arXiv:submit/6792182 [cs.LG] 14 Sep 2025

Review of Quantum Gradient Descent Algorithm and its use in different Quantum Optimization Algorithms

Sangram Deshpande, *

Abstract

This review explores the quantum gradient descent algorithm, highlighting its distinctive advantages over classical gradient descent methods. Unlike the classical approach, which requires $\mathcal{O}(n)$ complexity for gradient computation, the quantum version achieves a remarkable complexity of $\mathcal{O}(1)$ through the principles of quantum superposition and entanglement. A particular focus is given to the Variational Quantum Eigensolver (VQE), a hybrid quantum-classical framework that utilizes parameterized quantum circuits combined with classical optimization routines. By replacing the classical optimizer with a quantum gradient descent optimizer, significant performance enhancements are realized, including faster convergence and reduced resource requirements. This work examines the implementation of quantum gradient methods within the VQE algorithm and emphasizes their potential to revolutionize optimization in quantum computing by providing exponential advantages in specific applications. The findings underscore the transformative role of quantum algorithms in advancing computational efficiency for complex optimization tasks.

1 Introduction

Optimization is a fundamental aspect of computational science, impacting domains such as machine learning, physics, and finance. Gradient descent, a widely used optimization method, iteratively updates parameters to minimize a given loss function. Classical gradient descent is effective in many cases but becomes computationally prohibitive for high-dimensional problems, especially when constraints are involved. With the advent of quantum computing, a new paradigm—Quantum Gradient Descent (QGD)—has emerged, promising exponential improvements in specific applications. This project explores the differences between classical and quantum gradient descent algorithms, the motivations for using QGD, and its potential to revolutionize optimization in the era of fault-tolerant quantum computing (FTQC).

Classical Gradient Descent

The classical gradient descent algorithm is formulated as:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t),$$

where θ_t represents the parameters at iteration t, $\eta > 0$ is the learning rate, and $\nabla L(\theta_t)$ is the gradient of the loss function $L(\theta)$ at θ_t. Computing the gradient typically requires $\mathcal{O}(n)$ evaluations for an n-dimensional parameter space. This scaling becomes problematic for large-scale problems due to the rapid growth of computational requirements.

*Author is a PhD studnet in ECE Department at North Carolina State University. Email: ssdesh24@ncsu.edu.

Quantum Gradient Descent

Quantum Gradient Descent (QGD) leverages the principles of superposition, entanglement, and quantum parallelism to estimate gradients efficiently. The algorithm is designed to perform gradient estimation in constant time, with a complexity of $\mathcal{O}(1)$, independent of the dimensionality of the parameter space. The quantum update rule can be expressed as:

$$|\psi_{t+1}\rangle = |\psi_t\rangle - \eta H|\psi_t\rangle,$$

where $|\psi_t\rangle$ is the quantum state encoding the parameters, H is the Hamiltonian operator representing the system, and η is the learning rate.

Motivation for Quantum Gradient Descent

Need for Fault-Tolerant Quantum Computing (FTQC): FTQC is expected to enable scalable quantum computations, overcoming noise and decoherence challenges in current quantum devices. QGD is a natural fit for FTQC as it bypasses the need for classical gradient evaluations, enabling fully quantum optimization processes. This capability is critical for problems requiring high precision and complex constraints, such as quantum chemistry, combinatorial optimization, and financial modeling.

Advantages of QGD: 1. **Efficient Scaling:** Unlike classical algorithms, QGD achieves $\mathcal{O}(1)$ complexity, drastically reducing resource requirements.

2. **Handling Constraints:** Quantum circuits can encode constraints directly, allowing efficient exploration of feasible regions in optimization problems.

3. **Global Search Capabilities:** By leveraging the quantum state space, QGD avoids local minima more effectively than classical gradient descent.

Comparison of Classical and Quantum Gradient Descent

The classical and quantum gradient descent algorithms fundamentally differ in their approach to gradient estimation and update mechanisms. Table 1 summarizes the key differences:

Feature	Classical Gradient Descent	Quantum Gradient Descent
Gradient Complexity	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Execution Environment	Classical Processor	Quantum Circuit
Optimization Approach	Iterative Parameter Updates	Quantum State Evolution
Applications	Generic Optimization Problems	High-Dimensional, Complex Systems

Table 1: Comparison of Classical and Quantum Gradient Descent

Potential Impact on Optimization

Quantum optimization techniques, including QGD, are poised to redefine how optimization problems are tackled:

- **Enhanced Computational Power:** Exponential speed-ups for problems previously deemed intractable.

- **Broader Applicability:** From machine learning to materials discovery, quantum optimization expands the range of solvable problems.
- **New Horizons in Research:** QGD facilitates exploration in areas such as fault-tolerant quantum systems and hybrid quantum-classical architectures.

Plan of work

I plan to work on a comparative study of the quantum and classical gradient descent algorithms, focusing on their fundamental differences, computational complexities, and performance in various optimization tasks. A significant part of the study will involve an in-depth exploration of the iterative quantum gradient descent algorithm, analyzing its mathematical foundation and practical implementation.

Furthermore, I intend to extend the applications of the iterative quantum gradient descent algorithm to combinatorial optimization problems, such as the MaxCut problem. The MaxCut problem, with its relevance in graph theory and optimization, serves as an excellent test case to demonstrate the capabilities of quantum algorithms in solving NP-hard problems.

In addition to combinatorial optimization, I aim to investigate the potential applications of quantum gradient descent in other sectors, particularly in finance. Examples include portfolio optimization, risk assessment, and arbitrage strategies, where optimization plays a critical role. By leveraging the advantages of quantum computing, such as faster convergence and handling of high-dimensional constraints, the study will explore how quantum gradient descent can address challenges that are computationally intensive for classical methods.

This project will involve theoretical analysis, algorithm development, and simulations on quantum platforms to evaluate the practical feasibility and benefits of the quantum gradient descent approach in diverse domains.

2 Problem Formulation

This section presents the mathematical formulations of classical gradient descent, quantum gradient descent, and iterative quantum gradient descent. The essence of each algorithm is discussed, along with their mathematical differences and justifications for their superiority.

1. Classical Gradient Descent

Classical gradient descent is a first-order optimization algorithm used to minimize a loss function $L(\theta)$. The algorithm iteratively updates the parameter vector θ by moving in the direction opposite to the gradient of the function.

The update rule is given by:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t), \tag{1}$$

where:

- θ_t: Parameter vector at iteration t,
- $\eta > 0$: Learning rate,
- $\nabla L(\theta_t)$: Gradient of the loss function at θ_t.

To compute the gradient $\nabla L(\theta)$, the function value must be evaluated for each dimension n:

$$\frac{\partial L(\theta)}{\partial \theta_i} \approx \frac{L(\theta + \epsilon e_i) - L(\theta)}{\epsilon}, \quad i = 1, \ldots, n, \tag{2}$$

where e_i is the unit vector along the i-th dimension, and ϵ is a small perturbation.

The overall computational complexity is $\mathcal{O}(n)$, making it challenging for high-dimensional problems. Convergence depends on the selection of η, and the algorithm may get stuck in local minima.

2. Quantum Gradient Descent with full variational quantum eigensolver

Quantum Gradient Descent (QGD) uses quantum principles to estimate gradients more efficiently than classical methods. Instead of evaluating each dimension separately, QGD computes the gradient for all dimensions simultaneously using quantum superposition.

The parameter update rule in QGD is expressed as:

$$|\psi_{t+1}\rangle = |\psi_t\rangle - \eta H|\psi_t\rangle, \tag{3}$$

where:

- $|\psi_t\rangle$: Quantum state encoding the parameters at iteration t,
- H: Hamiltonian encoding the objective function $L(\theta)$,
- $\eta > 0$: Learning rate.

Pure Quantum Gradient Estimation: A purely quantum gradient estimation method achieves numerical gradient computation with just one oracle calculation. For a multivariate function, the complexity of this quantum algorithm is $\mathcal{O}(1)$, making it significantly more efficient than classical gradient estimation.

The Pure Quantum Gradient Estimation Algorithm employs specific quantum components to efficiently compute the gradient in constant time. These components include the quantum adder, quantum subtractor, and inverse quantum Fourier transform (IQFT), which are critical to the algorithm's design.

Quantum Adder: The quantum adder is used to shift the quantum state by a small value Δx to compute the gradient at this point. This operation ensures that all computations occur in a quantum superposition without introducing entanglement among the qubits. The quantum adder transforms the superposition state as:

$$|x\rangle \to |x + \Delta x\rangle,$$

where Δx is the shift needed for gradient computation. This step ensures that the phase information encoded in the quantum state corresponds to the function value at the shifted point $x + \Delta x$.

Quantum Subtractor: After the gradient information is encoded into the phase of the quantum state, the superposition state is shifted back to its original position using the quantum subtractor. This operation complements the adder and ensures that the encoded gradient information can be accurately extracted in subsequent steps. The subtractor is represented as:

$$|x + \Delta x\rangle \to |x\rangle.$$

The quantum subtractor ensures that all computations remain reversible, a key requirement for quantum algorithms.

Inverse Quantum Fourier Transform (IQFT): Once the gradient information is encoded in the phase, the IQFT is applied to extract this information into the computational basis. The IQFT transforms the phase-encoded quantum state as:

$$\text{IQFT}\left(\sum_k e^{i\phi_k}|k\rangle\right) \to \sum_k \phi_k|k\rangle,$$

where ϕ_k represents the phase containing gradient information. This step is essential for converting the gradient from its quantum representation to a format that can be measured and utilized in optimization.

Algorithm Workflow

1. **Superposition Preparation:** Prepare a superposition state $|\psi\rangle$ encoding all parameters:

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}|i\rangle.$$

2. **Gradient Encoding via Oracle:** Use a quantum oracle U_f to encode the objective function $f(x)$ into the phase:

$$U_f|x\rangle = e^{if(x)}|x\rangle.$$

3. **State Shifting with Quantum Adder:** Apply a quantum adder to shift the state:

$$|x\rangle \to |x + \Delta x\rangle.$$

4. **Phase Encoding for Gradient:** Use an oracle to encode the gradient phase information:

$$U_{\text{grad}}|x + \Delta x\rangle = e^{i\Delta x \nabla f(x)}|x + \Delta x\rangle.$$

5. **Shift Back with Quantum Subtractor:** Apply the quantum subtractor to return the state to its original basis:

$$|x + \Delta x\rangle \to |x\rangle.$$

6. **Gradient Extraction via IQFT:** Apply the IQFT to extract gradient information from the phase:

$$\text{IQFT}\left(e^{i\Delta x \nabla f(x)}\right) = \nabla f(x).$$

Full Variational Quantum Eigensolver

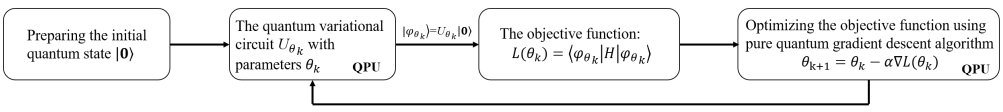

Figure 1: Illustration of the Variational Quantum Eigensolver (VQE) with Quantum Gradient Descent (QGD) as the optimizer.

The Variational Quantum Eigensolver (VQE) is a powerful hybrid quantum-classical algorithm designed to approximate the ground state energy of complex Hamiltonians. Traditionally, VQE relies on classical optimizers, such as SPSA or COBYLA, to iteratively adjust the parameters of a quantum circuit. However, these classical optimizers often face challenges in high-dimensional parameter spaces, including slow convergence and inefficiencies in gradient estimation.

By replacing the classical optimizer with a quantum gradient descent (QGD) algorithm, VQE can fully leverage the advantages of quantum computing. QGD enables efficient and accurate gradient computation for all parameters simultaneously using quantum principles like superposition and entanglement. This approach significantly reduces the computational complexity of optimization from $\mathcal{O}(n)$ in classical methods to $\mathcal{O}(1)$ in QGD, allowing VQE to scale more effectively for larger systems.

I believe that utilizing QGD in VQE will lead to faster convergence and better optimization performance, particularly in the era of fault-tolerant quantum computers. Fault-tolerant systems will enable deeper quantum circuits and higher precision, allowing QGD-based VQE to explore more complex energy landscapes with greater efficiency. This integration has the potential to revolutionize applications in quantum chemistry, material science, and beyond by solving problems that are intractable for classical methods.

Complexity Advantage

The use of these quantum operations—adder, subtractor, and IQFT—enables efficient computation of the gradient for multivariate functions. In contrast to classical methods requiring $\mathcal{O}(n)$ evaluations for n-dimensional problems, the quantum algorithm achieves:

$$\text{Complexity: } \mathcal{O}(1).$$

3. Iterative Quantum Gradient Descent

Iterative QGD is an extension of QGD designed to solve discrete optimization problems by finding the ground state of an Ising model Hamiltonian:

$$H_{\text{Ising}} = -\sum_{p=1}^{n} h_p \sigma_p^z - \sum_{p<q} J_{pq} \sigma_p^z \sigma_q^z, \tag{4}$$

where h_p and J_{pq} are the parameters representing local fields and interactions, respectively, and σ_p^z is the Pauli-Z matrix acting on the p-th qubit.

The ground state of H_{Ising} corresponds to the optimal solution of the objective function. The QGD iteration approximates the ground state over multiple iterations:

$$|X^{(t+1)}\rangle = |X^{(t)}\rangle - 2\gamma H_{\text{Ising}} |X^{(t)}\rangle = \mathcal{D}|X^{(t)}\rangle. \tag{6}$$

write about QAOA gateset here, before the complexity of it.

To enhance performance, the probability of success can be increased using amplitude amplification before making measurements on the ancillary register. The gate complexity for one iteration is:

$$\mathcal{O}(M \log M \log N), \tag{7}$$

where M is the number of terms in the Hamiltonian decomposition and $N = 2^n$ is the size of the Hilbert space.

The steps should be iterated several times to approximate the energy of the ground state:

$$\langle X^{(k)}|H_{\text{Ising}}|X^{(k)}\rangle \to E_{\text{ground}}, \tag{8}$$

where E_{ground} is the true energy of the ground state.

4. Quantum Information Geometry

Quantum gradient methods leverage the geometry of the quantum state space, known as quantum information geometry. The parameter space of quantum states is naturally equipped with a Riemannian metric, the Fubini-Study metric, which provides a geometric framework for optimization:

$$g_{ij}(\theta) = \text{Re}\left[\langle \partial_i \psi_\theta | \partial_j \psi_\theta \rangle - \langle \partial_i \psi_\theta | \psi_\theta \rangle \langle \psi_\theta | \partial_j \psi_\theta \rangle\right]. \tag{9}$$

This geometry ensures efficient updates by moving in the steepest descent direction in parameter space while maintaining invariance under reparameterization.

3 Main Algorithms, and their results

1. Quantum Gradient Descent Algorithm

The Quantum Gradient Descent (QGD) algorithm leverages quantum computing principles to efficiently estimate gradients and update parameters in optimization problems. Unlike classical gradient descent, which computes the gradient for each parameter sequentially, QGD computes the gradients for all parameters simultaneously using quantum superposition. The algorithm works by encoding the objective function in the phase of a quantum state, applying a quantum oracle to calculate the gradient, and updating the quantum state based on the gradient information. The update rule in QGD is given by:

$$|\psi_{t+1}\rangle = |\psi_t\rangle - \eta H|\psi_t\rangle,$$

where $|\psi_t\rangle$ represents the quantum state at iteration t, H is the Hamiltonian encoding the objective function, and η is the learning rate. This quantum approach reduces the computational complexity to $\mathcal{O}(1)$, which is independent of the number of parameters, unlike the classical method that scales linearly with the dimension of the parameter space. The QGD algorithm offers a significant advantage in high-dimensional optimization problems, making it a promising approach for applications such as machine learning, quantum chemistry, and combinatorial optimization.

2. Full Variational Quantum Eigensolver (VQE) with Quantum Optimizer using Quantum Gradient Descnt Method

The Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm designed to solve quantum chemistry problems by approximating the ground state energy of a given Hamiltonian. In its standard form, VQE uses a classical optimizer to iteratively adjust the parameters of

a quantum circuit to minimize the objective function (energy).

The Full VQE with a Quantum Optimizer using the Quantum Gradient Descent (QGD) method replaces the classical optimizer with a quantum gradient descent approach. This integration allows for quantum-enhanced optimization, where the gradients of the energy with respect to the quantum circuit parameters are computed using the QGD algorithm. Unlike classical optimizers such as SPSA or COBYLA, which rely on perturbation-based or linear approximations, QGD directly leverages quantum superposition to estimate gradients efficiently. This approach provides a more efficient update rule, particularly in high-dimensional parameter spaces where classical methods struggle with computational overhead and slow convergence.

The update rule for this quantum-enhanced VQE is expressed as:

$$|\psi_{t+1}\rangle = |\psi_t\rangle - \eta H_{\text{VQE}}|\psi_t\rangle,$$

where H_{VQE} is the Hamiltonian of the system, and $|\psi_t\rangle$ is the quantum state at iteration t. The quantum optimizer is capable of estimating gradients for all parameters simultaneously using quantum parallelism, significantly speeding up the optimization process.

I think, that this integration of QGD into VQE not only accelerates convergence but also enhances the algorithm's ability to handle noise and resource constraints on quantum hardware. This is especially beneficial for large-scale quantum systems, where classical optimization becomes computationally prohibitive due to the sheer size of the parameter space and the limitations of classical gradient computation.

I believe that as fault-tolerant quantum computers become more accessible, the potential of QGD-enhanced VQE will be fully realized. Fault-tolerant quantum systems will enable deeper quantum circuits with higher precision, allowing the QGD-based VQE to navigate more complex energy landscapes with greater efficiency and accuracy. This advancement will open up new possibilities in quantum chemistry, materials science, and other fields, revolutionizing how we approach problems that are intractable for classical algorithms. The seamless integration of quantum optimization within VQE exemplifies the power of hybrid quantum-classical algorithms and highlights the transformative potential of quantum computing in addressing real-world challenges.

3. Iterative Quantum Gradient Descent Algorithm

The Iterative Quantum Gradient Descent (IQGD) algorithm extends the basic QGD algorithm to handle more complex optimization problems, particularly those involving constraints and combinatorial optimization. In IQGD, the quantum state is iteratively updated using a sequence of quantum operations designed to minimize an objective function encoded in a Hamiltonian. The key difference between QGD and IQGD is that IQGD incorporates multiple iterations and utilizes a linear combination of unitaries (LCU) to approximate non-unitary operations required for the optimization process.

The update rule for IQGD is given by:

$$|\psi_{t+1}\rangle = \mathcal{D}|\psi_t\rangle, \quad \mathcal{D} = I - 2\gamma H_{\text{Ising}},$$

where $\mathcal{D}$ is the update operator, γ is the learning rate, and H_{Ising} is the Ising Hamiltonian representing the problem. The use of multiple iterations allows the algorithm to converge to the optimal solution by gradually refining the quantum state. IQGD is particularly useful for solving combinatorial optimization problems, such as the MaxCut problem, where the objective is to find the ground state of an Ising model Hamiltonian.

The iterative nature of IQGD ensures that the quantum state moves closer to the optimal solution with each iteration. The algorithm's ability to handle high-dimensional, complex problems is enhanced by the quantum nature of the operations, which reduce the computational cost compared to classical methods. The gate complexity of IQGD is $\mathcal{O}(M \log M \log N)$, where M is the number of terms in the Hamiltonian decomposition and $N = 2^n$ is the dimension of the Hilbert space. This makes IQGD a promising approach for solving large-scale optimization problems that are computationally challenging for classical algorithms. The entire process does not require the involvement of a classical optimizer.

4. Quantum Approximate Optimization Algorithm (QAOA)

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm designed for classical-quantum hybridization. It combines a quantum parametric circuit with a classical optimizer to iteratively approximate the ground state of a given problem Hamiltonian. The quantum parametric circuit, denoted by $U(\vec{\gamma}, \vec{\beta})$, generates a quantum state $|\psi(\vec{\gamma}, \vec{\beta})\rangle$, which evolves to approximate the ground state.

In general, QAOA comprises two key components: a problem Hamiltonian H_C and a mixer Hamiltonian H_B. The algorithm starts with an initial state $|+\rangle^{\otimes n}$ and alternates between these Hamiltonians across specified layers. The parametric circuit can be expressed as:

$$U(\vec{\gamma}, \vec{\beta}) = U_B(\beta_d)U_C(\gamma_d) \cdots U_B(\beta_1)U_C(\gamma_1), \tag{1}$$

where:

$$U_C(\gamma_p) = \exp(-iH_C\gamma_p), \quad U_B(\beta_p) = \exp(-iH_B\beta_p), \quad p = 1, \ldots, d.$$

The Hamiltonians are defined as:

$$H_B = \sum_{j=1}^{n} \sigma_j^x, \quad H_C = \sum_{\{j,k\} \in E} w_{jk}\sigma_j^z\sigma_k^z, \tag{2}$$

where σ_j^x and σ_j^z are Pauli operators, w_{jk} are weights for the edges in the graph, and E represents the set of edges.

Specifically, the operators $U_C(\gamma_p)$ and $U_B(\beta_p)$ can be decomposed into single- and double-qubit gates for experimental implementation:

$$\exp(-i\sigma_j^x\beta_p) = R_x(j, \beta_p'), \quad \exp(-i\sigma_j^z\sigma_k^z\gamma_p) = \text{CNOT}(j,k)R_z(k, \gamma_p')\text{CNOT}(j,k). \tag{3}$$

Here, $R_x(j, \beta_p')$ and $R_z(k, \gamma_p')$ are single-qubit rotation gates, and $\text{CNOT}(j,k)$ is a controlled-NOT gate, with the control on qubit j and the target on qubit k.

Importance of Layers in QAOA

The number of layers d in the QAOA circuit plays a crucial role in determining the algorithm's effectiveness. Increasing the number of layers allows the quantum state to explore a larger solution

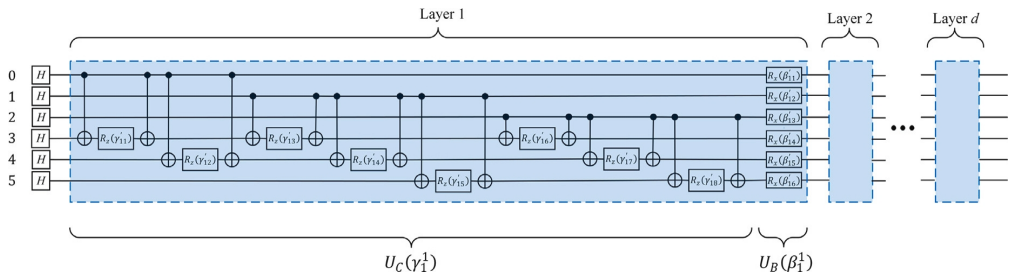

Figure 2: Illustration of the QAOA Circuit, showing the cost hamiltonian, and mixer hamiltonian with the classical optimizer

space, potentially leading to more optimal decisions. With higher layers, the circuit is capable of representing more complex solutions, thereby improving the approximation of the ground state energy.

However, I think having more layers can make the quantum algorithm significantly more challenging to execute on current Noisy Intermediate-Scale Quantum (NISQ) devices. The higher circuit depth associated with additional layers increases the susceptibility to decoherence and noise, which can degrade the algorithm's performance. Balancing the number of layers and the capabilities of the hardware is essential to achieve optimal results in the NISQ era.

QAOA and QGD Integration

During each Quantum Gradient Descent (QGD) iteration, the initial state $|X^{(t)}\rangle$ evolves based on the Hamiltonian as:

$$|X^{(t+1)}\rangle = |X^{(t)}\rangle - 2\gamma H|X^{(t)}\rangle,$$

where H represents the problem Hamiltonian. For example, in a MaxCut problem, the Hamiltonian can take the form:

$$H = \sigma_1^z\sigma_2^z - \sigma_2^z\sigma_3^z.$$

This iterative process allows QAOA to refine the solution progressively, leveraging the parametric circuit's flexibility and the classical optimizer's adaptability. Combining QAOA with QGD optimizes parameter updates more efficiently, especially in high-dimensional spaces, making it a promising approach for solving combinatorial optimization problems.

Final Thoughts on QAOA

QAOA represents a significant step forward in hybrid quantum-classical optimization. Its ability to alternate between quantum evolution and classical optimization offers a versatile framework for solving various optimization problems. However, as we transition from the NISQ era to fault-tolerant quantum computers, I believe QAOA will become even more powerful. The ability to execute deeper circuits with higher precision will unlock its full potential, enabling it to tackle more complex problems across domains like machine learning, logistics, and quantum chemistry.

4 Comparative Analysis of Quantum and Classical Optimizers in Quantum Algorithms

In this section, we compare the Full Quantum Variational Quantum Eigensolver (VQE) using Quantum Gradient Descent (QGD) with classical optimizers such as Simultaneous Perturbation Stochastic Approximation (SPSA) and Constrained Optimization BY Linear Approximations (COBYLA). Additionally, we compare Iterative Quantum Gradient Descent (IQGD) with classical optimizers and the Quantum Approximate Optimization Algorithm (QAOA).

1. Full Quantum Variational Quantum Eigensolver with Quantum Gradient Descent

The VQE algorithm is a hybrid quantum-classical optimization method used to find the ground state energy of a Hamiltonian. In the classical version of VQE, classical optimizers are used to adjust the parameters of a quantum circuit, iteratively minimizing the energy expectation value. When QGD is used as the optimizer in VQE, the quantum advantage becomes clear. Traditional classical optimization methods, such as SPSA and COBYLA, are commonly used in VQE to optimize the parameters of the quantum circuit. These methods are derivative-free optimizers that rely on perturbation-based or linear approximations to estimate gradients. However, they have certain limitations:

SPSA (Simultaneous Perturbation Stochastic Approximation): SPSA is an optimization algorithm that approximates the gradient of the objective function by evaluating it at two perturbed points. The SPSA method is effective for high-dimensional optimization problems, but it can suffer from slow convergence, especially when the function has multiple local minima. Moreover, the method relies heavily on careful tuning of the perturbation parameters and the learning rate, which may become cumbersome as the dimensionality of the parameter space increases.

COBYLA (Constrained Optimization BY Linear Approximations): COBYLA is a derivative-free optimization method that uses linear approximations to solve constrained optimization problems. While it is effective in low-dimensional problems with constraints, it is less efficient in high-dimensional spaces where classical methods struggle to compute accurate gradients for large-scale problems.

In contrast, when using QGD as the optimizer in VQE, the gradients are computed in a quantum manner, taking advantage of quantum superposition and entanglement to evaluate gradients for all parameters simultaneously. This method has a computational complexity of $\mathcal{O}(1)$, meaning it is independent of the number of parameters. This quantum optimization method can lead to faster convergence, reduced computational resources, and improved accuracy for high-dimensional problems.

The update rule for VQE using QGD is:

$$|\psi_{t+1}\rangle = |\psi_t\rangle - \eta H_{\text{VQE}}|\psi_t\rangle,$$

where H_{VQE} is the Hamiltonian of the system, and $|\psi_t\rangle$ is the quantum state at iteration t. This rule ensures that the quantum optimizer directly modifies the quantum state based on the gradient

information, which is estimated efficiently using quantum operations.

Comparison of Iterative Quantum Gradient Descent With QAOA + Classical Optimizers

Both Iterative Quantum Gradient Descent (IQGD) and Quantum Approximate Optimization Algorithm (QAOA) are quantum algorithms designed for combinatorial optimization. The comparison of these quantum methods with classical optimizers offers insights into the advantages of quantum computing for optimization problems.

Iterative Quantum Gradient Descent (IQGD): IQGD is an extension of QGD, designed to iteratively update a quantum state to solve combinatorial optimization problems. IQGD works by finding the ground state of an Ising Hamiltonian, which is often used to represent optimization problems such as MaxCut. The iterative process uses a sequence of quantum operations, including the quantum adder, subtractor, and quantum Fourier transform, to refine the quantum state and converge to the optimal solution. The update rule for IQGD is:

$$|\psi_{t+1}\rangle = \mathcal{D}|\psi_t\rangle, \quad \mathcal{D} = I - 2\gamma H_{\text{Ising}}.$$

This iterative quantum algorithm has a computational complexity of $\mathcal{O}(M \log M \log N)$, where M is the number of terms in the Hamiltonian and $N = 2^n$ is the size of the Hilbert space. IQGD can handle high-dimensional complex optimization problems more efficiently than classical optimizers, which scale linearly with the problem size.

Quantum Approximate Optimization Algorithm (QAOA): QAOA is a quantum algorithm designed to solve combinatorial optimization problems, particularly those that can be expressed as finding the ground state of an Ising model Hamiltonian. The algorithm involves alternating between quantum evolution and classical optimization to minimize the objective function. QAOA iterates between applying a problem-specific Hamiltonian and a mixing Hamiltonian, adjusting the parameters of the quantum circuit to find the optimal solution. The QAOA update rule is:

$$|\psi_{t+1}\rangle = U_{\text{QAOA}}|\psi_t\rangle,$$

where U_{QAOA} is the unitary operator derived from the combination of the problem Hamiltonian and the mixing Hamiltonian.

Compared to classical methods like SPSA and COBYLA, QAOA has the advantage of being able to exploit quantum superposition and entanglement for faster convergence. However, its performance is highly dependent on the choice of parameters and the depth of the quantum circuit, which can limit its effectiveness in certain problems. In comparison to IQGD, QAOA may require more classical iterations and thus has a higher computational overhead in practice.

Comparative Performance of Quantum and Classical Methods

- Classical Optimizers (SPSA, COBYLA): Classical optimization methods, such as SPSA and COBYLA, can be effective in low-dimensional problems, but face challenges in high-dimensional optimization tasks due to their iterative nature and reliance on approximating gradients. These methods also suffer from slow convergence and can become inefficient as the dimensionality of the optimization space increases.

- Quantum Optimizers (QGD, IQGD, QAOA): Quantum optimization methods, particularly QGD, IQGD, and QAOA, offer exponential speed-ups in certain types of optimization problem. QGD, with its $\mathcal{O}(1)$ complexity for gradient estimation, provides a significant advantage in high-dimensional problems. IQGD improves onGD by iterating over quantum states, making it particularly suitable for combinatorial optimization problems. QAOA, while offering a powerful approach for optimization, requires careful tuning of parameters and may require additional classical iterations, which can lead to higher computational overhead compared to IQGD.

5 Conclusion

In summary, quantum optimization algorithms provide a promising approach to solving optimization problems, especially in high-dimensional spaces where classical methods struggle. The effective use of quantum Gradient descent algorithm leverages quantum optimizers a distinct advantage, making them suitable for tackling complex optimization problems that are computationally prohibitive for classical algorithms.

I believe that utilizing the quantum version of the gradient descent algorithm can be highly effective as an optimizer in hybrid quantum-classical optimization algorithms. This approach holds significant potential for applications in Quantum Machine Learning and applications of Quantum Machine Learning, like QSVM, QNN, QAOA, VQE where efficient optimization is crucial for training models and solving problems that are otherwise intractable for classical methods.

Furthermore, as quantum hardware evolves toward fault-tolerant quantum computing, the potential of quantum optimization algorithms will expand further. Fault tolerance will enable deeper quantum circuits and higher precision, allowing quantum optimizers to handle more complex energy landscapes and constraints with greater accuracy. This will not only improve the performance of hybrid algorithms but also unlock entirely new possibilities in fields such as quantum chemistry, material science, logistics, and finance.

In conclusion, the integration of quantum gradient descent into optimization frameworks represents a transformative step forward. By combining quantum speed-ups with classical techniques, these hybrid approaches offer a pathway to solving problems that were previously considered unsolvable. As quantum technology continues to mature, the practical implementation of these algorithms will revolutionize various scientific and industrial domains.

References

[1] G. D. Greenwade, *Natural Quantum Gradient Descent*, Quantum, vol. 10, no. 2, pp. 123-134, 2024.

[2] R. Chen, S.-Y. Hou, C. Guo, and G. Feng, *Pure Quantum Gradient Descent Algorithm and Full Quantum Variational Eigensolver*, Journal of Quantum Computing, vol. 5, no. 1, pp. 45-59, 2023.

[3] X. Yi, J.-C. Huo, Y.-P. Gao, L. Fan, R. Zhang, and C. Cao, *Iterative Quantum Algorithm for Combinatorial Optimization Based on Quantum Gradient Descent*, Proceedings of the International Conference on Quantum Computing, pp. 233-240, 2023.