stochastic QCBOs, and large-scale LPs over the probability simplex; and (3) the solution quality and/or convergence rate of VQEC can be affected by under/over-parameterized VQCs.

In a nutshell, VQEC provides a principled solution to the practically relevant task of incorporating constraints into variational quantum approaches. It can thus patently expand the application domain of NISQ computers. Capitalizing on the promising results of this work, we are currently working toward several exciting research directions, such as (1) Coping with nondiagonal observables is highly desirable as it clearly broadens the applicability of VQEC from binary and linear programs to conic programs, including quadratically constrained quadratic programs (QCQPs) and semidefinite programs (SDPs). Dealing with mixed-integer programs involving binary and continuous variables is on current focus too. (2) The VQCs considered thus far have been confined to be parameterized solely by $\boldsymbol{\theta}$ to produce state $|\mathbf{x}(\boldsymbol{\theta})\rangle = \mathbf{S}(\boldsymbol{\theta})\,|0\rangle_n$. Nonetheless, in a quantum machine learning setting, a VQC may also encode *data (features)* $\{\mathbf{z}_t\}_{t=1}^T$ in the form of a data-embedding mechanism as in $\mathbf{U}(\boldsymbol{\theta};\mathbf{z}_t) = \mathbf{S}(\boldsymbol{\theta})\mathbf{W}(\mathbf{z}_t)$ or other forms. Measuring the quantum state $|\mathbf{x}(\boldsymbol{\theta};\mathbf{z}_t)\rangle = \mathbf{U}(\boldsymbol{\theta};\mathbf{z}_t)\,|0\rangle_n$ and applying VQEC to proper loss functions, can tackle pertinent (un)supervised quantum machine learning tasks. (3) In light of the performance analysis of Sec. IV, it is vital to further investigate Assumption 3 on the accuracy with which VQCs can approximate PMFs and other optimization objects. (4) Extending our performance analysis to other optimization classes and contrasting VQEC to stochastic classical counterparts.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation under Grants No. 1751085 and No. 2412947.

APPENDIX A: PROOF OF THEOREM 1

To establish zero duality gap, it suffices to show that the perturbation function associated with the parameterized primal problem is convex [67]. This perturbation function is defined by modifying the right-hand side of the linear inequality constraints of (33) as

$$P_\theta = \min_{\boldsymbol{\theta}}\ \mathbf{f}_0^\top \mathbf{p}(\boldsymbol{\theta}) \quad \text{s. to } \mathbf{F}^\top \mathbf{p}(\boldsymbol{\theta}) \leqslant \boldsymbol{\delta}, \tag{A1}$$

where $\boldsymbol{\delta} > -\mathbf{s}_\theta$ with $\mathbf{s}_\theta$ from Assumption 2. Consider two points $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$. Suppose solving (A1) for $\boldsymbol{\delta}_1$ yields the minimizer $\boldsymbol{\theta}_1$ generating $\mathbf{p}_1 = \mathbf{p}(\boldsymbol{\theta}_1)$. Likewise, solving (A1) for $\boldsymbol{\delta}_2$ yields the minimizer $\boldsymbol{\theta}_2$ generating $\mathbf{p}_2 = \mathbf{p}(\boldsymbol{\theta}_2)$. Under Assumption 1, any vector $\mathbf{p}_\alpha = \alpha\mathbf{p}_1 + (1-\alpha)\mathbf{p}_2$ for $\alpha \in [0, 1]$ belongs to the set $\mathcal{P}_\theta$ of admissible PMF vectors. That means that there exists a $\boldsymbol{\theta}_\alpha$ for which $\mathbf{p}_\alpha = \mathbf{p}(\boldsymbol{\theta}_\alpha)$. Vector $\mathbf{p}_\alpha$ is feasible for the perturbed problem if the latter is perturbed by $\boldsymbol{\delta}_\alpha = \alpha\boldsymbol{\delta}_1 + (1-\alpha)\boldsymbol{\delta}_2$ because

$$\mathbf{F}^\top\mathbf{p}_\alpha = \alpha\mathbf{F}^\top\mathbf{p}_1 + (1-\alpha)\mathbf{F}^\top\mathbf{p}_2 \leqslant \alpha\boldsymbol{\delta}_1 + (1-\alpha)\boldsymbol{\delta}_2 = \boldsymbol{\delta}_\alpha.$$

Since $\mathbf{p}_\alpha$ is feasible for the perturbed problem for $\boldsymbol{\delta}_\alpha$, it provides an upper bound for the perturbation function $P_\theta(\boldsymbol{\delta}_\alpha)$ as

$$\begin{aligned} P_\theta(\boldsymbol{\delta}_\alpha) &\leqslant \mathbf{f}_0^\top\mathbf{p}_\alpha = \alpha\mathbf{f}_0^\top\mathbf{p}_1 + (1-\alpha)\mathbf{f}_0^\top\mathbf{p}_2 \\ &= \alpha P_\theta(\boldsymbol{\delta}_1) + (1-\alpha)P_\theta(\boldsymbol{\delta}_2). \end{aligned}$$

This establishes that $P_\theta(\boldsymbol{\delta})$ is a convex function of $\boldsymbol{\delta}$, which in turn implies that the parameterized problem has zero duality gap.

APPENDIX B: PROOF OF LEMMA 1

Consider first a single layer of the VQC shown in Fig. 2. Suppose the input to this layer is $|0\rangle_n$. The layer consists of a parameterized block and an entanglement block. The parameterized block applies gate

$$\mathbf{R}_Y(2\theta_p) = \exp\left(-i\theta_p\mathbf{Y}\right) = \begin{bmatrix}\cos(\theta_p) & -\sin(\theta_p)\\ \sin(\theta_p) & \cos(\theta_p)\end{bmatrix}$$

on qubit p for $p = 1,\ldots,n$, where $\mathbf{Y}$ is the matrix representing the Pauli Y gate. If $|\boldsymbol{\psi}\rangle$ denotes the state after the parameterized block, its pth qubit is

$$|\boldsymbol{\psi}\rangle_p = \mathbf{R}_Y(2\theta_p)\,|0\rangle = \cos(\theta_p)\,|0\rangle + \sin(\theta_p)\,|1\rangle\,,\ \ p = 1:n.$$

The entanglement block includes controlled-Z (CZ) gates between all pairs of qubits. Since the roles of the control and target qubits are interchangeable for the CZ gate, we designate control and target qubits as shown in Fig. 2 without loss of generality. Let $|\boldsymbol{\phi}\rangle$ be the state after the entanglement block. As $|\boldsymbol{\psi}\rangle_1$ is not subjected to any control gate, qubit 1 of $|\boldsymbol{\phi}\rangle$ is

$$|\boldsymbol{\phi}\rangle_1 = |\boldsymbol{\psi}\rangle_1 = \cos(\theta_1)\,|0\rangle + \sin(\theta_1)\,|1\rangle\,. \tag{B1}$$

By drawing θ_1 from $\{0, \pi/2\}$, qubit $|\boldsymbol{\phi}\rangle_1$ can take any value in $\{|0\rangle\,, |1\rangle\}$.

The entanglement block shown in Fig. 2 consists of $(n-1)$ subblocks. The first subblock comprises $(n-1)$ CZ gates, in which qubit 1 controls qubits 2 to n. The target qubits after this subblock are denoted as $|\boldsymbol{\phi}\rangle_p^1$ for $p = 2,\ldots,n$. The second subblock comprises $(n-2)$ CZ gates, in which qubit

2 controls qubits 3 to n. The target qubits after this subblock are denoted as $|\boldsymbol{\phi}\rangle_p^2$ for $p = 3, \ldots, n$. Subsequent subblocks expand similarly up to subblock $(n-1)$.

Consider the output of the first subblock. If $|\boldsymbol{\phi}\rangle_1 = |0\rangle$, then $|\boldsymbol{\phi}\rangle_p^1 = |\boldsymbol{\psi}\rangle_p = \cos(\theta_p)\,|0\rangle + \sin(\theta_p)\,|1\rangle$. If $|\boldsymbol{\phi}\rangle_1 = |1\rangle$, then $|\boldsymbol{\phi}\rangle_p^1 = \mathbf{Z}\,|\boldsymbol{\psi}\rangle_p = \cos(\theta_p)\,|0\rangle - \sin(\theta_p)\,|1\rangle$. Since $|\boldsymbol{\phi}\rangle_1 = |0\rangle$ implies $\sin(\theta_1) = 0$, and $|\boldsymbol{\phi}\rangle_1 = |1\rangle$ implies $\sin(\theta_1) = 1$, the output of the first subblock can be compactly expressed as

$$|\boldsymbol{\phi}\rangle_p^1 = \cos(\theta_p)\,|0\rangle + (-1)^{\sin(\theta_1)}\sin(\theta_p)\,|1\rangle\,,\quad p = 2:n.$$

Because qubit 2 is not controlled by any more qubits, its state can be finalized here as

$$|\boldsymbol{\phi}\rangle_2 = |\boldsymbol{\phi}\rangle_2^1 = \cos(\theta_2)\,|0\rangle + (-1)^{\sin(\theta_1)}\sin(\theta_2)\,|1\rangle\,. \tag{B2}$$

It is now easy to verify that if we want to set $|\boldsymbol{\phi}\rangle_2 = |0\rangle$, we can simply set $\theta_2 = 0$. Otherwise, that is, to set $|\boldsymbol{\phi}\rangle_2 = |1\rangle$, parameter θ_2 is selected between $\{\pi/2, 3\pi/2\}$ depending on the value of θ_1. Specifically, if $\theta_1 = 0$, set $\theta_2 = \pi/2$; and if $\theta_1 = \pi/2$, set $\theta_2 = 3\pi/2$ to get the proper sign.

Consider now the second subblock, where CZ gates are controlled by $|\boldsymbol{\phi}\rangle_2$. If $|\boldsymbol{\phi}\rangle_2 = |0\rangle$, then $|\boldsymbol{\phi}\rangle_p^2 = |\boldsymbol{\phi}\rangle_p^1 = \cos(\theta_p)\,|0\rangle + (-1)^{\sin(\theta_1)}\sin(\theta_p)\,|1\rangle$. If $|\boldsymbol{\phi}\rangle_2 = |1\rangle$, then $|\boldsymbol{\phi}\rangle_p^2 = \mathbf{Z}\,|\boldsymbol{\phi}\rangle_p^1 = \cos(\theta_p)\,|0\rangle - (-1)^{\sin(\theta_1)}\sin(\theta_p)\,|1\rangle$. The two cases can be compactly expressed as

$$|\boldsymbol{\phi}\rangle_p^2 = \cos(\theta_p)\,|0\rangle + (-1)^{\sin(\theta_1)+\sin(\theta_2)}\sin(\theta_p)\,|1\rangle$$

for $p = 3:n$. This is because if $|\boldsymbol{\phi}\rangle_2 = |0\rangle$, Eq. (B2) implies that $\cos(\theta_2) = 1$, and thus, $\sin(\theta_2) = 0$ and $(-1)^{\sin(\theta_2)} = 1$. Otherwise, that is, if $|\boldsymbol{\phi}\rangle_2 = |1\rangle$, Eq. (B2) yields $\cos(\theta_2) = 0$, and thus, $\sin(\theta_2) = \pm 1$ and $(-1)^{\sin(\theta_2)} = -1$. Because qubit 3 is not controlled by any more qubits, its state can be finalized here as

$$|\boldsymbol{\phi}\rangle_3 = |\boldsymbol{\phi}\rangle_3^2 = \cos(\theta_3)\,|0\rangle + (-1)^{\sin(\theta_1)+\sin(\theta_2)}\sin(\theta_3)\,|1\rangle\,.$$

The previous argument carries along subsequent subblocks. Therefore, qubit p of $|\boldsymbol{\phi}\rangle$ can be expressed as

$$|\boldsymbol{\phi}\rangle_p = \cos(\theta_p)\,|0\rangle + (-1)^{\sum_{i=1}^{p-1}\sin(\theta_i)}\sin(\theta_p)\,|1\rangle \tag{B3}$$

for $p = 1:n$. The formula dictates how to set parameter θ_p so that qubit p takes a particular binary value. The process transitions from the first to the last qubit. For qubit p to be set to $|0\rangle$, simply set $\theta_p = 0$. For qubit p to be set to $|1\rangle$, select $\theta_p = \pi/2$ if $\sum_{i=1}^{p-1}\sin(\theta_i)$ is even, or $\theta_p = 3\pi/2$ if $\sum_{i=1}^{p-1}\sin(\theta_i)$ is odd. This shows that state $|\boldsymbol{\phi}\rangle$ can span all canonical vectors in $\mathbb{R}^N$.

So far, we have considered a single layer of the VQC. If there are d layers, we can set the parameters of layers 1 to $(d-1)$ to zero, and then apply the established claim for the single layer only to the last layer. Hence, when layer 1 is fed with $|0\rangle_n$, its output remains $|0\rangle_n$ and is fed as input to layer 2. The last layer is eventually fed with $|0\rangle_n$, and can thus, be treated as a single layer.

The previous analysis holds for the linear and circular entanglements too. Consider again a single layer. For the linear entanglement, CZ gates are implemented between successive qubits. Accordingly, the first qubit of $|\boldsymbol{\psi}\rangle$ again is not subjected to any control gate, and so (B1) still applies. As each qubit after the first one is controlled by its previous qubit, it is easy to see that

$$|\boldsymbol{\phi}\rangle_p = \cos(\theta_p)\,|0\rangle + (-1)^{\sin(\theta_{p-1})}\sin(\theta_p)\,|1\rangle \tag{B4}$$

for $p = 2:n$. Therefore, by properly sampling $\{\theta_p\}_{p=1}^n$ from $\{0, \pi/2, 3\pi/2\}$, state $|\boldsymbol{\phi}\rangle$ can be made to take the value of any canonical vector in $\mathbb{R}^N$.

Compared to the linear, the circular entanglement differs only in the first qubit, which becomes $|\boldsymbol{\phi}\rangle_1 = \cos(\theta_1)\,|0\rangle + (-1)^{\sin(\theta_n)}\sin(\theta_1)\,|1\rangle$. Albeit the recursion is different, each qubit of $|\boldsymbol{\phi}\rangle$ can again evaluate to either $|0\rangle$ or $|1\rangle$. Specifically, for qubit p to be selected to $|0\rangle$, we set $\theta_p = 0$. For qubit p to be selected to $|1\rangle$, we set $\theta_p = \pi/2$ if its previous qubit is $|0\rangle$; and $\theta_p = 3\pi/2$, if its previous qubit is $|1\rangle$.

APPENDIX C: PROOF OF THEOREM 2

We commence with the lower bound on D_θ^*. If $\boldsymbol{\lambda}^*$ and $\boldsymbol{\lambda}_\theta^*$ are the maximizers of the dual problems in (32) and (36), respectively, we will prove that

$$D^* = D(\boldsymbol{\lambda}^*) \leqslant D_\theta(\boldsymbol{\lambda}^*) \leqslant D_\theta(\boldsymbol{\lambda}_\theta^*) = D_\theta^*. \tag{C1}$$

The equalities in (C1) hold obviously by definition. To show the first inequality in (C1), note that the dual functions $D(\boldsymbol{\lambda})$ and $D_\theta(\boldsymbol{\lambda})$ are both defined as the result of the minimization problems defined in (30) and (34). Because (34) is a restriction of (30), it implies that $D(\boldsymbol{\lambda}) \leqslant D_\theta(\boldsymbol{\lambda})$ for all $\boldsymbol{\lambda} \geqslant \mathbf{0}$. Plugging $\boldsymbol{\lambda}^*$ in the previous inequality provides the first inequality in (C1). The second inequality in (C1) holds simply because $\boldsymbol{\lambda}_\theta^*$ maximizes the dual function $D_\theta(\boldsymbol{\lambda})$ over all $\boldsymbol{\lambda} \geqslant \mathbf{0}$.

We proceed with the upper bound on D_θ^* in (40). Upon combining (34)–(36), we can express

$$D_\theta^* = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda}).$$

Let us extend $\boldsymbol{\lambda}$ to $\bar{\boldsymbol{\lambda}}$ by adding $\lambda_0 = 1$ as $\bar{\boldsymbol{\lambda}}$'s first entry. Then, function $\mathcal{L}(\mathbf{p};\boldsymbol{\lambda})$ can be compactly expressed as

$$\mathcal{L}(\mathbf{p};\boldsymbol{\lambda}) = (\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}})^\top\mathbf{p}. \tag{C2}$$

Analogously, $\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{\theta};\boldsymbol{\lambda})$ can be expressed as

$$\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{\theta};\boldsymbol{\lambda}) = (\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}})^\top\mathbf{p}(\boldsymbol{\theta}). \tag{C3}$$

Using (C2) and (C3), we can write

$$\begin{aligned}\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda}) &= (\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}})^\top\mathbf{p} + (\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}})^\top(\mathbf{p}(\boldsymbol{\theta}) - \mathbf{p})\\ &= \mathcal{L}(\mathbf{p};\boldsymbol{\lambda}) + (\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}})^\top(\mathbf{p}(\boldsymbol{\theta}) - \mathbf{p})\\ &\leqslant \mathcal{L}(\mathbf{p};\boldsymbol{\lambda}) + \|\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}}\|_\infty \cdot \|\mathbf{p}(\boldsymbol{\theta}) - \mathbf{p}\|_1,\end{aligned} \tag{C4}$$

where the inequality follows from Hölder's inequality.

The ℓ_∞ norm in (C4) can be upper bounded using the matrix inequality

$$\|\bar{\mathbf{F}}\bar{\boldsymbol{\lambda}}\|_\infty \leqslant \|\bar{\mathbf{F}}\|_\infty\|\bar{\boldsymbol{\lambda}}\|_\infty = L\|\bar{\boldsymbol{\lambda}}\|_\infty \tag{C5}$$

with L defined in (38). Plugging (C5) into (C4) provides

$$\mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda}) \leqslant \mathcal{L}(\mathbf{p};\boldsymbol{\lambda}) + L\|\bar{\boldsymbol{\lambda}}\|_\infty \cdot \|\mathbf{p}(\boldsymbol{\theta}) - \mathbf{p}\|_1.$$

The last inequality holds for all $\boldsymbol{\theta}$. If we minimize both sides over $\boldsymbol{\theta}$, the direction of the inequality remains [68]:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta};\boldsymbol{\lambda}) \leqslant \mathcal{L}(\mathbf{p};\boldsymbol{\lambda}) + L\|\bar{\boldsymbol{\lambda}}\|_\infty \cdot \min_{\boldsymbol{\theta}} \|\mathbf{p}(\boldsymbol{\theta}) - \mathbf{p}\|_1. \tag{C6}$$

The inequality in (C6) holds for all $\mathbf{p} \in \mathcal{P}$. Consider a particular $\mathbf{p} \in \mathcal{P}$. By Assumption 3, there exists $\boldsymbol{\theta}_0$ for which $\|\mathbf{p} - \mathbf{p}(\boldsymbol{\theta}_0)\|_1 \leqslant \epsilon$ for this particular $\mathbf{p}$. Parameter vector $\boldsymbol{\theta}_0$ provides the upper bound

$$\min_{\boldsymbol{\theta}} \|\mathbf{p}(\boldsymbol{\theta}) - \mathbf{p}\|_1 \leqslant \|\mathbf{p}(\boldsymbol{\theta}_0) - \mathbf{p}\|_1 \leqslant \epsilon. \tag{C7}$$

The previous argument holds for all $\mathbf{p} \in \mathcal{P}$. Plugging (C7) into (C6) yields

$$\min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}) \leqslant \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\bar{\boldsymbol{\lambda}}\|_\infty. \tag{C8}$$

Minimizing both sides of (C8) over $\mathbf{p} \in \mathcal{P}$ provides

$$\min_{\boldsymbol{\theta}} \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}) \leqslant \min_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\bar{\boldsymbol{\lambda}}\|_\infty. \tag{C9}$$

The last inequality in (C9) holds for all $\boldsymbol{\lambda} \geqslant \mathbf{0}$. If we maximize both sides over $\boldsymbol{\lambda} \geqslant \mathbf{0}$, the direction of the inequality remains and yields

$$D_\theta^* = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\boldsymbol{\theta}} \ \mathcal{L}_\theta(\boldsymbol{\theta}; \boldsymbol{\lambda}) \leqslant \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \ \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\bar{\boldsymbol{\lambda}}\|_\infty. \tag{C10}$$

Since $\bar{\boldsymbol{\lambda}}$ is extended from $\boldsymbol{\lambda}$ by $\lambda_0 = 1$, it follows that $\|\bar{\boldsymbol{\lambda}}\|_\infty = \|\boldsymbol{\lambda}\|_\infty$ if $\|\boldsymbol{\lambda}\|_\infty \geqslant 1$; and $\|\bar{\boldsymbol{\lambda}}\|_\infty = 1$, otherwise. For the latter case, it is obvious that

$$D_\theta^* \leqslant \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L = D^* + \epsilon L. \tag{C11}$$

For the former case, the order of norms predicates that otherwise, $\|\bar{\boldsymbol{\lambda}}\|_\infty = \|\boldsymbol{\lambda}\|_\infty$ and

$$\begin{aligned} D_\theta^* &\leqslant \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \ \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\boldsymbol{\lambda}\|_\infty \\ &\leqslant \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \ \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\boldsymbol{\lambda}\|_1. \end{aligned} \tag{C12}$$

The summand in the right-hand side of (C12) seems to be related to $P^* = D^* = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda})$. Unfortunately, the maximization involves the additional term $\epsilon L\|\boldsymbol{\lambda}\|_1$. Interestingly, this summand relates to the perturbed primal problem introduced in (37). More specifically, the Lagrangian function of (37) is defined as

$$\tilde{\mathcal{L}}(\mathbf{p}; \boldsymbol{\lambda}) := (\mathbf{f}_0 + \mathbf{F}\boldsymbol{\lambda})^\top \mathbf{p} + \epsilon L \|\boldsymbol{\lambda}\|_1 = \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\boldsymbol{\lambda}\|_1,$$

where we have used the property that $\boldsymbol{\lambda}^\top \mathbf{1} = \|\boldsymbol{\lambda}\|_1$ since $\boldsymbol{\lambda} \geqslant \mathbf{0}$. Under Assumption 4, the perturbed primal problem is feasible, and thus, strong duality holds. Therefore, the optimal cost of the perturbed problem in (37) equals

$$\tilde{P}^* = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \tilde{\mathcal{L}}(\mathbf{p}; \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\geqslant\mathbf{0}} \min_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(\mathbf{p}; \boldsymbol{\lambda}) + \epsilon L \|\boldsymbol{\lambda}\|_1.$$

The summand on the right-hand side of (C12) coincides with $\tilde{P}^*$. Because the original primal LP is convex, the optimal cost of the perturbed problem is known to satisfy [30, Sec. 5.6.2]:

$$P^* \geqslant \tilde{P}^* - \epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1,$$

where $\tilde{\boldsymbol{\lambda}}$ is the vector of optimal Lagrange multipliers for (37). The last inequality provides an upper bound on $\tilde{P}^*$. Plugging this bound into (C12) gives

$$D_\theta^* \leqslant P^* + \epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1 = D^* + \epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1. \tag{C13}$$

Combining (C11) and (C13) proves the upper bound on D_θ^* in (40).

To bound $\|\tilde{\boldsymbol{\lambda}}\|_1$, a standard trick is adopted; see [69, Ex. 5.3.1] and [34]. Under Assumption 4, the perturbed primal problem in (37) is strictly feasible. Multiplying both sides of (39) by $\tilde{\boldsymbol{\lambda}} \geqslant \mathbf{0}$ and summing up gives

$$\tilde{\boldsymbol{\lambda}}^\top \mathbf{F}^\top \hat{\mathbf{p}} \leqslant -\epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1 - s_0 \|\tilde{\boldsymbol{\lambda}}\|_1. \tag{C14}$$

Because problem (37) satisfies strong duality and $\hat{\mathbf{p}} \in \mathcal{P}$ is feasible for (37), it follows that

$$\begin{aligned} \tilde{P}^* &= \min_{\mathbf{p}\in\mathcal{P}} (\mathbf{f}_0 + \mathbf{F}\tilde{\boldsymbol{\lambda}})^\top \mathbf{p} + \epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1 \\ &\leqslant (\mathbf{f}_0 + \mathbf{F}\tilde{\boldsymbol{\lambda}})^\top \hat{\mathbf{p}} + \epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1 \\ &\leqslant \mathbf{f}_0^\top \hat{\mathbf{p}} + \tilde{\boldsymbol{\lambda}}^\top \mathbf{F}^\top \hat{\mathbf{p}} + \epsilon L \|\tilde{\boldsymbol{\lambda}}\|_1 \\ &\leqslant \mathbf{f}_0^\top \hat{\mathbf{p}} - s_0 \|\tilde{\boldsymbol{\lambda}}\|_1, \end{aligned}$$

where the last inequality stems from (C14).

Because the perturbed problem in (37) is a restriction of the original primal LP in (29), we get that

$$P^* \leqslant \tilde{P}^* \leqslant \mathbf{f}_0^\top \hat{\mathbf{p}} - s_0 \|\tilde{\boldsymbol{\lambda}}\|_1,$$

from which we obtain the bound in (41). This completes the proof of Theorem 2.

[1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. **26**, 1484 (1997).

[2] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symp. on Theory of Computing*, STOC '96 (ACM Press, New York, 1996), pp. 212–219.

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).

[4] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput. **46**, 1920 (2017).

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

[6] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. A **474**, 20170551 (2018).

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

[8] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[9] O. Simeone, An introduction to quantum machine learning for engineers, Found. Trends Signal Process. **16**, 1 (2022).

[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, Nat. Commun. **5**, 4213 (2014).

[11] P. Díez-Valle, D. Porras, and J. J. García-Ripoll, Quantum variational optimization: The role of entanglement and problem hardness, Phys. Rev. A **104**, 062426 (2021).

[12] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, Improving variational quantum optimization using CVaR, Quantum **4**, 256 (2020).

[13] S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg, Formulating and solving routing problems on quantum computers, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[14] C. Gambella and A. Simonetto, Multiblock ADMM heuristics for mixed-binary optimization on classical and quantum computers, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[15] I. Turtletaub, G. Li, M. Ibrahim, and P. Franzon, Application of quantum machine learning to vlsi placement, in *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD* (Association for Computing Machinery, New York, 2020), pp. 61–66.

[16] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 1 (2014).

[17] M. Ohzeki, Breaking limitation of quantum annealer in solving optimization problems under constraints, Sci. Rep. **10**, 3126 (2020).

[18] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[19] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[20] S. Hadfield, Z. Wang, E. G. Rieffel, B. O'Gorman, D. Venturelli, and R. Biswas, Quantum approximate optimization with hard and soft constraints, in *ACM International Workshop on Post Moore's Era Supercomputing* (Association for Computing Machinery, New York, 2017), pp. 15–21.

[21] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, *XY* mixers: Analytical and numerical results for the quantum alternating operator ansatz, Phys. Rev. A **101**, 012320 (2020).

[22] I. Hen and M. S. Sarandy, Driver hamiltonians for constrained optimization in quantum annealing, Phys. Rev. A **93**, 062312 (2016).

[23] I. Hen and F. M. Spedalieri, Quantum annealing for constrained optimization, Phys. Rev. Appl. **5**, 034007 (2016).

[24] P. Ronagh, B. Woods, and E. Iranmanesh, Solving constrained quadratic binary problems via quantum adiabatic evolution, Quantum Info. Comput. **16**, 1029 (2016).

[25] T. Shirai and N. Togawa, Postprocessing variationally scheduled quantum algorithm for constrained combinatorial optimization problems, IEEE Trans. Quantum Eng. **5**, 1 (2024).

[26] J. Chen, H. Westerheim, Z. Holmes, I. Luo, T. Nuradha, D. Patel, S. Rethinasamy, K. Wang, and M. M. Wilde, QSlack: A slack-variable approach for variational quantum semi-definite programming, arXiv:2312.03830.

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

[28] I. G. Ryabinkin, S. N. Genin, and A. F. Izmaylov, Constrained variational quantum eigensolver: Quantum computer search engine in the Fock space, J. Chem. Theory Comput. **15**, 249 (2019).

[29] K. Kuroiwa and Y. O. Nakagawa, Penalty methods for a variational quantum eigensolver, Phys. Rev. Res. **3**, 013197 (2021).

[30] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, 2004).

[31] S. Karimi and P. Ronagh, A subgradient approach for constrained binary optimization via quantum adiabatic evolution, Quant. Inf. Process. **16**, 185 (2017).

[32] S. Gupta and V. Kekatos, A quantum approach for stochastic constrained binary optimization, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Process* (IEEE, New York, 2023), pp. 1–5.

[33] D. Patel, P. J. Coles, and M. M. Wilde, Variational quantum algorithms for semidefinite programming, Quantum **8**, 1374 (2024).

[34] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, Learning optimal resource allocations in wireless systems, IEEE Trans. Signal Process. **67**, 2775 (2019).

[35] S. Gupta, V. Kekatos, and M. Jin, Controlling smart inverters using proxies: A chance-constrained DNN-based approach, IEEE Trans. Smart Grid **13**, 1310 (2022).

[36] H. Uzawa, Iterative methods for concave programming, in *Studies in Linear and Nonlinear Programming* (Stanford University Press, Stanford, CA, 1958), pp. 154–165.

[37] M. Kallio and A. Ruszczyński, Perturbation methods for saddle point computation, Tech. Rep. (International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994), https://pure.iiasa.ac.at/id/eprint/4171/.

[38] M. Kallio and C. H. Rosa, Large-scale convex optimization via saddle point computation, Oper. Res. **47**, 93 (1999).

[39] G. M. Korpelevich, The extragradient method for finding saddle points and other problems, Matecon **12**, 747 (1976).

[40] H. Lu and J. Yang, Nearly optimal linear convergence of stochastic primal-dual methods for linear programming, arXiv:2111.05530.

[41] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).

[42] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers* (Springer, Cham, Switzerland, 2021).

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

[44] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[45] A. W. Harrow and J. C. Napp, Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms, Phys. Rev. Lett. **126**, 140502 (2021).

[46] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, Quantum **4**, 314 (2020).

[47] K. M. Nakanishi, K. Fujii, and S. Todo, Sequential minimal optimization for quantum-classical hybrid algorithms, Phys. Rev. Res. **2**, 043158 (2020).

[48] I. Y. Zabotin, Subgradient method to find the saddle point of a convex-concave function, J. Soviet Math. **61**, 2321 (1992).

[49] T.-H. Chang, A. Nedić, and A. Scaglione, Distributed constrained optimization by consensus-based primal-dual perturbation method, IEEE Trans. Auto. Control **59**, 1524 (2014).

[50] R. M. Karp, *Reducibility among Combinatorial Problems* (Springer, Boston, 2010).

[51] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv:1811.08419.

[52] 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).

[53] D-Wave Quantum Inc. Hybrid solvers for quadratic optimization, Tech. Rep. (2022), https://www.dwavesys.com/resources/white-paper/hybrid-solvers-for-quadratic-optimization.

[54] X. Wang and I. Davidson, Flexible constrained spectral clustering, in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, 2010), pp. 563–572.

[55] X. Wang, B. Qian, and I. Davidson, On constrained spectral clustering and its applications, Data Min. Knowl. Disc. **28**, 1 (2014).

[56] P. Wang, C. Shen, A. van den Hengel, and P. H. Torr, Large-scale binary quadratic optimization using semidefinite relaxation and applications, IEEE Trans. Pattern Anal. Mach. Intell. **39**, 470 (2016).

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

[58] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, Quantum **3**, 214 (2019).

[59] C.-Y. Park, M. Kang, and J. Huh, Hardware-efficient ansatz without barren plateaus in any depth, arXiv:2403.04844.

[60] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[61] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).

[62] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross *et al.*, Quantum computing with Qiskit, arXiv:2405.08810.

[63] Gurobi Optimization, LLC, Gurobi optimizer reference manual (2023), https://www.gurobi.com.

[64] J. Lofberg, Yalmip: A toolbox for modeling and optimization in Matlab, in *2004 IEEE International Conference on Robotics and Automation* (IEEE, New York, 2004), pp. 284–289.

[65] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, An adaptive optimizer for measurement-frugal variational algorithms, Quantum **4**, 263 (2020).

[66] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, Adv. Quantum Technol. **2**, 1900070 (2019).

[67] R. T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, NJ, 1970).

[68] It is easy to show that if $g_1(\boldsymbol{\theta}) \leqslant g_2(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, then $\min_{\boldsymbol{\theta}} g_1(\boldsymbol{\theta}) \leqslant \min_{\boldsymbol{\theta}} g_2(\boldsymbol{\theta})$ and $\max_{\boldsymbol{\theta}} g_1(\boldsymbol{\theta}) \leqslant \max_{\boldsymbol{\theta}} g_2(\boldsymbol{\theta})$.

[69] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. (Athena Scientific, Belmont, MA, 1999).