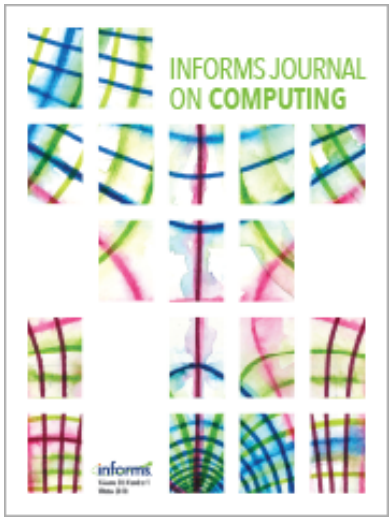


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Two-Stage Estimation and Variance Modeling for Latency-Constrained Variational Quantum Algorithms

Yunsoo Ha, Sara Shashaani, Matt Menickelly

To cite this article:
Yunsoo Ha, Sara Shashaani, Matt Menickelly (2024) Two-Stage Estimation and Variance Modeling for Latency-Constrained Variational Quantum Algorithms. INFORMS Journal on Computing

Published online in Articles in Advance 27 Nov 2024

. https://doi.org/10.1287/ijoc.2024.0575

https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Articles in Advance, pp. 1–21
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Two-Stage Estimation and Variance Modeling for Latency-Constrained Variational Quantum Algorithms

Yunsoo Ha,[a,*] **Sara Shashaani,**[b] **Matt Menickelly**[c]

[a] Computational Science Center, National Renewable Energy Laboratory, Golden, Colorado 80401; [b] Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, North Carolina 27695; [c] Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, Illinois 60439

*Corresponding author

Contact: yunsoo.ha@nrel.gov, https://orcid.org/0000-0002-9421-4768 (YH); sshasha2@ncsu.edu, https://orcid.org/0000-0001-8515-5877 (SS); mmenickelly@anl.gov, https://orcid.org/0000-0002-2023-0837 (MM)

Received: January 15, 2024
Revised: July 1, 2024; October 4, 2024
Accepted: October 11, 2024
Published Online in Articles in Advance: November 27, 2024

https://doi.org/10.1287/ijoc.2024.0575

Abstract. The quantum approximate optimization algorithm (QAOA) has enjoyed increasing attention in noisy, intermediate-scale quantum computing with its application to combinatorial optimization problems. QAOA has the potential to demonstrate a quantum advantage for NP-hard combinatorial optimization problems. As a hybrid quantum-classical algorithm, the classical component of QAOA resembles a simulation optimization problem in which the simulation outcomes are attainable only through a quantum computer. The simulation that derives from QAOA exhibits two unique features that can have a substantial impact on the optimization process: (i) the variance of the stochastic objective values typically decreases in proportion to the optimality gap, and (ii) querying samples from a quantum computer introduces an additional latency overhead. In this paper, we introduce a novel stochastic trust-region method derived from a derivative-free, adaptive sampling trust-region optimization method intended to efficiently solve the classical optimization problem in QAOA by explicitly taking into account the two mentioned characteristics. The key idea behind the proposed algorithm involves constructing two separate local models in each iteration: a model of the objective function and a model of the variance of the objective function. Exploiting the variance model allows us to restrict the number of communications with the quantum computer and also helps navigate the nonconvex objective landscapes typical in QAOA optimization problems. We numerically demonstrate the superiority of our proposed algorithm using the SimOpt library and Qiskit when we consider a metric of computational burden that explicitly accounts for communication costs.

History: Accepted by Giacomo Nannicini, Area Editor for Quantum Computing and Operations Research. Accepted for Special Issue.

Funding: This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers and the Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing program under contract number DE-AC02-06CH11357. Y. Ha and S. Shashaani also gratefully acknowledge the U.S. National Science Foundation Division of Civil, Mechanical and Manufacturing Innovation Grant CMMI-2226347 and the U.S. Office of Naval Research [Grant N000142412398].

Supplemental Material: The software that supports the findings of this study is available within the paper and its Supplemental Information (https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2024.0575) as well as from the IJOC GitHub software repository (https://github.com/INFORMSJoC/2024.0575). The complete IJOC Software and Data Repository is available at https://informsjoc.github.io/.

Keywords: quantum approximate optimization algorithm • simulation optimization • derivative-free optimization • two-stage estimation • state-dependent noise • trust-region optimization • adaptive sampling

1. Introduction

Quantum computers have the potential to outperform their classical counterparts on numerous critical calculations. Diverse fields, including data science (Biamonte et al. 2017), quantum chemistry (Lanyon et al. 2010), condensed matter (Smith et al. 2019), nuclear physics (Cloët et al. 2019), and even finance (Orus et al. 2019), stand to benefit from quantum algorithms in various ways in the future. However, in the near-term, noisy, intermediate-scale quantum (NISQ) era (Preskill 2018), realizing these theoretical advantages is challenging. This is because canonical quantum algorithms used in many of these fields necessitate gate depths, the longest sequence of operations from the input (the initial state) to the output (the final measurement), that are only expected to be achievable with fault-tolerant, error-corrected quantum computers (Preskill 1998).

Variational quantum algorithms (VQAs) aim to reduce gate depth requirements by exploiting classical computer-based optimization processes (Cerezo et al. 2021). These algorithms have demonstrated their effectiveness on NISQ hardware in tasks such as dynamical evolution (Otten et al. 2019, Yuan et al. 2019), eigenvalue estimation (O'Malley et al. 2016), machine learning (Mitarai et al. 2018, Otten et al. 2020), and various other problem domains (Cerezo et al. 2021). One of the primary challenges in VQAs lies in the optimization step, which is performed on classical computers. The optimization step involves estimating the expected value of a cost function (and, potentially, its derivative information) derived from the problem being solved, using a limited number of samples—also referred to as "shots" in this context. The estimation of an expectation objective function ideally necessitates the employment of stochastic optimization algorithms. A straightforward way to quantify the overall complexity of a classical stochastic optimizer is by counting the total number of simulation oracle calls to achieve a zeroth or first order optimality gap of ϵ. To estimate the cost function with a given a set of parameters (decision variables), multiple shots must be executed on a correspondingly parameterized quantum circuit. The estimation error is quantifiable analogous to Monte Carlo estimators. With this perspective, the optimization performed on the classical computer can be seen as a form of simulation optimization (SO). For flexibility, we make no assumptions about the accessibility of (directional) derivatives in the VQA context; such a setting necessitates derivative-free SO solvers. Derivative-free SO solvers generate solution paths for simulations (stochastic oracles) that do not provide direct derivative observations—also known as zeroth order oracles. Before discussing SO in further detail, we begin by introducing a specific example of a VQA especially relevant to operations research.

1.1. Quantum Approximate Optimization Algorithm

The quantum approximate optimization algorithm (QAOA) is a particular and well-studied instance of a VQA, designed to solve a class of combinatorial optimization problems. In QAOA, once a combinatorial optimization problem (such as a max-cut problem with a given graph) is provided, a matrix H_C—called the cost Hamiltonian—is specifically (and implicitly) constructed in such a way to ensure that its ground state (lowest eigenvalue) corresponds to the optimal solution to the original combinatorial optimization problem. QAOA relies on what is known as the variational principle, which states $\langle\psi(\mathbf{x})|H_C|\psi(\mathbf{x})\rangle \geq E_0$, where E_0 is the ground state energy, $|\psi(\mathbf{x})\rangle$ is a quantum state vector parameterized by $\mathbf{x}$, and $\langle\psi(\mathbf{x})|$ is the conjugate transpose of $|\psi(\mathbf{x})\rangle$. We, thus, aim to solve the problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^d} \langle\psi(\mathbf{x})|H_C|\psi(\mathbf{x})\rangle. \tag{1}$$

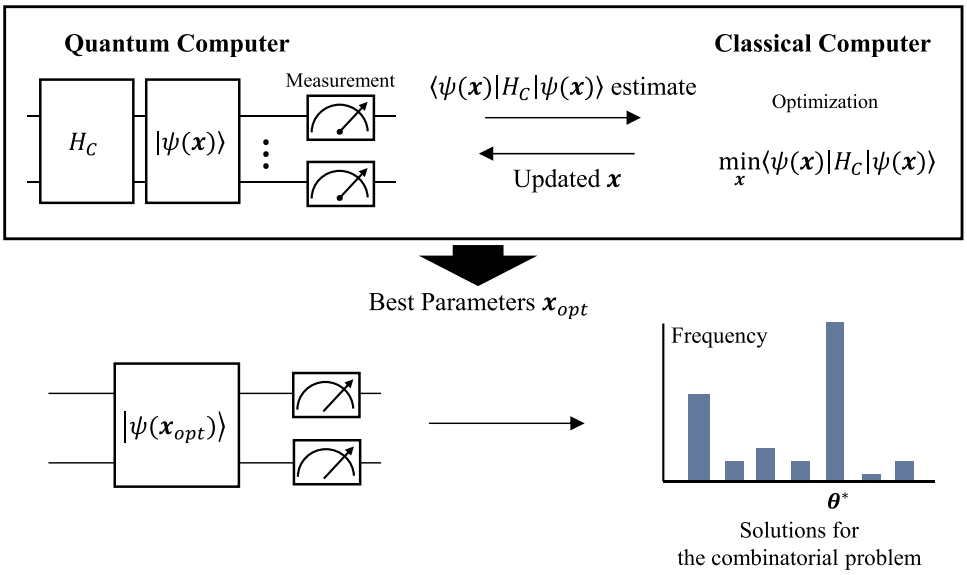
Given that the quantum state vector collapses to a single state upon measurement, we must estimate $\langle\psi(\mathbf{x})|H_C|\psi(\mathbf{x})\rangle$, which represents an expectation of a physical quantity by repeatedly measuring the quantum state and employing Monte Carlo sampling. For convenience of notation in discussing an optimization algorithm, we let $F(\mathbf{x},\xi)$ denote a stochastic objective function value (encoding all sources of stochasticity in the random variable ξ), and rewrite (1) as

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \mathbb{E}_\xi[F(\mathbf{x},\xi)].$$

Figure 1 illustrates the steps of a QAOA. As in our discussion of general VQAs, the quantum computer in Figure 1 can be viewed as a stochastic oracle that is iteratively queried by a classical computer. The parameters $\mathbf{x}\in\mathbb{R}^d$ that describe the state $|\psi(\mathbf{x})\rangle$ are updated by the classical computer based on the stochastic measurements (realizations) of $\langle\psi(\mathbf{x})|H_C|\psi(\mathbf{x})\rangle$ using the quantum computer. After expending a maximum number of shots (budget) or reaching some other stopping criteria determined by the algorithm implemented in the classical computer, the best observed parameters $\mathbf{x}_{opt}$ are measured one more time with a number of shots to yield an empirical discrete distribution on the (finite, but combinatorial, number of) possible bit strings feasible for the combinatorial optimization problem. Near the optimal eigenstate solution to (1), state vectors have a high probability of collapsing to the optimal solution of the original combinatorial problem upon measurement. Therefore, the solution(s) of highest frequency is/are interpreted as candidates for the global optimizer of the combinatorial optimization problem.

1.1.1. Latency. The overall computational expense of executing a QAOA can be assessed in a similar way to how SO typically quantifies expense, which is by counting the number of shots needed to attain a sufficiently accurate solution. However, the current state of the art for quantum computers involves additional latencies that are less seriously considered when designing algorithms for state-of-the-art classical computers. Latencies can differ across various architectures. For instance, a superconducting quantum processor has measurement times in the range of a few microseconds; see Gambetta et al. (2007). In contrast, a trapped ion system can require

Figure 1. (Color online) Illustration of the QAOA



Notes. Once a sufficiently good solution, $\mathbf{x}_{opt}$, is identified, a QAOA proceeds to obtain a probability distribution by measuring the quantum state $|\psi(\mathbf{x}_{opt})\rangle$. In this distribution, the solution with the highest frequency corresponds to the optimal solution for the original combinatorial problem, $\boldsymbol{\theta}^*$. In the context of a combinatorial optimization problem with cost Hamiltonian H_C, a QAOA iteratively updates a parameter vector $\boldsymbol{x}$ to minimize the objective function value in (1).

hundreds of microseconds to perform a measurement (Bruzewicz et al. 2019, Clark et al. 2021). These measurement times are in addition to the duration of gate operations and system resets, all contributing to the time needed to acquire a single sample of shots. Moreover, many modern quantum computers operate in a cloud environment, leading to potential extra overhead from network latency. For perspective on the scale of latency costs, Sung et al. (2020) suggest that, whereas acquiring a single shot could take about 1.0×10^{-5} seconds in a supercomputing hardware environment, the overhead can be on the order of several seconds.

Hence, designing optimization algorithms that consider these latencies is crucial for efficiently using quantum resources in the near term. Making explicit latency considerations in the design of (theoretical) algorithms for VQAs was previously studied (Menickelly et al. 2023). The work that we present in this paper is meant to provide a slightly more heuristic, but practical, means to controlling latency within an adaptive sampling framework; this is seen in our two-stage estimation approach.

1.1.2. State-Dependent Noise. Another distinction of the general VQA setting is that in problems of the form (1), it is well-known that an eigenstate should exhibit zero variance; see numerous references within Zhang et al. (2022). In this paper, we specifically define this characteristic as state-dependent noise, represented by the following equation (for some constants $C_0, C_1 \geq 0$):

$$\sigma^2(\mathbf{x}) \leq C_0 + C_1(f(\mathbf{x}) - f_{\min}), \tag{2}$$

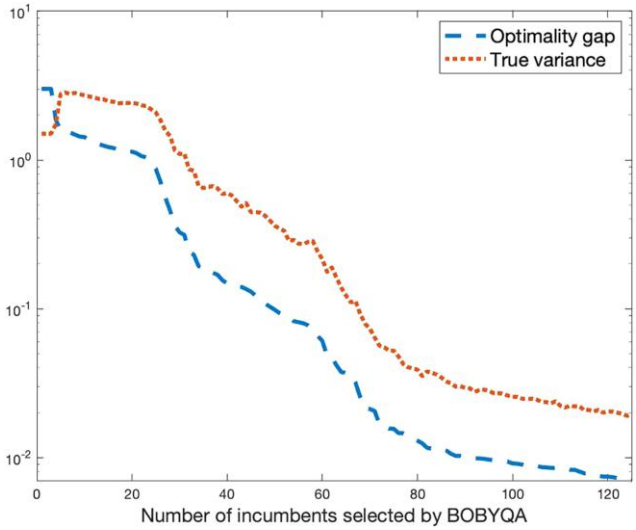
where $\sigma^2(\mathbf{x}) := \mathbb{E}_\xi[(F(\mathbf{x},\xi) - f(\mathbf{x}))^2]$ is the true variance function at $\mathbf{x}$ and $f_{\min}$ represents the optimal value. Whereas we do not practically require linearity in the optimality gap $f(\mathbf{x}) - f_{\min}$, as written in Equation (2), we coarsely imagine $\sigma^2(\mathbf{x})$ being bounded by some function of the optimality gap. In Figure 2, we demonstrate this phenomenon on a small-scale example of using a QAOA circuit for solving a max-cut problem on a toy graph. This zero-variance principle has been exploited recently in a quantum computer to self-verify whether a ground state for a given Hamiltonian was accurately prepared (Kokail et al. 2019). Recently, and also inspired by this phenomenon, Zhang et al. (2022) consider regularizing VQA cost functions with a measure of estimated variance.

We assume that f is bounded below and twice continuously differentiable in $\mathbb{R}^d$ and that the variance function is L_v-Lipschitz continuous, that is,

$$|\sigma^2(\mathbf{x}_1) - \sigma^2(\mathbf{x}_2)| \leq L_v\|\mathbf{x}_1 - \mathbf{x}_2\|, \ \ \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d.$$

The differentiability of f suggests that the distribution of solutions for the original combinatorial problem varies in proportion to perturbations in $\mathbf{x}$. Therefore, we can expect the variance of the distribution to be continuous in $\mathbf{x}$. This assumption is empirically justified in Section 6. We further remark that, in the VQA setting, there are no

Figure 2. (Color online) Optimality Gap and Population Variance of the Improving Sequence of Incumbents



Notes. We used BOBYQA (Powell 2009) with the deterministic expectation statevector value for this toy problem and record the improving sequence of incumbent solutions returned. We illustrate, on the same log scale, the optimality gap of the incumbents found as well as the population variance associated with the statevector value. As expected, population variance decays alongside the optimality gap but not necessarily monotonically. In this Plot, we simulate in Qiskit (Javadi-Abhari et al. 2021) a depth-10 QAOA circuit for solving a max-cut problem on a toy graph of six nodes for which the optimal solution is six.

common random numbers that would allow for the faithful reproduction of a sample path, a feature that precludes the use of some techniques in SO (Ha et al. 2024b).

1.2. Adaptive Sampling

In derivative-free SO, stochastic trust-region optimization (TRO) has become increasingly popular as a methodology for solving nonlinear and nonconvex optimization problems (Chang et al. 2013, Chen et al. 2018, Shashaani et al. 2018, Sun and Nocedal 2023, Cao et al. 2024). Stochastic TRO methods generate a random sequence of incumbents, denoted $\{\mathbf{X}_k\}$, during a single run. Incumbent selection depends on approximations of the objective function by means of local objective models (OMs) and respective approximate minimizers of these models within dynamically sized neighborhoods. In the derivative-free setting, in which derivative information is assumed unavailable, these local OMs are typically computed via interpolation or regression techniques, employing objective value estimates at design points near the current incumbent. To ensure the accuracy of local OMs, it is imperative to have access to sufficiently accurate estimates of the objective values. In the stochastic optimization setting, sufficient accuracy can be achieved by averaging a sufficiently large number of samples. Thus, if an excessive number of simulation oracle calls (shots on a quantum computer) are required to attain this necessary precision, it becomes challenging to find a satisfactory approximate solution to the optimization problem within a reasonable time frame. Hence, for the purpose of judiciously determining a suitable sample size, Shashaani et al. (2018) introduce an adaptive sampling approach within the TRO framework: derivative-free, adaptive sampling trust-region optimization (ASTRO-DF). An adaptive sampling strategy dynamically determines the sample size by balancing the estimation error at each point with a measure of first order optimality error. This dynamic strategy produces a random sample size that is a stopping time with respect to the generated observations at the design point of interest. ASTRO-DF is shown to almost surely achieve optimal worst case efficiency as measured by the expected total sample complexity (Ha et al. 2024b).

However, and as we set out to address in this paper, an adaptive sampling strategy requires repeated message passing between the optimization engine and the computer simulation, which may be prohibitive in latency-constrained settings. One of the key distinctions between the VQA setting and classical SO lies in the fact that in VQA, the simulation is entirely handled by the quantum computer. In the VQA setting, a quantum circuit is calibrated according to parameters $\mathbf{x}$, which is then executed a number of times. Consequently, this process involves a nontrivial amount of communication between the quantum computer and the classical computer, adding an equally nontrivial computational burden to the overall optimization procedure. Thus, whereas an adaptive sampling strategy can reduce the total number of replications by incrementally adding shots until the estimated

variance satisfies a particular condition, it may not necessarily alleviate the overall computational burden because of the communication costs.

We remark that, because ASTRO-DF requires an estimate of $\sigma^2(\mathbf{x})$ to compute the number of samples (shots) requested at a design point $\mathbf{x}$, and because this number of shots scales linearly with the estimate of $\sigma^2(\mathbf{x})$, we anticipate that in state-dependent noise settings, decaying variance near optimality plays a mitigating role in the number of required shots as contrasted with the decreasing trust-region radius.

1.3. Our Contributions

Motivated by the particularities of VQA problems, we propose (i) a replacement for adaptive sampling in ASTRO-DF with a two-stage sampling strategy and (ii) a refinement for the local OM construction employed in ASTRO-DF. Both of these contributions hinge on a secondary local model that interpolates or regresses variance estimates of the stochastic objective function in order to locally approximate the variance of previously unevaluated design points or incumbents. We term this secondary model the variance model (VM).

The proposed two-stage estimation approach ensures at most two communications between the optimization engine and the quantum computer for evaluation of each choice of parameters. The VM helps to achieve this by predicting the variance at previously unevaluated parameters. The two-stage estimation procedure, hence, begins with that initial prediction of variance from the local VM and then requests another set of shots through an additional communication with the quantum device only if further refinement of the objective estimate was deemed necessary. This alleviates the insistence, as in the usual adaptive sampling setting, on incremental sampling.

Predictions from the VM additionally aid in choosing interpolation points in new incumbent neighborhoods. In particular, when selecting design points for OM-building, we prioritize points that are predicted to exhibit lower variance. A particular advantage of this prioritization is that it provides a heuristic intended to escape local minima in objective functions exhibiting state-dependent noise.

We delve into the details of our proposed uses of a VM throughout the optimization process in Sections 3 and 4, respectively.

2. Simulation Optimization with Trust Regions

Stochastic TRO is effective at solving zeroth order nonconvex stochastic optimization problems. Its salient feature is a natural ability to self-tune step sizes and facility for incorporating approximate curvature information. We provide a fairly generic framework that describes the vast majority of derivative-free stochastic TRO methods. A set of design points $\mathcal{X}_k$ are evaluated in a neighborhood of the incumbent $\mathbf{X}_k$. A local OM denoted by $M_k : \mathbb{R}^d \to \mathbb{R}$ is fitted to those objective evaluations at $\mathcal{X}_k$. An approximate minimizer of M_k over a trust region of size Δ_k, that is, $\mathcal{B}(\mathbf{X}_k; \Delta_k)$, is computed and denoted $\tilde{\mathbf{X}}_{k+1}$. If $\tilde{\mathbf{X}}_{k+1}$ witnesses sufficient decrease over $\mathbf{X}_k$ in objective estimates, then we set the next incumbent as $\mathbf{X}_{k+1} = \tilde{\mathbf{X}}_{k+1}$. ASTRO-DF is a variant of this class of algorithms that embeds adaptive sampling to determine a judicious lower bound on the number of oracle calls (shots) required at each design point to guarantee efficient optimization progress. The key element of this approach involves allocating computational resources based on a measure of the optimality gap, such as $\|\nabla f(\mathbf{X}_k)\|$, which ASTRO-DF consistently monitors by means of a positive power of the trust-region radius Δ_k. As a result, it is typical that more computational effort is expended on points in closer proximity to first order critical regions. Before we expound on recent developments in ASTRO-DF, we begin by introducing the notation and definitions that are employed throughout this paper.

2.1. Notation and Definition

We use capital letters for random variables or random functions, bold fonts for vectors, script fonts for sets and random fields, and a sans serif font for matrices. For example, there are random sample sizes N, local models M, incumbents $\mathbf{X}$, minimizers of local models $\tilde{\mathbf{X}}$, and sets of design points $\mathcal{X}$, each augmented with various superscripts and subscripts.

In addition to $f(\mathbf{x})$ and $F(\mathbf{x}, \xi)$, we define a sample average estimate using $N(\mathbf{x})$ samples

$$\overline{F}(\mathbf{x}, N(\mathbf{x})) = \frac{1}{N(\mathbf{x})} \sum_{i=1}^{N(\mathbf{x})} F(\mathbf{x}, \xi_i),$$

where ξ_i, $i = 1, \ldots, N(\mathbf{x})$ denote independent and identically distributed realizations of ξ.

Definition 1 (Stochastic Interpolation Models). Let $\Phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \ldots, \phi_q(\mathbf{x}))$ form a linearly independent set of polynomials on $\mathbb{R}^d$. With $q = p+1$, $\mathbf{X}_k^0 := \mathbf{X}_k$ and the design set $\mathcal{X}_k := \{\mathbf{X}^j\}_{j=0}^p \subset \mathcal{B}(\mathbf{X}_k; \Delta_k)$, consider the linear system

$$\mathsf{M}(\Phi, \mathcal{X}_k)\boldsymbol{\beta}_k = \overline{F}(\mathcal{X}_k, N(\mathcal{X}_k)), \tag{3}$$

where

$$\mathsf{M}(\Phi, \mathcal{X}_k) = \begin{bmatrix} \phi_1(\mathbf{X}_k^0) & \phi_2(\mathbf{X}_k^0) & \cdots & \phi_q(\mathbf{X}_k^0) \\ \phi_1(\mathbf{X}_k^1) & \phi_2(\mathbf{X}_k^1) & \cdots & \phi_q(\mathbf{X}_k^1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(\mathbf{X}_k^p) & \phi_2(\mathbf{X}_k^p) & \cdots & \phi_q(\mathbf{X}_k^p) \end{bmatrix}, \overline{F}(\mathcal{X}_k, N(\mathcal{X}_k)) = \begin{bmatrix} \overline{F}(\mathbf{X}_k^0, N(\mathbf{X}_k^0)) \\ \overline{F}(\mathbf{X}_k^1, N(\mathbf{X}_k^1)) \\ \vdots \\ \overline{F}(\mathbf{X}_k^p, N(\mathbf{X}_k^p)) \end{bmatrix}.$$

We say the set $\mathcal{X}_k$ is poised with respect to $\Phi(\mathbf{x})$ if the matrix $\mathsf{M}(\Phi, \mathcal{X}_k)$ is nonsingular, guaranteeing the existence of a solution $\boldsymbol{\beta}_k = (\beta_{k,j}, j = 0, 1, 2, \ldots, p)$ to (3). In that case, the function $M_k(\mathbf{x}) = \sum_{j=0}^p \beta_{k,j}\phi_j(\mathbf{x})$ is a stochastic interpolation model of estimated values of f on $\mathcal{B}(\mathbf{X}_k; \Delta_k)$.

We specialize Definition 1 to the case in which a diagonal model Hessian is employed.

Definition 2 (Stochastic Quadratic Interpolation Models with Diagonal Hessian). Let $\Phi(\mathbf{x})$ be the polynomial basis $\{1, x_1, \ldots, x_d, x_1^2, \ldots, x_d^2\}$ so that $p = 2d$. Let $\mathcal{X}_k$ be poised with respect to $\Phi(\mathbf{x})$ and let $M_k(\mathbf{x})$ be a stochastic interpolation model. Then, denoting $\boldsymbol{G}_k := [\beta_{k,1} \;\; \beta_{k,2} \;\; \cdots \;\; \beta_{k,d}]^\top$ and letting H_k be a $d \times d$ matrix with $[\mathsf{H}_k]_{j,j} = \beta_{k,d+j}$ and zeros off the diagonal, we refer to

$$M_k(\mathbf{x}) = \beta_{k,0} + (\mathbf{x} - \mathbf{X}_k)^\top \boldsymbol{G}_k + \frac{1}{2}(\mathbf{x} - \mathbf{X}_k)^\top \mathsf{H}_k(\mathbf{x} - \mathbf{X}_k) \tag{4}$$

as a stochastic quadratic interpolation model of f with a diagonal Hessian.

A particular utility of Definition 2 is that any coordinate stencil, such as $\mathcal{X}_k^c = \{\mathbf{X}_k, \mathbf{X}_k + \boldsymbol{e}_1\Delta_k, \ldots, \mathbf{X}_k + \boldsymbol{e}_d\Delta_k, \mathbf{X}_k - \boldsymbol{e}_1\Delta_k, \ldots, \mathbf{X}_k - \boldsymbol{e}_d\Delta_k\}$, where $\boldsymbol{e}_i$ denotes the ith elementary basis vector of $\mathbb{R}^d$, is clearly poised with respect to $\Phi(\mathbf{x})$; Definition 2 is, therefore, immediately nonvacuous.

Definition 3 (Filtration). A filtration $\{\mathcal{F}_k\}_{k\geq 1}$ over a probability space $(\Omega, \mathbb{P}, \mathcal{F})$ is an increasing sequence of random fields within $\mathcal{F}$, where $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ for every k, and all are subsets of $\mathcal{F}$. The filtration $\mathcal{F}_k$ roughly represents all available information when iteration k begins.

2.2. History-Informed ASTRO-DF

Recent augmentations to ASTRO-DF have aimed at boosting computational efficiency. We refer to ASTRO-DF with these collective augmentations as history-informed ASTRO-DF. History-informed ASTRO-DF includes a direct search (see, e.g., Ha and Shashaani 2023a, b) component in each iteration to increase the likelihood of finding a new incumbent in each iteration without increasing the allotted budget. In cases in which $\tilde{\mathbf{X}}_{k+1}$, the approximate local OM minimizer, does not lead to a sufficient reduction in the estimated objective value, ASTRO-DF would declare the iteration as unsuccessful, immediately contracting the trust-region radius. This declaration of an unsuccessful iteration and trust-region contraction would occur even if some design points in $\mathcal{X}_k$ yielded improvements over the incumbent. Electing to replace the next incumbent with the best design point from $\mathcal{X}_k$ is tantamount to a direct search iteration. Practically, having fewer unsuccessful iterations because of the direct search feature amounts to a slower rate of decay in the trust-region radius Δ_k; in turn, this slower rate keeps the sample size (which is proportional to Δ_k^{-4}; see (6)) from growing too quickly.

The efficiency of a TRO algorithm is closely tied to the geometry of the design set $\mathcal{X}_k$ in each iteration because the choice of $\mathcal{X}_k$ directly impacts the quality of the local OM. As an abstract example, if the design points in $\mathcal{X}_k$ lie entirely in a single half-space intersected with the trust region, M_k is prone to having poor predictive accuracy on the complementary half-space. In addition to geometry, the quantity of design points contained in $\mathcal{X}_k$ in each iteration plays a crucial role. Employing an excessive number of design points per iteration can lead to prolonged computation times. Conversely, if we use too few design points relative to the choice of basis $\Phi(\mathbf{x})$ in Definition 1 so that $p+1 < q$, then System (3) is underdetermined, and local OMs need only satisfy error bounds for underdetermined systems that are generally worse than the error bounds derivable for determined systems of Equation (3); see, for example, Conn et al. 2009, section 5). To address all of these issues, and based on practical experience, history-informed ASTRO-DF makes the explicit choice to select $2d+1$ design points per iteration and always employs a coordinate basis; see Definition 2. This choice of basis and $\mathcal{X}_k^c$ allows us to capture some curvature

information within a reasonable computational time frame and simultaneously realizes a well-distributed placement of design points within the trust region to faithfully approximate the objective function.

Moreover, history-informed ASTRO-DF consistently employs a (rotated) coordinate stencil like $\mathcal{X}_k^c$ as the set of design points via a reuse strategy to conserve computational resources in each iteration. If there are previously evaluated design points located within the trust region in an iteration, the design point (denote it $\boldsymbol{Y}$) that is farthest from the incumbent $\mathbf{X}_k$ is added to the design set $\mathcal{X}_k^c$, and the simulation results at that point are reused. The remaining members of the design set $\mathcal{X}_k^c$ are selected deterministically by computing a set of mutually orthonormal vectors, all orthogonal to $\boldsymbol{Y} - \mathbf{X}_k$, and using these vectors to generate a rotated coordinate basis. The benefit of employing a rotated coordinate basis is that we obtain a more precise gradient estimate at $\mathbf{X}_k$ than if we had used alternative bases (Ha and Shashaani 2023b); in fact, among all design sets satisfying $|\mathcal{X}_k| = 2d + 1$, a rotated coordinate basis is optimal in a precise sense (Ragonneau and Zhang 2023). As a practical matter, it is often the case that $\boldsymbol{Y} = \mathbf{X}_{k-1}$ after a successful iteration; therefore, this reuse strategy, in tandem with the direct search strategy, extrapolate the search direction from the previous iteration. See Figure A.1 for an illustration of the rotated coordinate basis selection performed by history-informed ASTRO-DF.

Past empirical evidence demonstrates that history-informed ASTRO-DF is superior in performance to the older implementation of ASTRO-DF that makes a more general choice of basis (Ha and Shashaani 2023b) on a range of SO problems as measured by progress made within a simulation budget. Table 1 summarizes several key differences between the original ASTRO-DF and history-informed ASTRO-DF.

3. Point Selection

In this paper, we propose exploiting a previously discussed characteristic of VQAs, namely, the diminishing variance with respect to the optimality gap. To harness the potential acceleration of ASTRO-DF through the utilization of this distinctive property, we propose an extension of the point-selection strategy of history-informed ASTRO-DF. Central to the new point selection strategy is the construction of the VM: a local model that interpolates or regresses variance estimates. Within ASTRO-DF, upon computation of sample average objective estimates at each design point $\mathbf{X}_k^j \in \mathcal{X}_k$, we also estimate the variance, that is,

$$\hat{\sigma}^2(\mathbf{X}_k^j, N(\mathbf{X}_k^j)) = \frac{1}{N(\mathbf{X}_k^j)} \sum_{i=1}^{N(\mathbf{X}_k^j)} (F(\mathbf{X}_k^j, \xi_i) - \overline{F}(\mathbf{X}_k^j, N(\mathbf{X}_k^j)))^2,$$

with no overhead incurred by the simulation oracle (quantum computer). Then, without additional oracle effort, we can construct a local VM by solving (3), in which we replace the right-hand side with estimated variance as opposed to estimated objective value. We denote this model by $M_k^v : \mathbb{R}^d \to \mathbb{R}$ and its (approximate) minimizer within the trust region by $\tilde{\mathbf{X}}_k^v$. Coupled with the utilization of a direct search at each iteration in the history-informed ASTRO-DF, if the design set $\mathcal{X}_k^c$ includes $\tilde{\mathbf{X}}_k^v$, then $\tilde{\mathbf{X}}_k^v$ may be considered as a potential next incumbent $\mathbf{X}_{k+1}$. Hence, $\tilde{\mathbf{X}}_k^v$ can serve as the next incumbent with both predicted low variance and objective value despite not necessarily following the approximate gradient descent direction from $\mathbf{X}_k$. Owing to this property and the correlation between variance and optimality gap frequently observed in VQAs, this proposed scheme provides a heuristic intended to drive the incumbents $\mathbf{X}_k$ to global optimality, particularly under specific variance structures such as convexity and Lipschitz continuity. We stress that this proposed scheme is indeed a heuristic because its utility is also a function of the trust-region radius, but it is a seemingly useful heuristic. Figure 3 demonstrates the effect of using two local models M_k and M_k^v in an example. We summarize one iteration of what we refer to as the two-model approach in Figure 3(b):

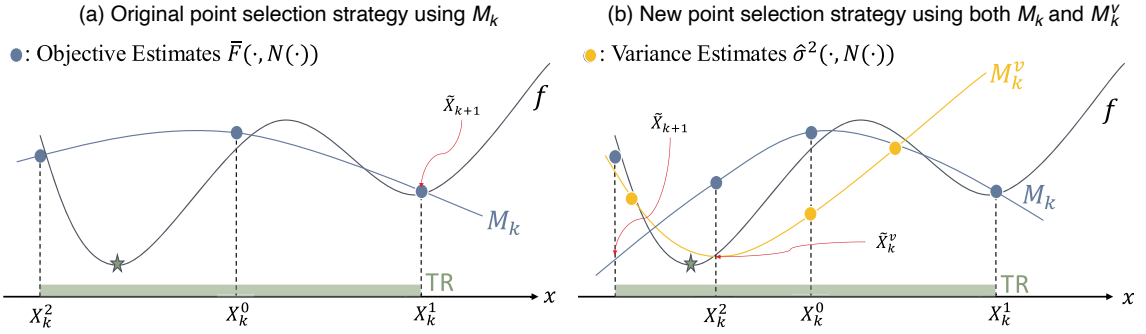
1. Construct M_k^v using variance estimates of previously evaluated design points and find its approximate minimizer within the trust region, $\tilde{\mathbf{X}}_k^v$ (steps 2–9 in Algorithm 5).

2. Determine the design set $\mathcal{X}_k$, containing $\tilde{\mathbf{X}}_k^v$ (steps 1 and 10 in Algorithm 5) and estimate the function at each point in $\mathcal{X}_k$.

Table 1. Differences Between ASTRO-DF and HISTORY-INFOrmed ASTRO-DF

Algorithm	ASTRO-DF	History-informed ASTRO-DF
Selection of $\mathcal{X}_k$	Random	Rotated coordinate basis
$\lvert\mathcal{X}_k\rvert$	$(d+1)(d+2)/2$	$2d+1$
Source of next incumbent	Model	Model and $\mathcal{X}_k$
$\lvert\mathcal{X}_k \cap \mathcal{X}_{k-1}\rvert$	≥ 0	2

Figure 3. (Color online) A Cartoon Illustrating the Effect of Using the Two Local Models



Notes. (a) The performance of the history-informed ASTRO-DF without the inclusion of M_k^v. In this case, $\tilde{\mathbf{X}}_{k+1}$ is further drawn to the basin of a local minimum. (b) The performance using two models, M_k and M_k^v. In this case, the point that minimizes M_k^v is used as a design point for interpolation, changing the shape of M_k. The minimizer of M_k in this case exhibits a global optimality-seeking behavior.

3. Construct M_k using objective estimates of design set $\mathcal{X}_k$ and then find $\tilde{\mathbf{X}}_{k+1}$ that approximately minimizes M_k within the trust region (steps 3 and 4 in Algorithm 4).

4. Obtain an objective estimate at $\tilde{\mathbf{X}}_{k+1}$ and then utilize the direct search on all points in $\mathcal{X}_k \cup \{\tilde{\mathbf{X}}_{k+1}\}$ to determine the next incumbent $\mathbf{X}_{k+1}$ (steps 5 and 6 in Algorithm 4).

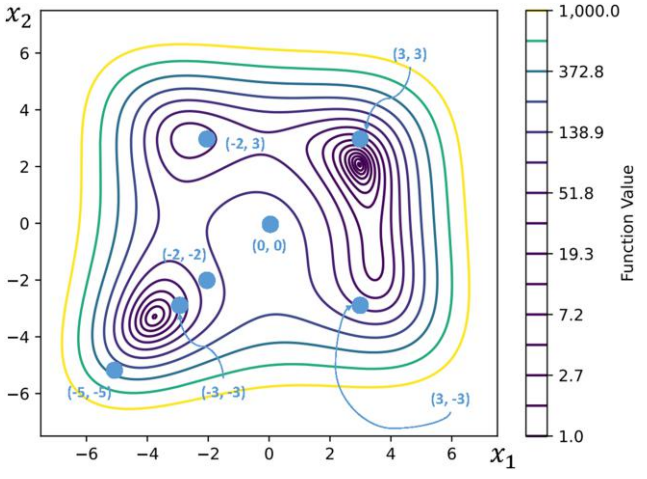
We now demonstrate the efficacy of this two-model approach by performing some preliminary experiments designed to highlight the utility of seeking variance-minimizing design points. The test problem involves a stochastic variant of the Himmelblau function. We intentionally design the stochastic noise to be state-dependent; in particular, we make the variance vanish at the global minimum of the Himmelblau function. The test problem is

$$F(\mathbf{x}, \xi) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + (x_1 - x_2)^2 + |x_1 - 3| + \xi, \tag{5}$$

where $\xi \sim \mathcal{N}(0, |(x_1 - 3)(x_2 - 2)|)$. The global minimum for the noiseless Himmelblau function is located at (3, 2); by construction, the variance at (3, 2) is zero, whereas the other local optima exhibit positive variance. Figure 4 depicts the expected objective function contours.

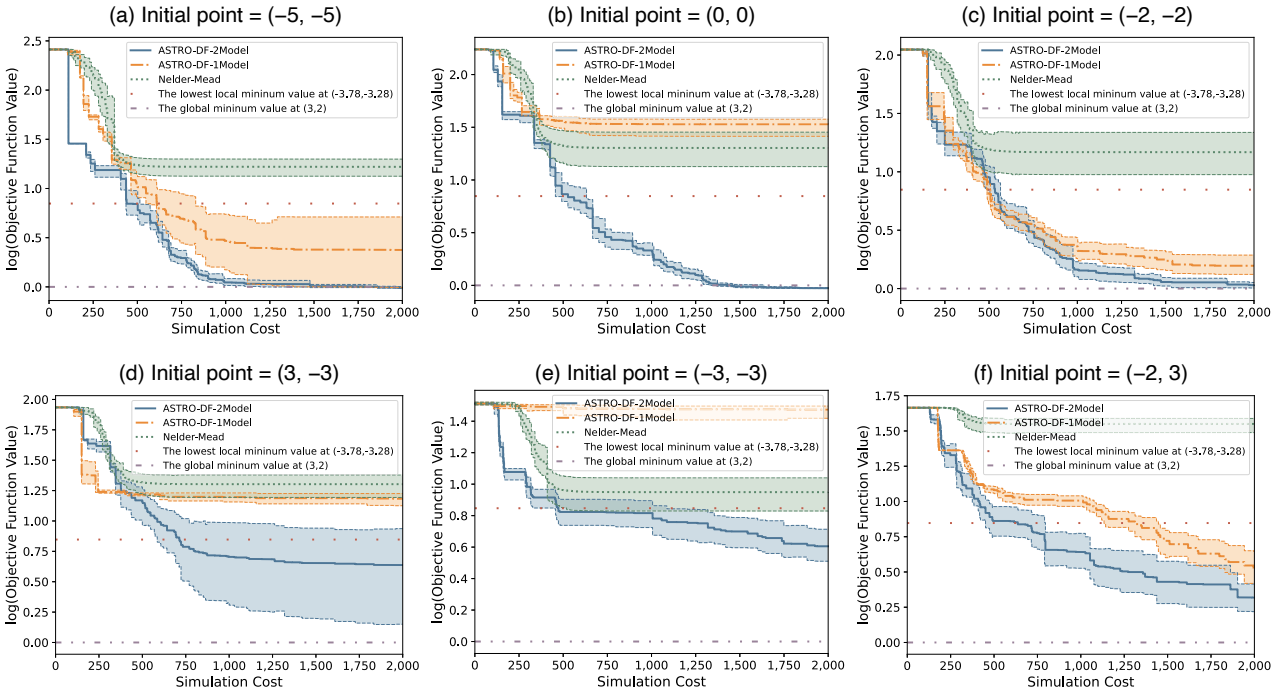
We begin our investigation by examining whether ASTRO-DF with a two-model approach, referred to as ASTRO-DF-2Model, can improve the likelihood of discovering the global optimum when compared directly with what we refer to as ASTRO-DF-1Model, which is history-informed ASTRO-DF. In Figure 5, we initialize ASTRO-DF-1Model and ASTRO-DF-2Model (and the Nelder–Mead algorithm for a baseline comparison) from various initial solutions and plot the best (estimated) objective values attained as a function of oracle calls, labeled as the simulation cost. To capture run-to-run variability of performance, each algorithm was executed 20

Figure 4. (Color online) Contour Plot of the Expectation of (5)



Notes. Each labeled point corresponds to an initial solution shown in Figures 5 and 6. The global minimum is attained at (3, 2).

Figure 5. (Color online) Performance of Solvers on (5) with Various Initial Points

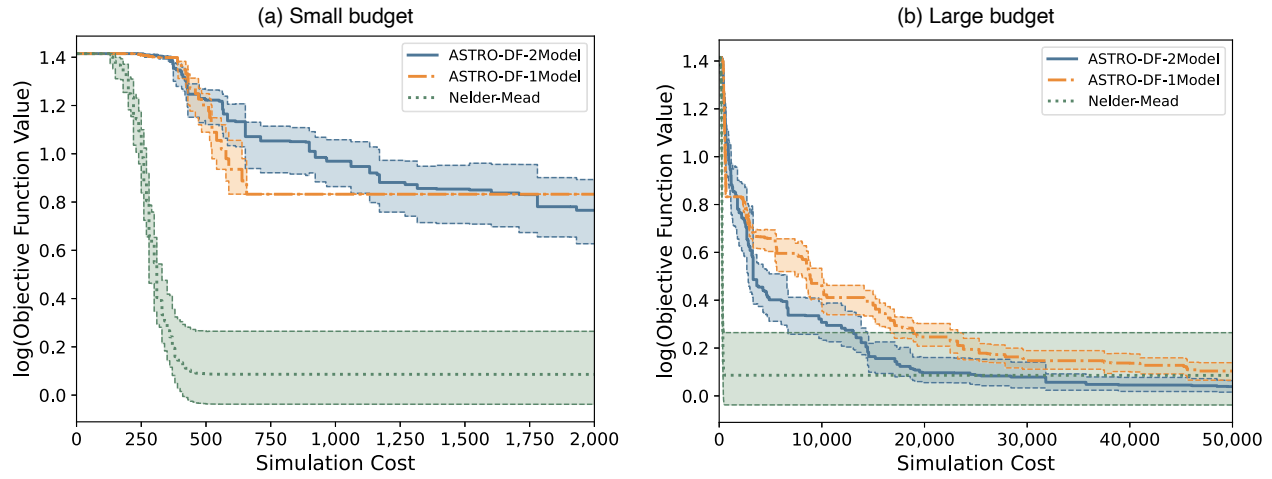


Notes. Translucent bands represent a 95% confidence interval over 20 macroreplications and solid lines represent mean performance. The y-axis is on a logarithmic scale.

times (macroreplications). This entailed generating 20 distinct sets of $\{\mathbf{X}_k\}$ until our predefined computational budget was exhausted.

From each initial point, ASTRO-DF-2Model exhibits a higher probability of identifying the basin of (3, 2) than ASTRO-DF-1Model. Notably, when the initial point is relatively far from any local minimum (Figure 5, (a) and (b)), it becomes evident that ASTRO-DF-2Model excels at discovering the global minimum. Furthermore, when the optimization process is initialized near a local optimum (Figure 5, (d)–(f)), whereas some runs of ASTRO-DF-2Model may be slower than others, the majority of them still manage to reach the global minimum eventually. Finally, when the initial point is placed relatively near the global minimum (Figure 6, (a) and (b)), the convergence rate can be relatively slow, taking a while to make the trust region radius sufficiently small to identify a

Figure 6. (Color online) Performance on (5) with 95% Confidence Interval



Notes. The y-axis is again on a logarithmic scale. The initial point, (3, 3), is relatively close to the global minimum.

new incumbent; nonetheless, with a reasonably large budget, both variants of ASTRO-DF eventually find a decrease that is more stable across macroreplications compared with Nelder–Mead.

4. Sample Size Selection

In the original implementation of ASTRO-DF (Shashaani et al. 2018), the sample size at any design point $\mathbf{x} \in \mathbb{R}^d$ is determined by the formula

$$N_k(\mathbf{x}) = \min\left\{ n \geq \lambda_k : \frac{\hat{\sigma}(\mathbf{x}, n)}{\sqrt{n}} \leq \frac{\kappa \Delta_k^2}{\sqrt{\lambda_k}} \right\}, \tag{6}$$

where $\{\lambda_k\}$ represents a deterministically increasing sequence with logarithmic growth that represents the minimum sample size at iteration k and $\kappa > 0$ is a user-defined constant. The right-hand side of the inequality in (6) is a mildly deflated proxy for the first order optimality gap; without additional assumptions beyond smoothness, in a derivative-free setting, the true gradient norm $\|\boldsymbol{G}_k\|$ is tracked by the trust-region radius if the error in objective estimates is bounded by a factor of Δ_k^2—a Taylor-like error bound. Enforcing a lockstep between estimation error and optimization error (the left- and right-hand sides in Equation (6)) helps identify a sample size that is sufficiently large for the purpose of bounding errors but not so large as to unnecessarily hamstring algorithm efficiency. The ideal sample size (provided it is larger than λ_k) should satisfy the lower bound

$$N_k(\mathbf{x}) \geq \lambda_k \frac{\sigma^2(\mathbf{x})}{\kappa^2 \Delta_k^4}. \tag{7}$$

In practice, because $\sigma^2(\mathbf{x})$ is typically unknown, its most recent estimate is used to assess (7). If the sample size is too small to satisfy (7), then the sample size is increased by one (or a small batch size) until (6) is satisfied, rendering $N_k(\mathbf{x})$ a stopping time learned on the fly (using the new observations) and adapting tightly to the stipulated precision in estimated quantities. In particular, $N_k(\mathbf{x})$ is the first sample size satisfying

$$N_k(\mathbf{x}) \geq \lambda_k \frac{\hat{\sigma}^2(\mathbf{x}, N_k(\mathbf{x}))}{\kappa^2 \Delta_k^4}. \tag{8}$$

As discussed, when faced with the issue of latency in accessing a quantum device in the VQA setting, this streaming approach to adaptive sampling becomes costly because each pass through the streaming loop incurs additional communication costs. To address this issue, we propose two-stage estimation strategies to restrict the number of communications with the quantum computer to at most two. We consider three distinct two-stage estimation strategies, each differing in how they select their first stage sample size.

4.1. Two-Stage Estimation Using First Stage Sample Size λ_k

In this case, irrespective of the actual variance at $\mathbf{x}$, the first stage sample size is $N_{k,1}(\mathbf{x}) = \lceil \lambda_k \rceil$ (which slowly grows with k); the corresponding estimated variance becomes $\hat{\sigma}^2(\mathbf{x}, \lambda_k)$. Then, the second stage sample size is computed as

$$N_k(\mathbf{x}) = \left\lceil \lambda_k \max\left\{ 1, \frac{\hat{\sigma}^2(\mathbf{x}, \lambda_k)}{\kappa^2 \Delta_k^4} \right\} \right\rceil, \tag{9}$$

which is derived from (6). This two-stage process is summarized in Algorithm 1.

Algorithm 1 ($\boldsymbol{N_k}(\mathbf{x})$=TwoStageLambda($\boldsymbol{\Delta_k}, \mathbf{x}, \boldsymbol{\kappa}, \mathcal{F}_k$))

Require: trust-region radius Δ_k, design point $\mathbf{x}$, and minimum sample size λ_k.

1: Set $N_{k,1}(\mathbf{x}) = \lceil \lambda_k \rceil$ and evaluate the estimate $\hat{\sigma}^2(\mathbf{x}, N_{k,1}(\mathbf{x}))$.
2: Return $N_k(\mathbf{x})$ using (9)

An obvious criticism of this strategy is the large estimation error that can result when λ_k is too small. If $\hat{\sigma}^2(\mathbf{x}, \lambda_k)$ greatly overestimates the true variance, then efficiency suffers from requiring an unnecessarily large $N_k(\mathbf{x})$ in the second stage. On the other hand, if $\hat{\sigma}^2(\mathbf{x}, \lambda_k)$ greatly underestimates the true variance, then progress may be impeded; by stopping the sampling prematurely, we may incur less communication cost, but the probability of choosing worse incumbents is heightened. Therefore, the main goal of the first stage sample size should be to attain an accurate variance estimate, motivating the next strategy.

4.2. Two-Stage Estimation Using Variance Estimates from M_k^v

To ameliorate errors in variance estimation, we double-purpose the VM described in Section 3 to derive a first stage sample size $N_{k,1}$ with the predicted variance at $\mathbf{x}$, that is,

$$N_{k,1}(\mathbf{x}) = \left\lceil \lambda_k \max\left\{1, \frac{M_k^v(\mathbf{x})}{\kappa^2 \Delta_k^4}\right\}\right\rceil. \tag{10}$$

We then compute the estimated variance $\hat{\sigma}^2(\mathbf{x}, N_{k,1}(\mathbf{x}))$ and readjust the sample size, if needed, for the second stage in a bid to meet the (8) criterion. This two-stage estimation technique is summarized in Algorithm 2.

Algorithm 2 ($\boldsymbol{N_k}(\mathbf{x})$=TwoStageVarianceModel($\Delta_k, \mathbf{x}, M_k^v, \kappa, \mathcal{F}_k$))

Require: trust-region radius Δ_k, design point $\mathbf{x}$, variance model M_k^v, and history $\mathcal{F}_k$.

1: Set $N_{k,1}(\mathbf{x})$ using (10) and evaluate the estimate $\hat{\sigma}^2(\mathbf{x}, N_{k,1}(\mathbf{x}))$.
2: Return

$$N_k(\mathbf{x}) = \max\left\{N_{k,1}(\mathbf{x}), \left\lceil \lambda_k \frac{\hat{\sigma}^2(\mathbf{x}, N_{k,1}(\mathbf{x}))}{\kappa^2 \Delta_k^4}\right\rceil\right\}.$$

This strategy presents a new challenge: a poor local model M_k^v leads to excessively large predicted variance, again leading to an unnecessary over-expense to our budget. This consideration motivates our final strategy, which hybridizes Algorithms 1 and 2.

4.3. Hybrid Two-Stage Estimation

Although our general preference is to utilize M_k^v to derive the first stage sample size (Algorithm 2) over λ_k (Algorithm 1), in cases in which M_k^v proves to be inaccurate (particularly, M_k^v overestimates $\sigma^2(\mathbf{x})$), this strategy pivots to employing λ_k many first stage samples instead. As a criterion for measuring the quality of the M_k^v prediction, we use a heuristic exploiting the assumed Lipschitz continuity of the variance function. Having a Lipschitz-continuous variance function implies that large differences (more than a constant factor times $\|\mathbf{x} - \mathbf{X}_k\|$) between $M_k^v(\mathbf{x})$ and the variance estimate at the current incumbent, $M_k^v(\mathbf{X}_k) = \hat{\sigma}^2(\mathbf{X}_k, N_{k-1})$, signals inaccuracy in prediction of variance at $\mathbf{x}$. To see why, note that, given a constant $c_\Delta > 0$, for large k (that depends on c_Δ) with probability one, we have

$$|M_k^v(\mathbf{x}) - \hat{\sigma}^2(\mathbf{X}_k, N_{k-1})| \le |M_k^v(\mathbf{x}) - \sigma^2(\mathbf{x})| + |\sigma^2(\mathbf{x}) - \sigma^2(\mathbf{X}_k)| + |\sigma^2(\mathbf{X}_k) - \hat{\sigma}^2(\mathbf{X}_k, N_{k-1})| \le (G_{\max}^v + L_v + c_\Delta)\Delta_k,$$

where $G_{\max}^v$ is an upper bound on the gradient of the variance function. The inequality follows from a Taylor expansion, the assumption of Lipschitz continuous variance, and the adaptive sample size.

Hence, if $M_k^v(\mathbf{x}) \ge \hat{\sigma}^2(\mathbf{X}_k, N_{k-1}) + c_v\Delta_k$, for some $c_v > 0$ dependent on the variance Lipschitz constant (approximating the constant coefficient of Δ_k in the above upper bound of $|M_k^v(\mathbf{x}) - \hat{\sigma}^2(\mathbf{X}_k, N_{k-1})|$), then the prediction is deemed inaccurate, and we resort to employing $N_{k,1}(\mathbf{x}) = \lambda_k$. This hybrid two-stage estimation algorithm is outlined in Algorithm 3.

Algorithm 3 ($N_k(\mathbf{x})$=TwoStageHybrid($\Delta_k, \mathbf{x}, M_k^v, \kappa, L_v, \mathcal{F}_k$))

Require: trust-region radius Δ_k, design point $\mathbf{x}$, the VM M_k^v, constant c_v, and history $\mathcal{F}_k$.

1: **if** $M_k^v(\mathbf{x}) \ge \hat{\sigma}^2(\mathbf{X}_k, N_{k-1}) + c_v\Delta_k$ **then**
2: Call Algorithm 1.
3: **else**
4: Call Algorithm 2.
5: **end if**

A primary challenge with Algorithm 3 is that L_v and $G_{\max}^v$ are unknown, requiring c_v to be determined empirically. Because the aim is to avoid using excessively large sample sizes for the initial variance estimate in (10), a problem caused by the inaccurate VM, we can set c_v as a threshold for the sample size (10) that fits within our practical budget. We suggest setting c_v larger than the approximated Lipschitz constant, considering the available computational budget. We set this threshold to 10 in our numerical experiments.

It is important to note that Algorithms 1–3 are exclusively employed for new design points $\mathbf{x}$ that have not been previously evaluated. When $\mathbf{x}$ is a design point that has been previously evaluated, a variance estimate and

initial sample sizes at $\mathbf{x}$ are already known. Whenever we reevaluate points, we obtain the second stage sample size directly as

$$N_k(\mathbf{x}) = \max\left\{N_{k-1}(\mathbf{x}), \left\lceil \lambda_k \max\left\{1, \frac{\hat{\sigma}^2(\mathbf{x}, N_{k-1}(\mathbf{x}))}{\kappa \Delta_k^4}\right\}\right\rceil\right\}, \tag{11}$$

where $N_{k-1}(\mathbf{x})$ denotes the total number of simulation oracle calls performed up until the $(k-1)$th iteration.

We remark that all our proposed two-stage sample sizes are subject to a probability of not actually satisfying the criterion in (8). Therefore, they are theoretically suboptimal compared with employing (6) to find the right sample size. We recall that, despite sacrificing the theoretical guarantees that come with determining an optimal amount of simulation effort, these two-stage sample sizes come with a practical benefit of substantially reducing communications between the classical and quantum computers.

5. ASTRO-DF Augmented by a Variance Model (ASTRO-DF-VM)

In this section, we present the algorithm ASTRO-DF-VM, which incorporates the novel components discussed in the two previous sections, namely, the VM-informed point selection (Section 3) and two-stage estimation strategy (Section 4). Pseudocode for ASTRO-DF-VM is presented in Algorithm 4. Note that, for notational brevity, the index m distinguishing ASTRO-DF-VMm correspond to which of the three distinct two-stage estimation algorithms is employed in Algorithm 4.

Algorithm 4 (ASTRO-DF-VM$\boldsymbol{m}$ for $\boldsymbol{m}\in\{1,2,3\}$)

Require: Initial incumbent $\mathbf{x}_0 \in \mathbb{R}^d$, initial and maximum trust-region radius $\Delta_0, \Delta_{\max} > 0$, model fitness thresholds $0 < \eta_1 < \eta_2 < 1$ and certification threshold $\mu > 0$, sufficient reduction constant $\theta > 0$, expansion and shrinkage constants $\gamma_1 > 1$ and $\gamma_2 \in (0,1)$, sample size lower bound sequence $\{\lambda_k\}$, adaptive sampling constant $\kappa > 0$, and a Lipschitz constant estimate for the variance function c_v.

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: *Design Set Selection:* Select $\mathcal{X}_k = \{\mathbf{X}_k^j\}_{j=0}^{2d} \subset \mathcal{B}(\mathbf{X}_k; \Delta_k)$ by calling Algorithm 5.

3: *OM Construction:* For all $j = 0, 1, \ldots, 2d$, estimate $\overline{F}(\mathbf{X}_k^j, N(\mathbf{X}_k^j))$ using the sample size determined by Algorithm m and construct the model M_k via (3).

4: *Subproblem:* Compute $\tilde{\mathbf{X}}_{k+1} \approx \arg\min_{\mathbf{x}\in\mathcal{B}(\mathbf{X}_k;\Delta_k)} M_k(\mathbf{x})$.

5: *Candidate Evaluation:* Use $\tilde{N}_{k+1} = N(\tilde{\mathbf{X}}_{k+1})$ with Algorithm m to get $\overline{F}(\tilde{\mathbf{X}}_{k+1}, \tilde{N}_{k+1})$. Define the best design point $\hat{\mathbf{X}}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}_k\setminus\mathbf{X}_k} \overline{F}(\mathbf{x}, N_k(\mathbf{x}))$, its sample size $\hat{N}_{k+1} = N(\hat{\mathbf{X}}_{k+1})$, incumbent's sample size $\hat{N}_k = N(\mathbf{X}_k)$, and reductions obtained by
- direct search: $\hat{R}_k = \overline{F}(\mathbf{X}_k, \hat{N}_k) - \overline{F}(\hat{\mathbf{X}}_{k+1}, \hat{N}_{k+1})$,
- subproblem: $\tilde{R}_k = \overline{F}(\mathbf{X}_k, \hat{N}_k) - \overline{F}(\tilde{\mathbf{X}}_{k+1}, \tilde{N}_{k+1})$,
- and the OM: $R_k = M_k(\mathbf{X}_k) - M_k(\tilde{\mathbf{X}}_{k+1})$.

6: *Update:* Set

$$(\mathbf{X}_{k+1}, N_{k+1}, \Delta_{k+1}) = \begin{cases} (\hat{\mathbf{X}}_{k+1}, \hat{N}_{k+1}, \gamma_1 \Delta_k \wedge \Delta_{\max}) & \text{if } \hat{R}_k > \max\{\tilde{R}_k, \theta\Delta_k^2\}, \\ (\tilde{\mathbf{X}}_{k+1}, \tilde{N}_{k+1}, \gamma_1 \Delta_k \wedge \Delta_{\max}) & \text{if } \tilde{R}_k \geq \eta_2 R_k \text{ and } \|\boldsymbol{G}_k\| \geq \dfrac{\Delta_k}{\mu}, \\ (\tilde{\mathbf{X}}_{k+1}, \tilde{N}_{k+1}, \Delta_k) & \text{if } \tilde{R}_k \geq \eta_1 R_k \text{ and } \|\boldsymbol{G}_k\| \geq \dfrac{\Delta_k}{\mu}, \\ (\mathbf{X}_k, \hat{N}_k, \gamma_2 \Delta_k) & \text{otherwise,} \end{cases}$$

and $k = k + 1$.

7: end for.

5.1. Construction of the Local Model M_k^v

In each iteration of ASTRO-DF-VM, the construction of M_k^v occurs prior to the selection of $\mathcal{X}_k$; hence, M_k^v is always constructed using previously evaluated points from observations in the filtration $\mathcal{F}_{k-1}$, the set of which we denote by $\mathcal{X}_k^v$. This means the deterministic coordinate basis geometry employed for constructing $\mathcal{X}_k$ cannot be used for $\mathcal{X}_k^v$. Moreover, in the early iterations of ASTRO-DF-VM, $\mathcal{X}_k^v$ may not have many points in it; an interpolation or regression with insufficient data can lead to low predictive accuracy. To address this, we propose a consistent approach for constructing M_k^v in Algorithm 5, expanding the trust region with a parameter $w > 1$ until the number of previously evaluated design points within the trust region exceeds $2d$. This condition is satisfiable

in every iteration except the zeroth iteration. In our effort to minimize the expansion of the trust region, the parameter w is set to be close to one. We construct the VM M_k^v using all the design points within the expanded trust region $\mathcal{X}_k^v$, provided $\mathsf{M}(\Phi, \mathcal{X}_k^v)$ in Definition 1 with the monomial basis Φ employed in Definition 2 has a defined pseudoinverse. If $|\mathcal{X}_k^v| = 2d+1$, we build a stochastic quadratic interpolation model with a diagonal Hessian. If $|\mathcal{X}_k^v| > 2d+1$, we construct a regression model with the same monomial basis.

In instances in which it is not possible to construct the VM M_k^v, that is, in the first iteration or when $\mathsf{M}(\Phi, \mathcal{X}_k^v)$ does not admit a pseudoinverse, we elect not to construct M_k^v at all. In such cases, we select the design set for M_k using Algorithm A.1, the same approach used in history-informed ASTRO-DF for the purpose of model construction. It is worth noting that the case in which $\mathsf{M}(\Phi, \mathcal{X}_k)$ was not pseudoinvertible very rarely occurred after the zeroth iteration in our numerical experiments. This is because the previously evaluated design points are likely well-poised as the design set $\mathcal{X}_k$ includes at least $2d-1$ new design points with a rotated coordinate basis (see the appendix).

Algorithm 5 ($[\mathcal{X}_k, M_k^v] = \mathsf{DesignSetSelection\text{-}VM}(\mathbf{X}_k, \Delta_k, \mathcal{F}_k, w, k)$)

Require: current incumbent $\mathbf{X}_k$, trust-region radius Δ_k, history $\mathcal{F}_k$, and some constants $w > 1$ and $c_d \in (0, 0.3)$.

1: Select $\mathcal{X}_k = \{\mathbf{X}_k^j\}_{j=0}^{2d} \subset \mathcal{B}(\mathbf{X}_k; \Delta_k)$ by calling Algorithm A.1.
2: **if** $k > 0$ **then**
3: Initialize $\ell = 0$
4: **repeat**
5: Set $\ell = \ell + 1$.
6: **until** $|\mathcal{X}_k^v| \geq 2d+1$.
7: **if** $\mathsf{M}(\Phi, \mathcal{X}_k^v)$ is pseudoinvertible **then**
8: *VM Subproblem:* Construct M_k^v and compute $\tilde{\mathbf{X}}_k^v \approx \arg\min_{\mathbf{x} \in \mathcal{B}(\mathbf{X}_k; \Delta_k)} M_k^v(\mathbf{x})$.
9: *Two-Stage Estimation:* Use $\tilde{N}_k^v = N(\tilde{\mathbf{X}}_k^v)$ with Algorithm m to get $\overline{F}(\tilde{\mathbf{X}}_k^v, \tilde{N}_k^v)$.
10: *Design Set Update:* Find the closest point in $\mathcal{X}_k$ to the VM minimizer $\tilde{\mathbf{X}}_k^v$, that is, $\tilde{j} = \arg\min_{j \in \{0,\ldots,2d\}} \|\mathbf{X}_k^j - \tilde{\mathbf{X}}_k^v\|$ and replace $(\mathbf{X}_k^{\tilde{j}}, \overline{F}(\mathbf{X}_k^{\tilde{j}}, N(\mathbf{X}_k^{\tilde{j}}))$ with $(\tilde{\mathbf{X}}_k^v, \overline{F}(\tilde{\mathbf{X}}_k^v, \tilde{N}_k^v))$ if $\tilde{j} \notin \{0\}$ and $\|\mathbf{X}_k^{\tilde{j}} - \tilde{\mathbf{X}}_k^v\| > c_d \Delta_k$ (ensuring that the two points are not too close).
11: **end if**
12: end if
13: **return** $[\mathcal{X}_k, M_k^v]$.

5.2. Selection of the Design Set $\mathcal{X}_k$

ASTRO-DF-VM constructs a stochastic quadratic interpolation model with a diagonal Hessian. This construction requires $2d+1$ design points. Algorithm 5 outlines the procedure for computing $\mathcal{X}_k$ motivated by a desire to reuse some previously evaluated design points but maintaining a well-poised geometry. Algorithm 5 begins with a scheme employed in history-informed ASTRO-DF, summarized in Algorithm A.1, in which $\mathbf{X}_k^1$ is chosen as a previously evaluated point within the trust region that is the farthest from $\mathbf{X}_k$, breaking ties arbitrarily. Algorithm A.1 then generates a basis of orthonormal vectors for $\mathbb{R}^d$ and completes the design set $\mathcal{X}_k$ by moving Δ_k in the positive and negative direction away from $\mathbf{X}_k$ in each orthonormal direction. If there are no previously evaluated points within the trust region, the design set is formed using the coordinate basis, that is, $\mathcal{X}_k = \mathcal{X}_k^c$ (see the appendix and Definition 2).

Upon completion of point selection, $\mathcal{X}_k$ may contain $2d$ points not previously evaluated. To keep the total number of function evaluations low, we include $\tilde{\mathbf{X}}_k^v$, the minimizer of the VM, in $\mathcal{X}_k$. We replace the point in $\mathcal{X}_k$ closest to $\tilde{\mathbf{X}}_k^v$ with $\tilde{\mathbf{X}}_k^v$, provided the selected point in $\mathcal{X}_k$ is not $\mathbf{X}_k^0$ and is not too close to $\tilde{\mathbf{X}}_k^v$. Therefore, at the termination of Algorithm 5, the set $\mathcal{X}_k$ may contain only $2d-1$ previously unevaluated points.

6. Numerical Results

We assessed and compared ASTRO-DF-VM with other SO solvers. The evaluation of SO solvers such as SPSA and Nelder–Mead was conducted using SimOpt (Eckman et al. 2023a, b). We implemented various QAOA circuits using Qiskit (Javadi-Abhari et al. 2021). All code and data for these experiments can be found in an accompanying GitHub repository (Ha et al. 2024a).

It is worth noting that, in order to facilitate an effective comparison between the solvers, we employed common random numbers (CRN) through SimOpt alongside the quantum simulator available in Qiskit. CRN is a method for variance reduction that queries the simulation-based oracle with the same random number stream.

SimOpt offers versatile capabilities for applying CRN in diverse ways, enabling us to carry out sharper (with less variance) comparisons among the solvers. Whereas CRN cannot be applied on a real quantum computer, the Qiskit quantum simulator in our numerical experiments does allow us to fix the random number seed. We reiterate that nothing in ASTRO-DF-VM assumes access to CRN streams, and this choice in experimentation was made only to reduce sources of variance in comparing across solvers, thereby ensuring that the experiments could be conducted within a reasonable time frame.

6.1. Comparison of the Two-Stage Estimation Algorithms

We start by comparing three two-stage estimation strategies via Algorithms 1–3 as indexed by ASTRO-DF-VMm in Algorithm 4.

To guarantee both robustness and accuracy, we ran each algorithm 20 times. In the numerical experiments, we use a metric for the simulation cost that takes into account both the costs related to acquiring a single sample, denoted by c_s, and the communication costs c_n. Hence, the calculation of the total simulation cost is

$$c_n Q + c_s W, \tag{12}$$

where W quantifies the total count of oracle calls (shots) requested by the solver, in which in our context $W = \sum_{k=1}^{T}\sum_{i=0}^{p} N_k^p$ with T being the number of iterations, and Q quantifies the total count of communications made with the quantum computer. In our experiments, we considered two cases: $(c_n, c_s) = (0, 1)$ and $(c_n, c_s) = (1{,}000, 1)$; we expect that the performance gap of ASTRO-DF-VM in the second case will widen further with a real quantum computer on which communications costs can be even larger.

Our first test was performed with a QAOA circuit for solving an instance of a max-cut problem (Farhi et al. 2014). Specifically, we implemented the standard QAOA with depths of 1 and 10. We now provide a brief description of the max-cut problem and implementation. Let us consider a graph $\mathcal{G}$ represented as $\mathcal{G} = [\mathcal{V}, \mathcal{D}]$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{D}$ represents the set of edges. A cut of the graph $\mathcal{G}$ is defined as a partition $(\mathcal{V}_1, \mathcal{V}_2)$ of the graph vertices such that there are no edges between $\mathcal{V}_1$ and $\mathcal{V}_2$. A maximum cut of a graph $\mathcal{G}$ is a cut such that, among all possible cuts, the number of edges between $\mathcal{V}_1$ and $\mathcal{V}_2$ is maximized. This problem can be formulated as a quadratic unconstrained binary optimization problem $\max \frac{1}{2}\sum_{(i,j)\in\mathcal{D}}(1 - x_i x_j)$, where $x_i \in \{-1, +1\}$ for $i \in \mathcal{V}$.

We begin by exploring the structure in the variance function. Our explorations suggest that, when the objective function is easy to optimize, for example, it is convex, most solvers can find the optimal solution regardless of whether the variance information is used. However, in nonconvex settings, we identify cases in which the VM can be very helpful and cases in which the VM may be harmful. To visualize these findings, we report two instances of the max-cut problem with a depth of one (see Figure 7). When the objective function is difficult to optimize (and the landscape is nearly flat), exploiting the variance information alleviates the struggle to find the global optimum (see Figure 7(a)). In addition, we observe the positive impact of the VM in high-dimensional problems although we did not include that in the visualization. When the objective function is less flat and the variance function is fairly nonconvex, using a VM may lead to a suboptimal local solution instead of the global optimum. For example, in Figure 7(b), iterates generated by ASTRO-DF can converge to a near-global optimum, such as $(2.8, 0.25)$, using the coordinate basis and direct search method. However, iterates generated by ASTRO-DF-VM might converge to a suboptimal region, such as $(2.2, 1.4)$, because of the design set incorporating the VM minimizer and the direct search method. In low dimensions, this was the only case we discovered in which the VM can be harmful. On the other hand, because low depth in QAOA contexts results in inaccurate energy approximations, the tendency is often to increase the depth and dimension of the problem, whereby the VM becomes advantageous, as is shown numerically in Section 6.2. In the test examples, the original combinatorial optimization problems have optimal values of -4 and -6, respectively. The best objective values obtained at a depth of one, however, are approximately -2 and -3.5, indicating the need to increase the depth to achieve closer approximations to these optimal values.

As a second experiment, we reconsider the noisy Himmelblau function, described in (5), with zero-mean Gaussian noise $\xi \sim \mathcal{N}(0, V(x_1, x_2))$ and four different variance functions:

i. $V_1(x_1, x_2) = |(x_1 - 3)(x_2 - 2)|$; V_1 features zero variance at (3, 2), the global minimum of the Himmelblau function.

ii. $V_2(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$; V_2 features four local minima coinciding with those of the Himmelblau function.

iii. $V_3(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + 10(x_1 - x_2)^2$; the global minimum of V_3 intentionally does not align with the global minimum of the Himmelblau function, an undesirable property for ASTRO-DF-VMs.

iv. $V_4(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + (x_1 - x_2)^2 + |x_0 - 3|$; V_4 features the same local minima and global minimum as the Himmelblau function.

Figure 7. (Color online) Illustration of the Contours of the Objective and Variance Functions with Two Distinct Graphs in Which Darker Areas Show Smaller Values

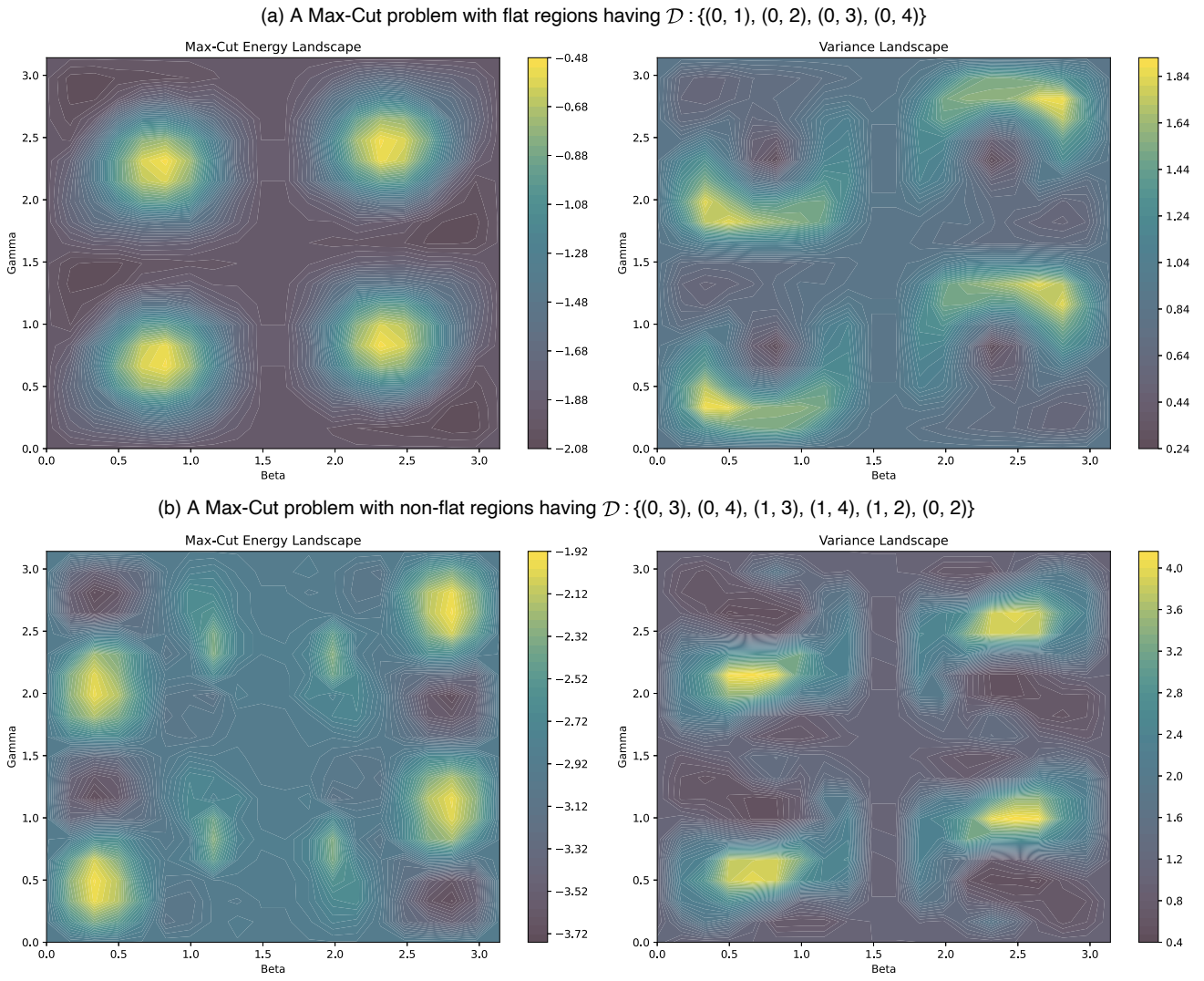


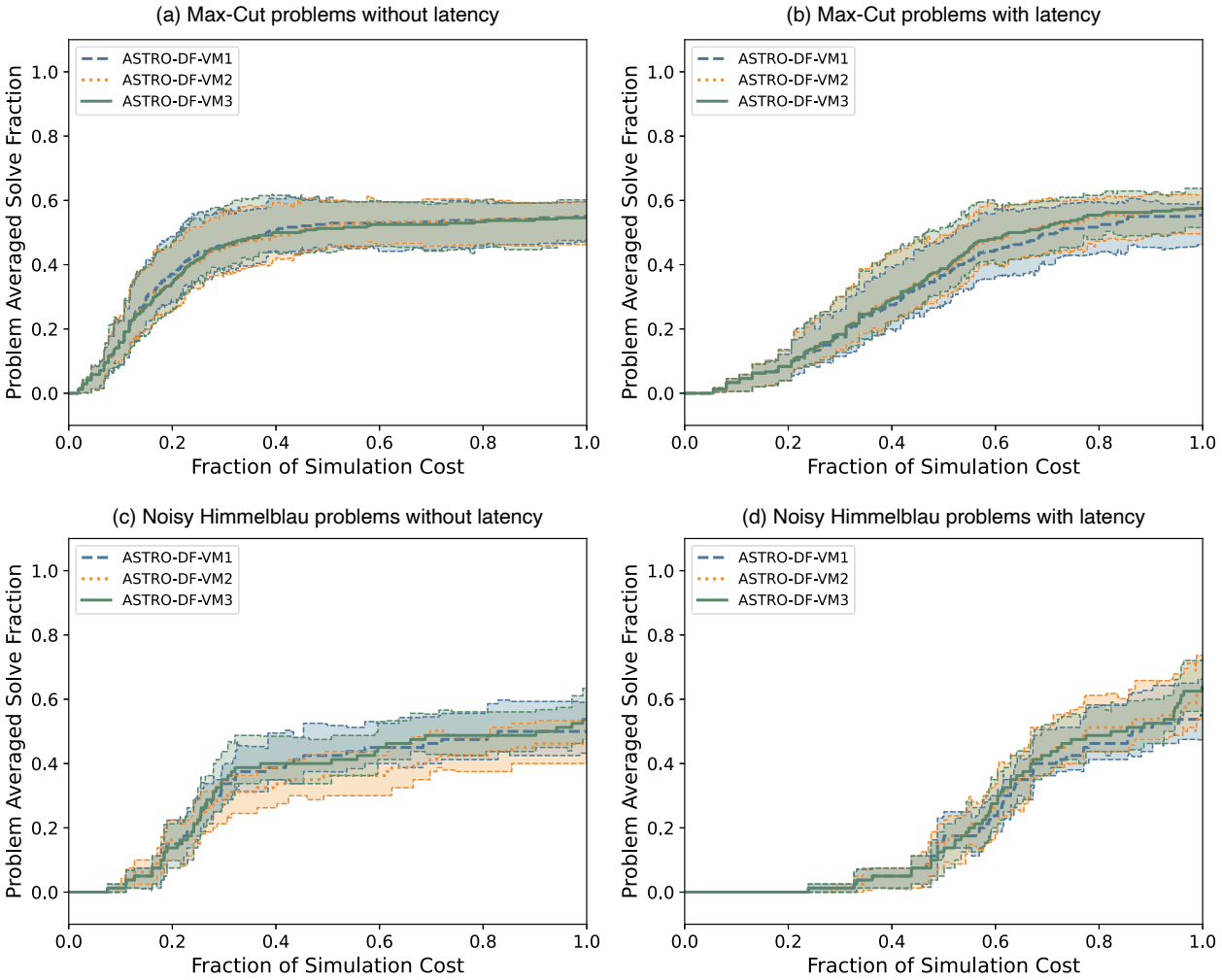
Figure 8 displays the solvability profiles for 12 instances of the max-cut problem (with varying edge structure) and four instances of the noisy Himmelblau function. Each solvability profile roughly shows the proportion of tested problems that a method can solve to within a specific relative optimality gap specified in the title of each plot (0.1 for the max-cut and 0.001 for the noisy Himmelblau, chosen appropriately based on the problem context). The use of VM in ASTRO-DF-VM2 and ASTRO-DF-VM3 slightly improves the performance when there is latency. We believe this is because, as intended, the VM reduces the likelihood of needing a second stage by selecting a good first stage sample size, whereas the inaccurate variance estimates in ASTRO-DF-VM1 often necessitate a second stage and, hence, a larger number of communications.

Before proceeding with ASTRO-DF-VM3, which uses the VM for the first stage unless its predictions appear poor based on the c_v threshold, and comparing its performance with other solvers, we examine the effect of c_v. Figure 9 suggests that the choice of c_v will not significantly affect the performance of ASTRO-DF-VM3 as long as excessively large sample sizes, which may result from an inaccurate VM, are avoided.

6.2. Comparison with Stochastic Optimization Solvers

In this section, we compare ASTRO-DF-VM3 with ASTRO-DF, SPSA, and Nelder–Mead on the two previous problems. Figures 10 and 11 show progress curves (improvement in the objective function value) on the noisy Himmelblau problems. Even when presented with a variance function that is not necessarily minimized at the

Figure 8. (Color online) Solvability Profiles of ASTRO-DF-VMm for $m \in \{1,2,3\}$ on 12 Max-Cut Problems with a Depth-1 QAOA Circuit Solved to 10% Optimality and Four Noisy Himmelblau Functions Solved to 0.1% Optimality

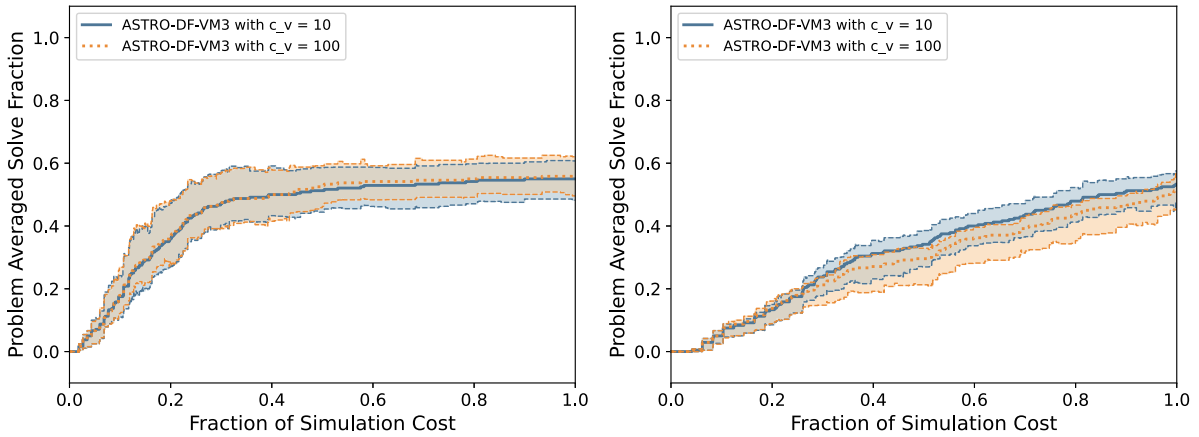


Notes. Solid lines denote mean fraction, and bands denote 95% confidence intervals. The x-axis shows the fraction of the computational burden measured by (12) with communication costs c_n and sample cost c_s, where $(c_n, c_s) = (0,1)$ without latency and $(c_n, c_s) = (1{,}000, 1)$ with latency. The y-axis shows the fraction of the solved problem.

objective's global optimum, ASTRO-DF-VM3 consistently outperforms other solvers in finding the optimal solution. In particular, with the variance function $V_3(x_1, x_2)$ whose global optimum does not align with the global optima of the objective function (even though the four local minima of the variance and objective functions coincide), ASTRO-DF-VM3 demonstrates superior performance compared with both ASTRO-DF and the Nelder-Mead method in finding the global optimum for the synthetic problem. This remains true regardless of latency (see Figures 10 and 11).

On the max-cut problems, Figure 12 displays solvability profiles (percentage of solved problem instances) in which, irrespective of communication costs, ASTRO-DF-VM3 demonstrates superior performance compared with other solvers in finding optimal solutions. More specifically, Nelder–Mead and SPSA tend to become stuck at solutions significantly distant from the global optimum, and ASTRO-DF is limited in the number of iterations (steps) it can perform because of the substantial communication costs. Interestingly, even with no latency costs, we still observe ASTRO-DF-VM3 finding better solutions than ASTRO-DF. We attribute this to the assistance the VM provides in finding better design sets that either help construct better models or are used in a direct search regime (as elaborated in Section 3) to guide progress toward the global solution. For a closer look, we compare ASTRO-DF-VM3 and ASTRO-DF on one instance of the max-cut problem tracked per iteration and per expended budget without latency. As shown by Figure 13, ASTRO-DF-VM3 expends a similar number of samples per

Figure 9. (Color online) Solvability Profiles and 95% Confidence Intervals of ASTRO-DF-VM3s with $c_v = 10$ and $c_v = 100$ on 12 Max-Cut Problems with a Depth-1 (Left) and Depth-10 (Right) QAOA Circuit Solved to 10% Optimality



Note. The computational burden (simulation cost) is measured by (12) with $c_n = 0$ and $c_s = 1$.

iteration (and a similar number of iterations) to ASTRO-DF. Therefore, its ability to reach better solutions compared with ASTRO-DF (as shown in Figure 12) should be solely attributed to the VM's aid in design set selection.

The advantage of ASTRO-DF-VM3 is amplified in the presence of latency costs as it will also benefit from fewer communications with the quantum computer. That is, as communication costs escalate, the performance gap between ASTRO-DF-VM3 and other solvers widens.

Figure 10. (Color online) Mean Performance and 95% Confidence Interval on the Noisy Himmelblau Function Without Latency

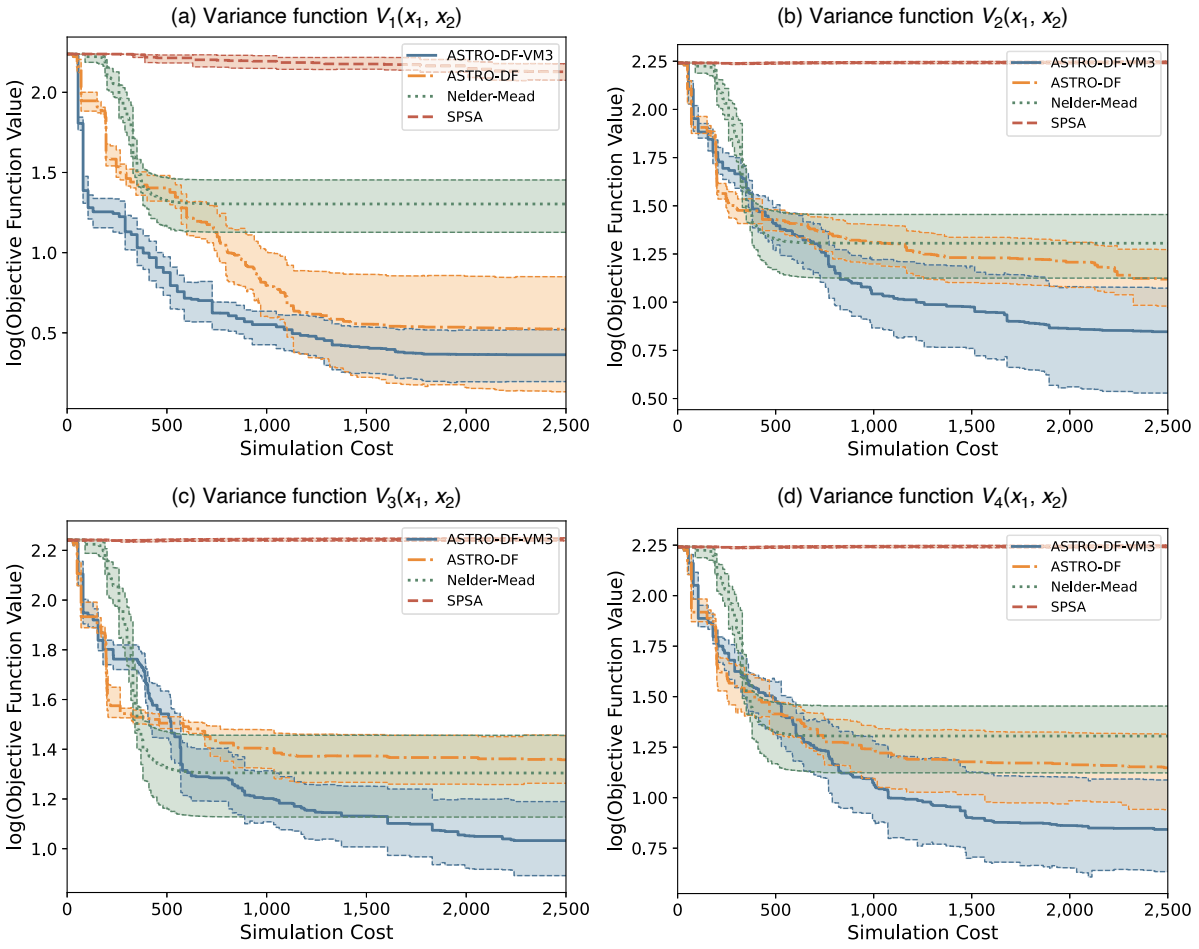
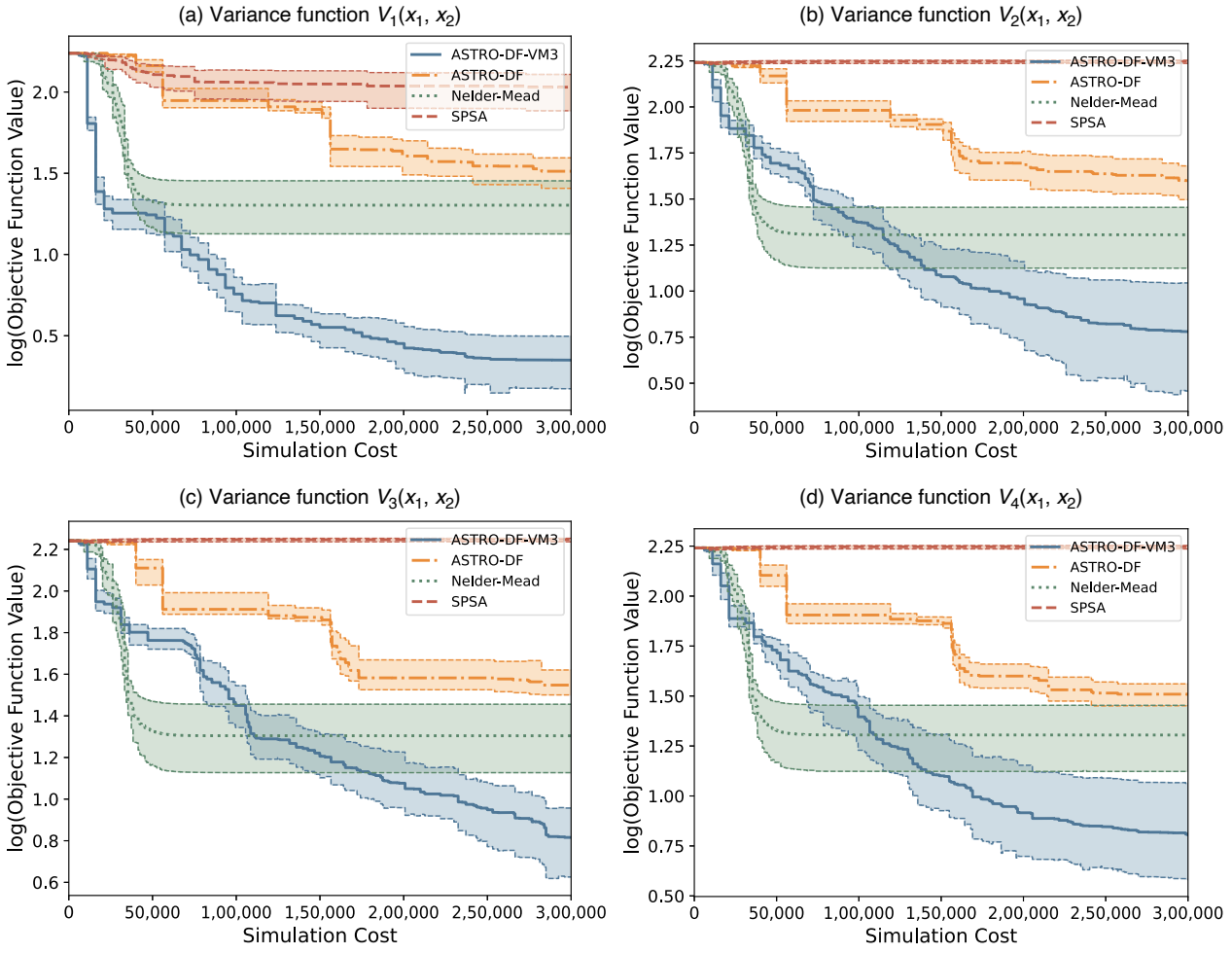


Figure 11. (Color online) Mean Performance and 95% Confidence Interval on the Noisy Himmelblau Function with Latency



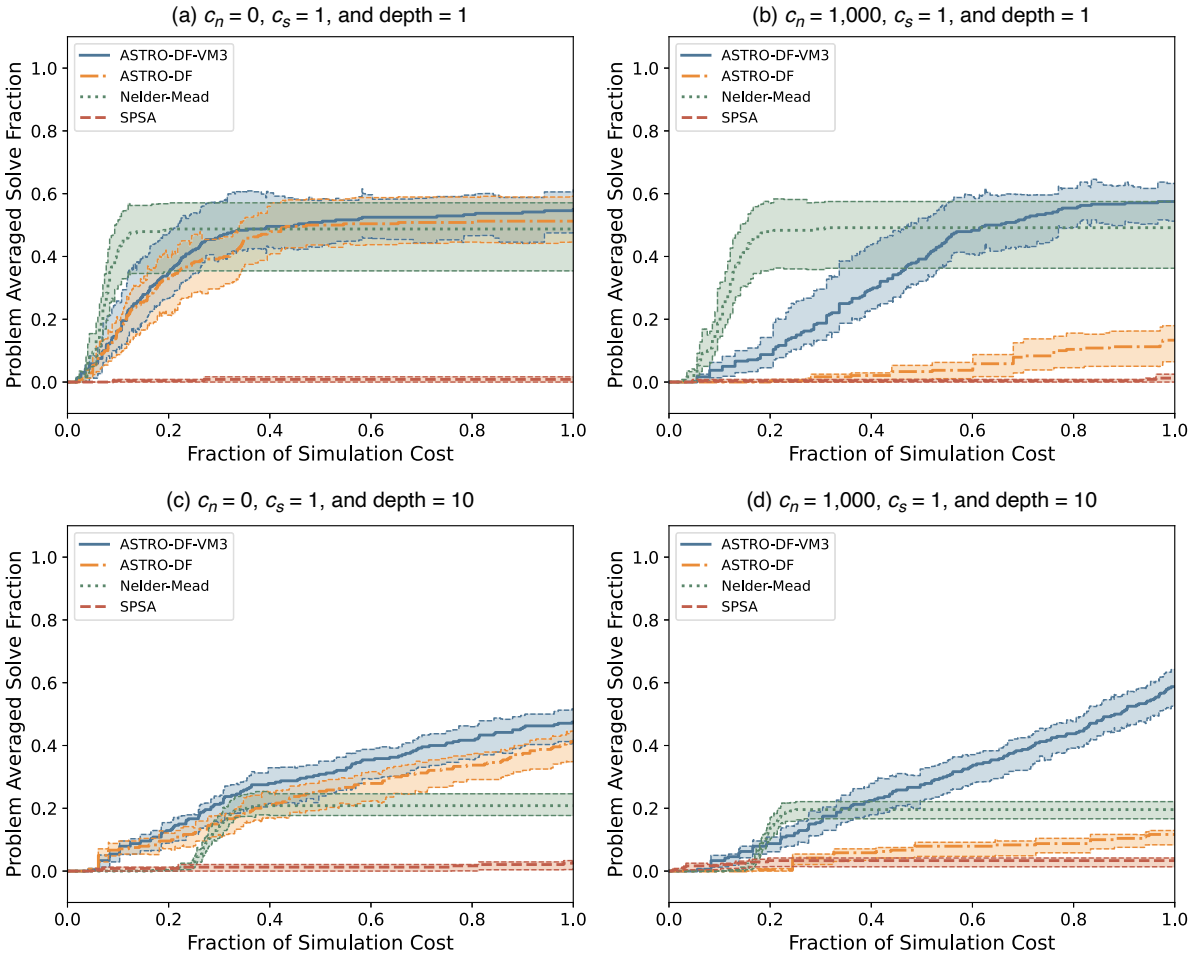
6.3. Comparison on a Regularized Function

Another natural way to utilize the zero-variance principle is to minimize the regularized objective function $f(\mathbf{x}) + c_p\sigma^2(\mathbf{x})$ for some positive constant c_p. In this section, we discuss the comparison between ASTRO-DF-VM3 and the original algorithm of ASTRO-DF on the regularized function. Figure 14 illustrates that ASTRO-DF-VM3 shows better performance on both depth-1 and depth-10 circuits. One should expect this overperformance gap to widen as the communication costs c_n increases (depicted by Figure 12). This suggests that using the VM is a better strategy than adding a regularization.

7. Conclusion

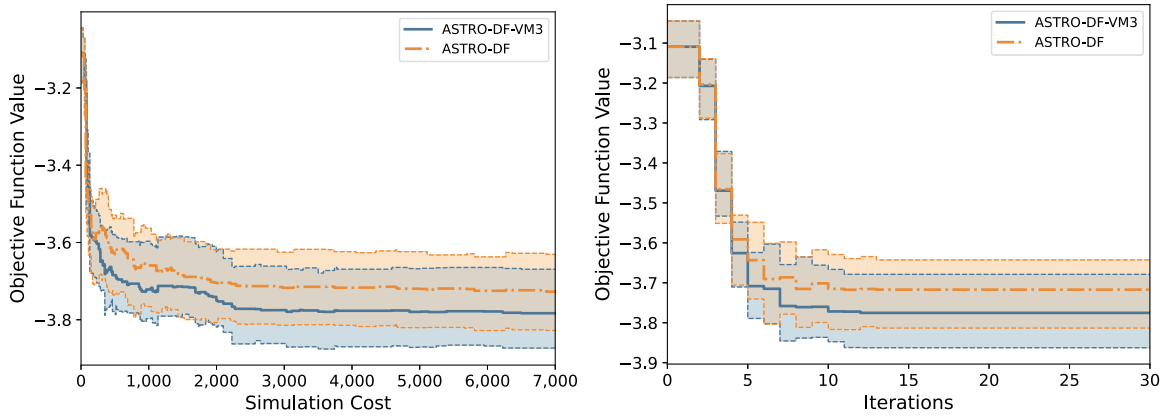
In the optimization problem that forms an essential part of VQAs, two phenomena come to the forefront. The first pertains to latency, which leads to an increase in the time required to acquire a single sample of shots. Given its substantial computational overhead during the optimization process, it is imperative for the optimization solver to be thoughtfully designed with a focus on minimizing the quantum computer access frequency. The second phenomenon is the gradual reduction of true variance that accompanies the decaying optimality gap. In this paper, we introduce a novel stochastic trust region method to tackle VQA optimization problems, leveraging the distinctive characteristics of diminishing variance and communication costs. We name this method ASTRO-DF-VM. To leverage the characteristic of diminishing variance, ASTRO-DF-VM constructs a second local model, the VM, on variance using variance estimates at points previously evaluated. The minimizer of the VM is included in the design set, helping to find better solutions at each iteration and the global optimum. Moreover, to reduce

Figure 12. (Color online) Solvability Profiles for 12 Instances with Varied Edges on the Max-Cut Problem, Utilizing 20 Macroreplications and 95% Confidence Intervals Solved to 10% Optimality



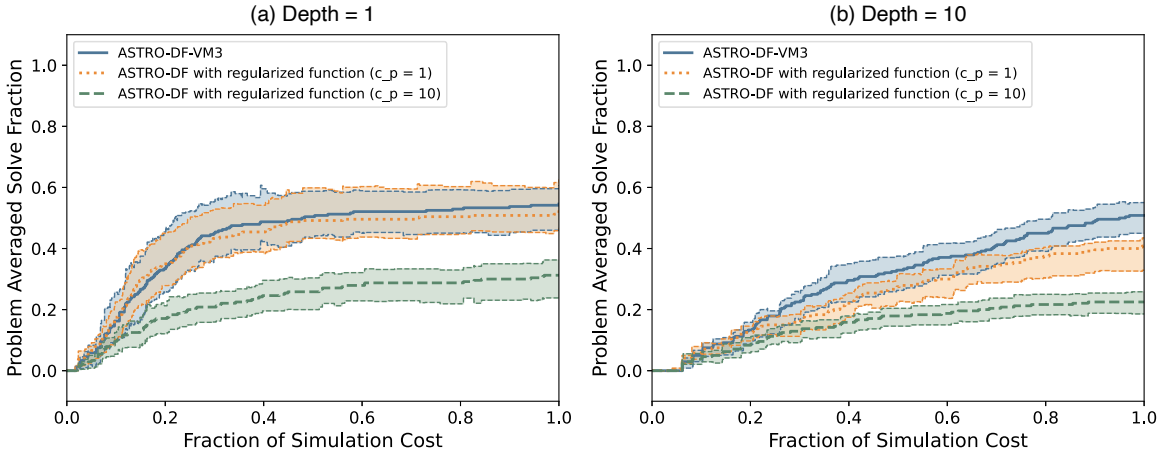
communication costs, ASTRO-DF-VM restricts the access to the quantum computer to twice per design point at each iteration. Our numerical results showcase the effectiveness of ASTRO-DF-VM on two problems from QAOA. Even in scenarios without communication costs, the use of VM leads to outpacing other solvers in terms of finding quality solutions more efficiently. Moreover, as communication costs increase, the performance gap

Figure 13. (Color online) Finite Time Performance on the Max-Cut Problem with Edges = {[0,1],[0,2],[1,4],[2,3],[2,5],[3,5]}



Notes. The x-axis shows the number of iterations (right figure) or the simulation cost measured by (12) with $c_n = 0$ and $c_s = 1$ (left figure). The y-axis shows the objective function value.

Figure 14. (Color online) Solvability Profiles for 12 Max-Cut Problems of ASTRO-DF-VM3 on the Original Objective Function and ASTRO-DF on the Regularized Function with $c_p = 1, 10$, Utilizing 20 Macroreplications and 95% Confidence Intervals Solved to 10% Optimality



between ASTRO-DF-VM and other solvers widens, further highlighting its advantage in latency-constrained settings.

Appendix A. Algorithm for Selecting $\mathcal{X}_k$ in History-Informed ASTRO-DF

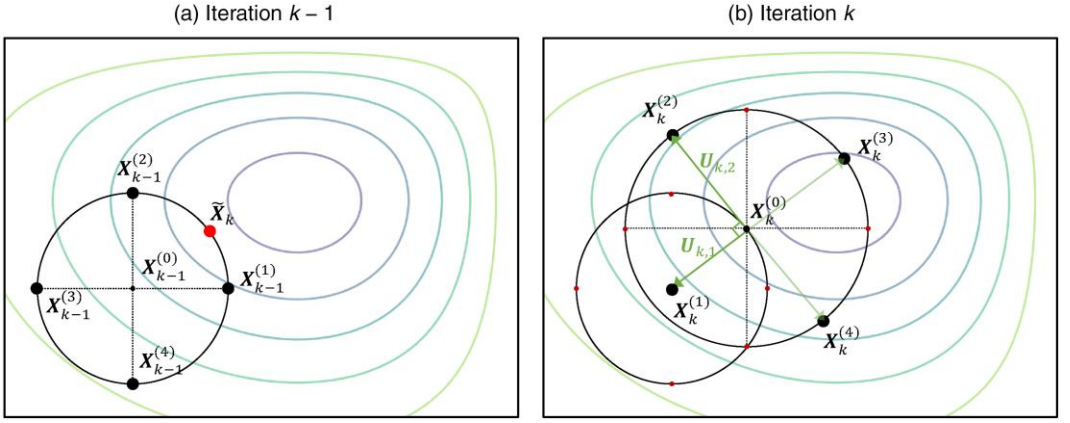
Algorithm A.1 shows the pseudocode for selecting the design set at iteration k in history-informed ASTRO-DF (Ha and Shashaani 2023b). The example illustrating the application of Algorithm A.1 is detailed in Figure A.1.

Algorithm A.1 ($\mathcal{X}_k = \text{DesignSetSelection}(\Delta_k, \mathbf{X}_k, \mathcal{F}_k)$)

Require: trust-region radius Δ_k, incumbent $\mathbf{X}_k$.

1: Find all previously evaluated design points within $\mathcal{B}(\mathbf{X}_k, \Delta_k)$; denote it $\mathcal{R}_k$.
2: **if** $\mathcal{R}_k = \{\mathbf{X}_k\}$ **then**
3: Select the design set $\mathcal{X}_k = \mathcal{X}_k^c$ following Definition 2.
4: else
5: Find $\mathbf{X}_k^1 = \arg\max_{\mathbf{x}\in\mathcal{R}_k} \|\mathbf{X}_k - \mathbf{x}\| = \mathbf{X}_k + P_k \boldsymbol{U}_{k,1}$, where $\|\boldsymbol{U}_{k,1}\| = 1$ and $P_k = \|\mathbf{X}_k^1 - \mathbf{X}_k^0\|$.
6: Compute a set of $d-1$ vectors $\{\boldsymbol{U}_{k,2}, \ldots, \boldsymbol{U}_{k,d}\}$ mutually orthonormal to $\boldsymbol{U}_{k,1}$
7: Set $\mathcal{X}_k = \{\mathbf{X}_k, \mathbf{X}_k + P_k\boldsymbol{U}_{k,1}, \mathbf{X}_k + \Delta_k\boldsymbol{U}_{k,2}, \ldots, \mathbf{X}_k + \Delta_k\boldsymbol{U}_{k,d}, \mathbf{X}_k - \Delta_k\boldsymbol{U}_{k,1}, \ldots, \mathbf{X}_k - \Delta_k\boldsymbol{U}_{k,d}\}$.
8: end if
9: **Return** $\mathcal{X}_k$.

Figure A.1. (Color online) An Example Demonstrating the Application of the Rotated Coordinate Basis (Ha and Shashaani 2023b)



Notes. (a) The coordinate basis using a coordinate system defined by elementary basis vectors, which is history-informed ASTRO-DF's default coordinate basis in the absence of reusable design points within the trust region. (b) A rotated coordinate basis. In this case, in the kth iteration of history-informed ASTRO-DF, the design point $\mathbf{X}_{k-1}^0$ is the farthest from $\mathbf{X}_k^0$ among all previously evaluated design points, and so we choose $\mathbf{X}_k^1 = \mathbf{X}_{k-1}^0$. Orthogonalizing against $\boldsymbol{U}_{k,1} := \mathbf{X}_k^0 - \mathbf{X}_k^1$ deterministically defines the rotated coordinate basis in the kth iteration.

References

Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. *Nature* 549(7671):195–202.

Bruzewicz CD, Chiaverini J, McConnell R, Sage JM (2019) Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.* 6(2):021314.

Cao L, Berahas AS, Scheinberg K (2024) First- and second-order high probability complexity bounds for trust-region methods with noisy oracles. *Math. Programming* 207(1):55–106.

Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, et al. (2021) Variational quantum algorithms. *Nature Rev. Phys.* 3:625–644.

Chang KH, Hong LJ, Wan H (2013) Stochastic trust-region response-surface method (strong)—A new response-surface framework for simulation optimization. *INFORMS J. Comput.* 25(2):230–243.

Chen R, Menickelly M, Scheinberg K (2018) Stochastic optimization using a trust-region method and random models. *Math. Programming* 169(2):447–487.

Clark SM, Lobser D, Revelle MC, Yale CG, Bossert D, Burch AD, Chow MN, et al. (2021) Engineering the quantum scientific computing open user testbed. *IEEE Trans. Quantum Engrg.* 2:1–32.

Cloët IC, Dietrich MR, Arrington J, Bazavov A, Bishof M, Freese A, Gorshkov AV, et al. (2019) Opportunities for nuclear physics & quantum information science. Preprint, submitted March 13, https://arxiv.org/abs/1903.05453.

Conn AR, Scheinberg K, Vicente LN (2009) *Introduction to Derivative-Free Optimization*, 1st ed. (Society for Industrial and Applied Mathematics, Philadelphia).

Eckman DJ, Henderson SG, Shashaani S (2023a) Diagnostic tools for evaluating and comparing simulation-optimization algorithms. *INFORMS J. Comput.* 35(2):350–367.

Eckman DJ, Henderson SG, Shashaani S (2023b) Simopt: A testbed for simulation-optimization experiments. *INFORMS J. Comput.* 35(2):495–508.

Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. Preprint, submitted November 14, https://arxiv.org/abs/1411.4028.

Gambetta J, Braff W, Wallraff A, Girvin S, Schoelkopf R (2007) Protocols for optimal readout of qubits using a continuous quantum nondemolition measurement. *Phys. Rev. A* 76(1):012325.

Ha Y, Shashaani S (2023a) Iteration complexity and finite-time efficiency of adaptive sampling trust-region methods for stochastic derivative-free optimization. Preprint, submitted May 18, https://arxiv.org/abs/2305.10650.

Ha Y, Shashaani S (2023b) Toward greener stochastic derivative-free optimization with trust regions and adaptive sampling. Corlu CG, Hunter SR, Lam H, Onggo BS, Shortle J, Biller B, eds. *Proc. 2023 Winter Simulation Conf.* (Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ), 3457–3467.

Ha Y, Shashaani S, Menickelly M (2024a) Two-stage estimation and variance modeling for latency-constrained variational quantum algorithms. http://dx.doi.org/10.1287/ijoc.2024.0575.cd, https://github.com/INFORMSJoC/2024.0575.

Ha Y, Shashaani S, Pasupathy R (2024b) Complexity of zeroth-and first-order stochastic trust-region algorithms. Preprint, submitted May 30, https://arxiv.org/abs/2405.20116.

Javadi-Abhari A, Treinish M, Krsulich K, Wood CJ, Lishman J, Gacon J, Martiel S, et al. (2024) Quantum computing with Qiskit. Preprint, submitted May 14, https://arxiv.org/abs/2405.08810.

Kokail C, Maier C, van Bijnen R, Brydges T, Joshi MK, Jurcevic P, Muschik CA, et al. (2019) Self-verifying variational quantum simulation of lattice models. *Nature* 569(7756):355–360.

Lanyon BP, Whitfield JD, Gillett GG, Goggin ME, Almeida MP, Kassal I, Biamonte JD, et al. (2010) Toward quantum chemistry on a quantum computer. *Nature Chemistry* 2(2):106–111.

Menickelly M, Ha Y, Otten M (2023) Latency considerations for stochastic optimizers in variational quantum algorithms. *Quantum* 7:949.

Mitarai K, Negoro M, Kitagawa M, Fujii K (2018) Quantum circuit learning. *Phys. Rev. A* 98(3):032309.

O'Malley PJ, Babbush R, Kivlichan ID, Romero J, McClean JR, Barends R, Kelly J, et al. (2016) Scalable quantum simulation of molecular energies. *Phys. Rev. X* 6(3):031007.

Orus R, Mugel S, Lizaso E (2019) Quantum computing for finance: Overview and prospects. *Rev. Phys.* 4:100028.

Otten M, Cortes CL, Gray SK (2019) Noise-resilient quantum dynamics using symmetry-preserving ansatzes. Preprint, submitted October 14, https://arxiv.org/abs/1910.06284.

Otten M, Goumiri IR, Priest BW, Chapline GF, Schneider MD (2020) Quantum machine learning using Gaussian processes with performant quantum kernels. Preprint, submitted April 23, https://arxiv.org/abs/2004.11280.

Powell MJD (2009) The BOBYQA algorithm for bound constrained optimization without derivatives. Cambridge NA Report NA2009/06, University of Cambridge, Cambridge, UK.

Preskill J (1998) Fault-tolerant quantum computation. Lo H-K, Popescu S, Spiller T, eds. *Introduction to Quantum Computation and Information* (World Scientific, Singapore), 213–269.

Preskill J (2018) Quantum computing in the NISQ era and beyond. *Quantum* 2:79.

Ragonneau TM, Zhang Z (2023) An optimal interpolation set for model-based derivative-free optimization methods. Preprint, submitted February 20, https://arxiv.org/abs/2302.09992.

Shashaani SS, Hashemi F, Pasupathy R (2018) ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization. *SIAM J. Optim.* 28(4):3145–3176.

Smith A, Kim M, Pollmann F, Knolle J (2019) Simulating quantum many-body dynamics on a current digital quantum computer. *Quantum Inform.* 5(1):1–13.

Sun S, Nocedal J (2023) A trust region method for noisy unconstrained optimization. *Math. Programming* 202(1):445–472.

Sung KJ, Yao J, Harrigan MP, Rubin NC, Jiang Z, Lin L, Babbush R, McClean JR (2020) Using models to improve optimizers for variational quantum algorithms. *Quantum Sci. Tech.* 5(4):044008.

Yuan X, Endo S, Zhao Q, Li Y, Benjamin SC (2019) Theory of variational quantum simulation. *Quantum* 3:191.

Zhang DB, Chen BL, Yuan ZH, Yin T (2022) Variational quantum eigensolvers by variance minimization. *Chinese Phys. B* 31(12):120301.