

Practical Optimization for Hybrid Quantum–Classical Algorithms

Gian G. Guerreschi and Mikhail Smelyanskiy — Intel Labs, California, USA

Sumit Tapas Chongder

Supervisor: Dr. Kirankumar Hiremath

M.Tech in Quantum Technologies

Roll No.: M25IQT013

Course: Quantum–Inspired Optimization

Indian Institute of Technology Jodhpur

Paper Overview

Objective:

- Investigate practical optimization techniques for hybrid quantum–classical algorithms.
- Focus on QAOA under finite sampling precision on NISQ devices.
- Evaluate trade-offs between precision and repetition cost.

Key Contributions:

- Introduces a unified cost metric: *repetition cost*.
- Benchmarks classical optimizers: gradient-based vs. derivative-free.
- Analyzes convergence, efficiency, and robustness via numerical simulations.

Techniques Used:

- Nelder–Mead (NM): gradient-free
- BFGS with Finite Difference (FD): numerical gradient
- BFGS with Analytical Gradient (AG): circuit-based gradient

Evolution of Hybrid Quantum–Classical Computing

Evolution of Hybrid Quantum Systems

- Since the early 2000s, quantum computing moved from theory to lab-scale prototypes, enabling real experimentation.
- By 2015, hybrid algorithms like **VQE** and **QAOA** began leveraging NISQ devices by combining quantum circuits with classical optimization.

Industry Adoption and Research

- Tech leaders: **IBM (Qiskit Runtime)**, **Google (TensorFlow Quantum)**, **Intel (Spin Qubits)**, **Microsoft (Azure Quantum, Q#)** — offer hybrid platforms, orchestration tools, and scalable hardware.
- Providers like **Amazon Braket**, **Rigetti**, **IonQ**, and **D-Wave** deliver hybrid-ready access via cloud APIs.

Current Focus

- Advancing hybrid algorithms for real-world tasks in chemistry, optimization, and ML by improving noise resilience and minimizing repetition cost through variational techniques.
- Integrating quantum workflows into classical HPC and cloud infrastructure.

Hybrid Quantum–Classical Algorithms

Three Core Steps:

- ➊ **State Preparation (Quantum Hardware)** Apply a parameterized sequence of gates γ_n on an initial reference state.
- ➋ **Measurement (Objective Evaluation)** Measure the quantum state and compute the expectation value of the observable \hat{C} .
- ➌ **Parameter Update (Classical Optimization)** Use measured results to update parameters γ for the next iteration.

This cycle repeats until convergence or a satisfactory solution is reached.

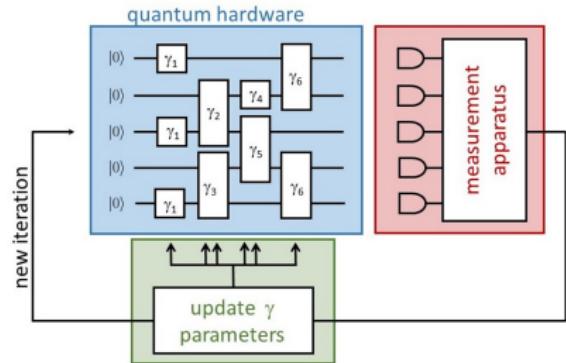


FIG. 1. Iterative cycle of hybrid quantum–classical algorithms: State preparation (blue), Measurement (red), and Optimization (green).

Hybrid Quantum–Classical Algorithms

1. Parameterized State Preparation

Quantum state is prepared via sequential unitary layers with tunable parameters.

$$\hat{W}_1^p(\gamma) = \hat{U}_p(\gamma_p) \cdots \hat{U}_2(\gamma_2) \hat{U}_1(\gamma_1), \quad |\gamma\rangle = \hat{W}_1^p(\gamma)|\phi_0\rangle$$

2. Objective Function (Quantum Observable)

Evaluates the expectation value of a cost Hamiltonian \hat{C} composed of measurable operators.

$$\hat{C} = \sum_{\nu=1}^{k_C} c_\nu \hat{C}_\nu, \quad F_p(\gamma) = \langle \gamma | \hat{C} | \gamma \rangle = \sum_{\nu=1}^{k_C} c_\nu \langle \gamma | \hat{C}_\nu | \gamma \rangle$$

3. Finite-Precision Estimator and Repetition Cost

Statistical sampling: higher repetitions (M) yield smaller uncertainty (ϵ) but greater cost.

$$F_{p,\epsilon}(\gamma) \approx F_p(\gamma) \pm \epsilon, \quad M \geq \frac{\text{Var}[\hat{C}]}{\epsilon^2} = \frac{\sum_{\nu=1}^{k_C} c_\nu^2 \text{Var}[\hat{C}_\nu]}{\epsilon^2}$$

Interpretation: Repetition cost M scales inversely with ϵ^2 , defining the trade-off between precision and runtime in hybrid optimization.

Nelder–Mead (NM) Optimization

- A **derivative-free optimization method** that evaluates the cost function at $(2p + 1)$ points, forming a simplex in parameter space.
- Iteratively updates the vertex with the **lowest objective value** through reflection, expansion, or contraction to approach the minimum.
- **Well-suited for noisy QAOA settings** where gradients are unreliable or costly to estimate.

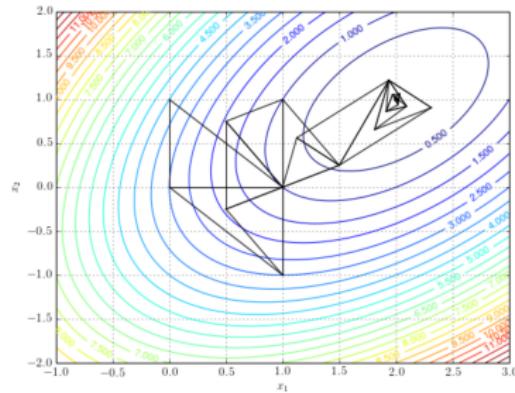


FIG. 2. Simplex evolution in parameter space during Nelder–Mead optimization for noisy QAOA.

Credit: Jakub Konka (2013).

Limitations of Nelder–Mead (NM) Optimization

Despite its simplicity, Nelder–Mead can fail in certain scenarios. It may converge to a local minimum instead of the global one, or fail to converge at all when no minimum exists within the search region.

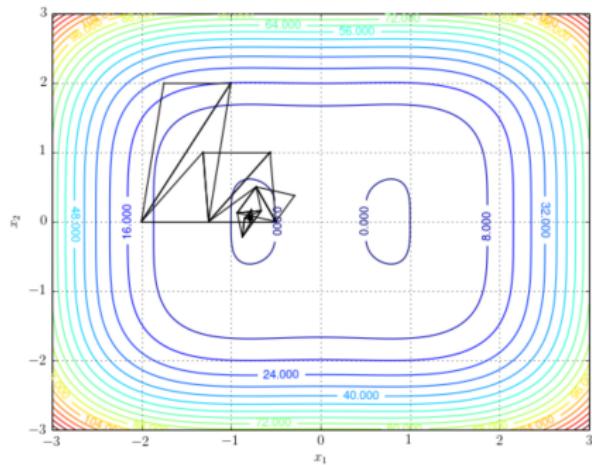


FIG. 3. Multiple Minima: The simplex may get trapped in a local basin, failing to reach the global minimum.

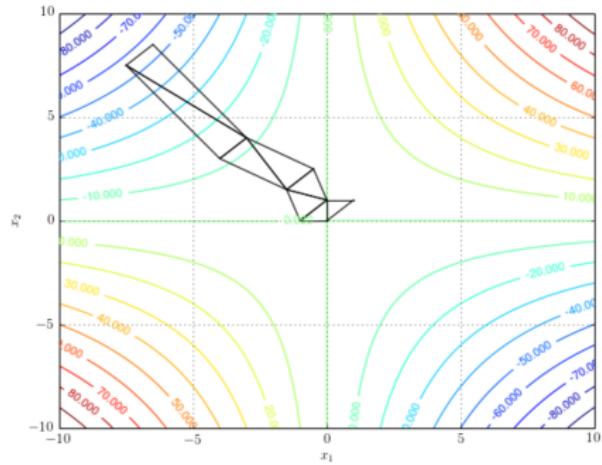


FIG. 4. No Minimum: The simplex may drift indefinitely when the objective function is unbounded or ill-conditioned.

Broyden–Fletcher–Goldfarb–Shanno Optimization

Overview: Broyden–Fletcher–Goldfarb–Shanno (**BFGS**) is a *quasi-Newton* optimization method that refines the search direction by building an approximate inverse Hessian, achieving faster convergence than standard Gradient Descent.

Key Idea: Rather than computing the exact second derivatives (as in Newton's method), BFGS updates a Hessian approximation using gradient and position differences between successive steps.

Update Rule:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$$

Where:

- x_k — parameter vector at iteration k
- $f_k = f(x_k)$ — objective (cost) function value
- $s_k = x_{k+1} - x_k$ — change in parameters
- $y_k = \nabla f_{k+1} - \nabla f_k$ — change in gradient
- B_k — approximation of the Hessian matrix at step k

Advantages:

- Uses curvature information for faster convergence than Gradient Descent.
- Avoids explicit second derivatives unlike Newton's Method.
- Efficient and stable for smooth cost landscapes in QAOA optimization.

BFGS with Finite-Difference (FD) Derivatives

In hybrid quantum–classical optimization, analytical gradients are often inaccessible due to noisy measurements. The Finite-Difference (FD) variant of BFGS estimates gradients numerically from nearby evaluations of the cost function.

Working Principle:

- BFGS builds a quadratic model of the objective function $F_p(\gamma)$ and updates parameters iteratively.
- The gradient components $\nabla_\gamma F_p$ are approximated by evaluating F_p at slightly shifted parameter values.
- This approach allows gradient-based optimization without direct derivative computation.

Trade-Offs:

- *Pros:* Enables use of BFGS where analytical gradients are unavailable.
- *Cons:* Sensitive to step size δ and sampling noise ϵ' — small δ improves accuracy but increases cost.
- Achieving stable convergence requires balancing accuracy ($O(\delta^2)$) and precision ($O(\frac{\epsilon'}{\delta})$).

BFGS with Finite-Difference (FD) Derivatives

1. Central Finite-Difference Gradient Estimate

$$\frac{\partial F_p(\gamma)}{\partial \gamma_n} \approx \frac{F_p(\gamma_1, \dots, \gamma_n + \frac{\delta}{2}, \dots) - F_p(\gamma_1, \dots, \gamma_n - \frac{\delta}{2}, \dots)}{\delta}$$

This approximates $\nabla_{\gamma} F_p$ by comparing function values at symmetric points separated by δ .

2. Error Contributions

$$\frac{\partial F_{p,\epsilon'}(\gamma)}{\partial \gamma_n} = \frac{\partial F_p(\gamma)}{\partial \gamma_n} + O(\delta^2) + O\left(\frac{\epsilon'}{\delta}\right)$$

- $O(\delta^2)$ — truncation error (accuracy)
- $O(\frac{\epsilon'}{\delta})$ — stochastic noise from finite sampling precision ϵ'

3. Repetition Cost for Desired Precision

$$M \geq \frac{\text{Var}[\hat{C}]_{\gamma_n + \delta/2} + \text{Var}[\hat{C}]_{\gamma_n - \delta/2}}{2(\epsilon')^2} \approx \frac{\text{Var}[\hat{C}]_{\gamma_n}}{(\epsilon')^2}$$

Interpretation: To achieve smaller uncertainty ϵ' , the required number of measurements M grows as $1/(\epsilon')^2$, defining a precision–cost trade-off central to hybrid quantum optimization.

BFGS with Analytical Gradient (AG)

- Unlike finite-difference (FD) methods, analytical gradients compute exact derivatives — eliminating bias due to discretization.
- The gradient components are derived from the circuit's internal parameters using the generators of the unitaries:

$$\hat{U}_n(\gamma_n) = e^{-i\gamma_n \hat{G}_n}, \quad \hat{W}_n^k(\gamma) = \hat{U}_k(\gamma_k) \cdots \hat{U}_n(\gamma_n)$$

- The derivative of the cost function is expressed as:

$$\frac{\partial F_p(\gamma)}{\partial \gamma_n} = -2 \sum_{\mu=1}^{k_G} \sum_{\nu=1}^{k_C} g_\mu c_\nu \Im \left[\langle \gamma | \hat{W}_n^p \hat{\sigma}_\mu^{(G)} \hat{W}_n^{p\dagger} \hat{\sigma}_\nu^{(C)} | \gamma \rangle \right]$$

- This analytical form captures the true response of the quantum state to parameter variations, avoiding noisy finite-difference estimates.

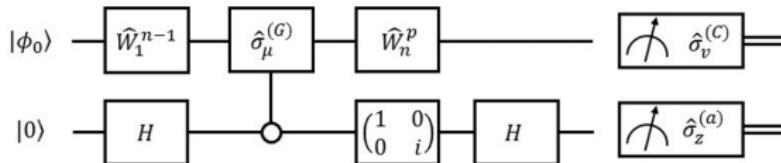


Fig. 5. Quantum circuit for analytical gradient evaluation using ancilla-assisted interference.

Analytical Gradient Evaluation using Qiskit

Overview: The circuit employs an ancilla qubit to extract each gradient component via interference:

$$-\Im \langle \gamma | \hat{W}_n^\dagger \hat{O}_\mu^\dagger \hat{W} \hat{\rho}^\dagger \hat{W} \hat{O}_\mu | \gamma \rangle$$

- QAOA and VQE particularly benefit, since \hat{C} and \hat{G}_n are typically Pauli operators and hence directly measurable.
- The repetition cost for achieving precision ϵ'' scales as:

$$M \geq \frac{4}{(\epsilon'')^2} \sum_{v=1}^{k_C} \sum_{\mu=1}^{k_G} g_{\mu v}^2 c_v^2 \text{Var}[\hat{C}_v]$$

This gradient method provides accurate gradients with lower sampling overhead than finite-difference methods.

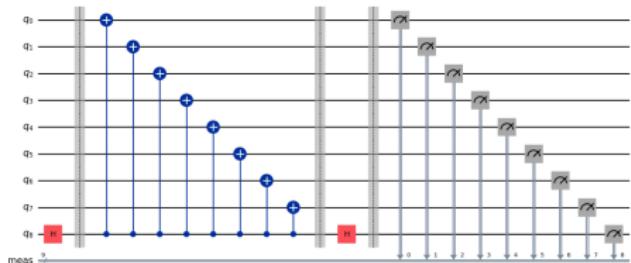


FIG. 6. Analytical gradient circuit for a 6-node 3-regular graph with four QAOA layers (Qiskit).

Quantum Approximate Optimization Algorithm (QAOA)

QAOA is a hybrid quantum–classical algorithm designed to solve **constraint satisfaction problems (CSPs)**.

- A classical objective function $C(z_1, \dots, z_N)$ is mapped to a quantum cost Hamiltonian:

$$\hat{C}(\hat{\sigma}_z^{(1)}, \dots, \hat{\sigma}_z^{(N)})$$

- The algorithm alternates two unitaries:
 $\hat{U}(\gamma) = e^{-i\gamma\hat{C}}$, $\hat{V}(\beta) = e^{-i\beta\hat{B}}$, $\hat{B} = \sum_{i=1}^N \hat{\sigma}_x^{(i)}$

Output: Starting from the uniform superposition $|s\rangle = |+\rangle^{\otimes N}$, a p -layer circuit prepares the state:

$$|\gamma, \beta\rangle = \hat{V}(\beta_p)\hat{U}(\gamma_p) \cdots \hat{V}(\beta_1)\hat{U}(\gamma_1)|s\rangle$$

The expectation value gives the average number of satisfied clauses.

$$F_p(\gamma, \beta) = \langle \gamma, \beta | \hat{C} | \gamma, \beta \rangle$$

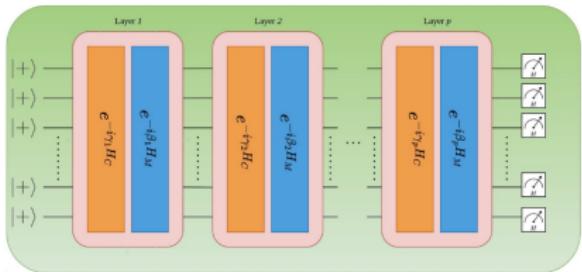


FIG. 7. QAOA circuit structure with alternating cost and mixer layers.

QAOA: Optimization and Key Insights

Optimization Goal: Maximize the cost function by tuning variational parameters using classical optimizers such as BFGS and Nelder-Mead:

$$F_p(\gamma, \beta) = \langle \gamma, \beta | \hat{C} | \gamma, \beta \rangle$$

Analytical Gradients:

- Gradients are derived to guide optimization:

$$\frac{\partial F_p}{\partial \gamma_n} = -2 \sum_{\mu} c_{\mu} \Im \left[\langle \gamma, \beta | \hat{W}_n^p \hat{C}_{\mu} \hat{W}_n^{p\dagger} \hat{C} | \gamma, \beta \rangle \right]$$

$$\frac{\partial F_p}{\partial \beta_n} = -2 \sum_i \Im \left[\langle \gamma, \beta | \hat{W}_n^p \hat{\sigma}_x^{(i)} \hat{W}_n^{p\dagger} \hat{C} | \gamma, \beta \rangle \right]$$

- Since all terms in \hat{C} are products of Pauli-Z operators, the cost function is directly measurable in the computational basis.

Expressiveness:

- For deeper circuits ($p \rightarrow \infty$), QAOA can reach the exact solution, certifying the ansatz's expressiveness.
- For small p , it still yields high-quality approximate solutions with modest quantum resources.

Experimental Setup

Objective: Analyze the performance of a hybrid quantum–classical optimization framework using the Quantum Approximate Optimization Algorithm (QAOA) applied to the **MAX-CUT problem** on random 3-regular graphs.

Graph Instances: We consider instances of 3-regular graphs, where each node is connected to exactly three others.

Implementation:

- Simulations performed using **Qiskit**.
- Backend: **Qiskit AerSimulator** with **NVIDIA GPU** acceleration.
- QAOA circuit depth: $p = 4$.

Reference Simulation Framework:

- Original study used **qHiPSTER**—a high-performance quantum simulator developed by Intel and Harvard.
- Supports massively parallel simulation of quantum circuits for High Performance Computing (HPC).

Results for Single Instance

Observation: BFGS-AG achieves a higher fraction of satisfied clauses with fewer repetitions compared to Nelder–Mead, indicating better optimization efficiency for the single instance (number 111 in our simulations).

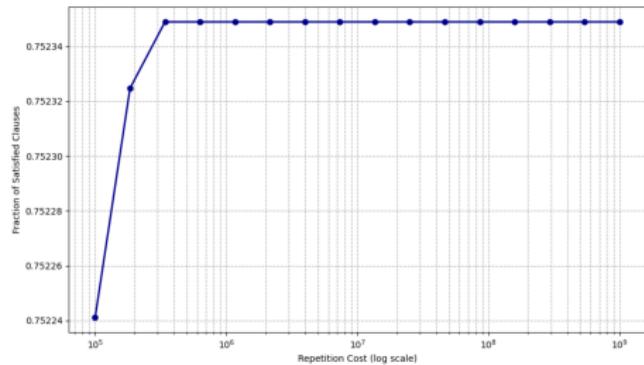


FIG. 8. Performance of BFGS-AG: Fraction of satisfied clauses vs. repetition cost (log scale).

Both methods converge, but BFGS-AG reaches optimal satisfaction faster and more consistently.

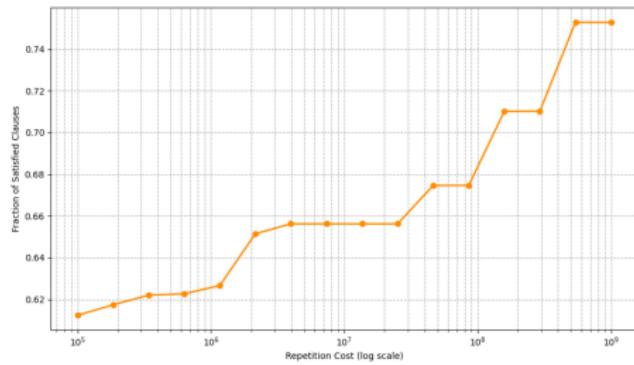


FIG. 9. Performance of Nelder–Mead: Fraction of satisfied clauses vs. repetition cost (log scale).

Results for Multiple Instances

Observation: Across all four instances (33, 63, 120, and 111), **AG** demonstrates superior performance—achieving higher clause satisfaction with fewer repetitions compared to other methods.

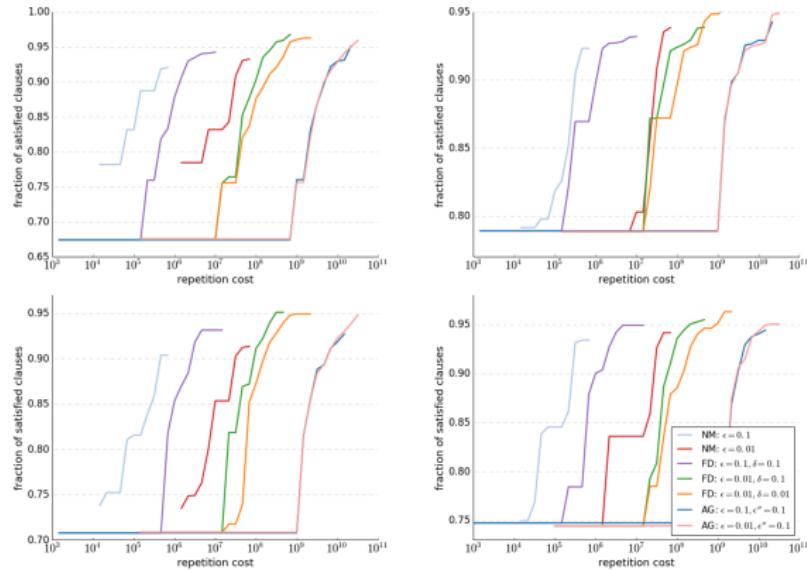


FIG. 10. Clause satisfaction vs. repetition cost (log scale) across four instances: 33, 63, 120, and 111. AG shows consistent advantage in convergence and efficiency.

Results for Depth Scaling Comparison

Observation: Across varying circuit depths, **AG** consistently achieves the highest clause satisfaction. While its advantage over **FD** is modest, it significantly outperforms **NM**, especially as depth increases.

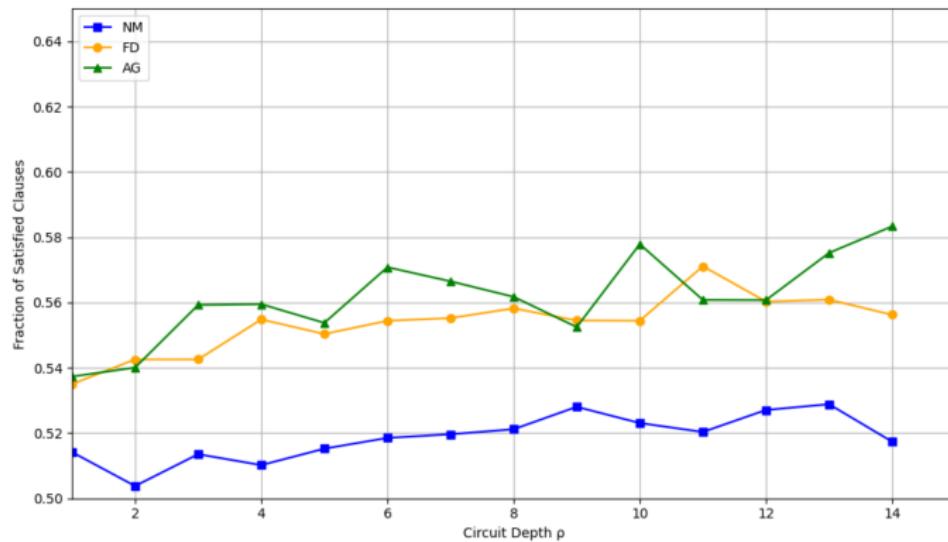


FIG. 11. Fraction of satisfied clauses vs. circuit depth ρ . AG consistently outperforms FD and NM across all depths, indicating better optimization scalability.

Critique and Open Questions

Critique:

- While the Analytical Gradient (AG) method provides improved accuracy, its higher sampling cost limits scalability for large qubit systems.
- Classical simulators (e.g., qHiPSTER, Qiskit Aer) simplify experimentation but do not fully capture real-device noise or decoherence effects.
- The study focuses on MAX-CUT over 3-regular graphs; broader benchmarking across other combinatorial problems would strengthen generalization.

Open Questions:

- How can adaptive hybrid optimizers balance precision and cost in noisy intermediate-scale quantum (NISQ) devices?
- Can machine learning or reinforcement learning guide optimal parameter initialization for QAOA to improve convergence?
- What is the impact of realistic hardware noise on the accuracy of analytical gradient evaluation?

References

Hybrid Quantum Algorithms and Optimization

- A. Peruzzo et al., Nature Communications 5, 4213 (2014).
- J. R. McClean et al., New J. Phys. 18, 023023 (2016).
- E. Farhi et al., arXiv:1411.4028 (2014).
- E. Farhi and A. Harrow, arXiv:1602.07674 (2016).

Optimization Techniques

- J. A. Nelder and R. Mead, Comput. J. 7, 308 (1965).
- J. Nocedal and S. Wright, Numerical Optimization, Springer (2000).
- S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge Univ. Press (2004).

Quantum Hardware and Experimental Platforms

- R. Barends et al., Nature 534, 222 (2016).
- T. Monz et al., Science 351, 1068 (2016).
- S. Debnath et al., Nature 536, 63 (2016).

Simulation and Benchmarking

- D. Wecker et al., Phys. Rev. A 94, 022309 (2016).
- M. Smelyanskiy et al., arXiv:1601.07195 (2016).
- N. P. D. Sawaya et al., arXiv:1602.01857v2 (2016).