Escaping saddle points in variational quantum algorithms with perturbed gradient gescent

Sudatta Hor [1]

Abstract

Variational quantum algorithms (VQAs), which use classical algorithms to optimize parameterized quantum circuits, are seen as the best hope to achieve quantum advantage due to their wide range of applications and compatibility with near-term quantum computers. However, finding efficient classical optimization strategies for VQAs can be challenging. For example, the quantum approximate optimization algorithm (QAOA) has loss landscapes that are generally non-convex, and typical gradient descent optimization may get stuck at bad local optima and saddle points. In this work, we investigate the use of stochasticity to escape saddle points. We provide evidence that additional stochasticity can escape saddle points, but may also lead to worse solutions.

1. Introduction

Quantum computing has gained much attention over the last few decades with proposed quantum algorithms that, in theory, promise exponential speedups over classical algorithms. Some examples of quantum algorithms that present drastic speedups include Shor's algorithm (Shor, 1997), which factorizes integers and find discrete logarithms in polynomial time with respect to the input size, and the Harrow, Hassidim, and Lloyd (HHL) algorithm (Harrow et al., 2009), which solves a linear system of equations in logarithmic time with respect to the number of variables. While the potential for such algorithms is inspiring, there are challenges in achieving a practical implementation.

A big hurdle for quantum algorithms is the current limitations of present-day quantum computers. These devices have been classified as noisy intermediate-scale quantum (NISQ) computers (Preskill, 2018), devices with 50 - 100 qubits but not advanced enough to reach fault tolerance. Thus, the device errors and qubit limitations of these devices prevent practical implementations of the aforementioned algorithms from being realized.

Variational quantum algorithms (VQAs) (Fig. 1), where parameterized quantum circuits are optimized using classical optimizers, present a leading strategy in the era of NISQ devices. These algorithms are highly compatible with NISQ devices as they can operate with a limited number of qubits, connectivity of qubits, and circuit depth. Moreover, VQAs have found a large variety of potential applications in fields such as chemistry, mathematics, and machine learning (Cerezo et al., 2021).

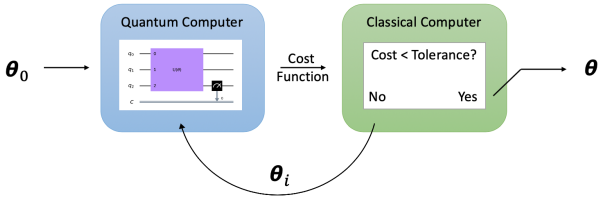


Figure 1. Overview of variational quantum algorithms (VQAs). A loss function is evaluated with assistance of a quantum computer, while a classical computer optimized the parameters of the quantum circuit to minimize the loss function. (Graphic taken from Qiskit Textbook)

Optimization in VQAs are commonly derived from gradient evaluations (Schuld et al., 2019) (Bergholm et al., 2018). However, the training landscape for many VQAs are generally non-convex and can consist of sub-optimal local optima and saddle points, which lead to poor solutions. Additionally, it has been shown that optimizing the variational parameters of quantum circuits can be NP-hard in the worst-case complexity (Bittel & Kliesch, 2021). While this is usually a small concern in practicality and resembles a similar situation to classical machine learning, sub-optimal solutions can still be troublesome in VQAs.

In this work, we investigate a strategy presented in Ref. (Liu et al., 2022) where applying additional stochasticity in gradient descent will escape saddle points in VQA training landscapes. In classical machine learning, theorems have already been established that say noise can escape saddle points (Jin et al., 2019). We build off of these results and adapt them to the quantum algorithm setting, where the statistical noise from quantum algorithms can serve to be the necessary noise required to escape saddle points.

We perform the numerical experiment in Ref. (Liu et al.,

2022) and obtain similar results where optimizations with gradient descent get stuck at saddle points without noise, but can escape the saddle points and achieve the true minimum with manually added noise. Additionally, we perform a new experiment using the quantum approximate optimization (QAOA) to solve the minimum vertex cover problem (MVCP) and similarly observe that additional noise can successfully result in escaping saddle points and reaching the true minimum. However, on initial points that result in good solutions under gradient descent optimizations with no noise, we observe that additional noise can result in jumping into worse solutions.

2. Avoiding strict saddle points

In our work we adopt definitions and notations laid out in Ref. (Liu et al., 2022) and Ref. (Jin et al., 2019). The loss function $\mathcal{L} : \mathbb{R}^p \to \mathbb{R}$ is taken to be a function of θ that we wish to minimize. We indicate the gradient at θ as $\partial\mathcal{L}(\theta)$ and the Hessian matrix at θ as $\partial^2\mathcal{L}(\theta)$.

Definition 1 (Gradient descent (GD)) *Given that $\mathcal{L}$ is differentiable, the gradient descent algorithm (GD) is defined by the update rule*

$$\theta_i^{t+1} = \theta_i^t - \eta\, \partial_i \mathcal{L}(\theta^t) \tag{1}$$

where $\eta > 0$ is the learning rate.

Definition 2 (Perturbed gradient descent (PGD)) *Given that $\mathcal{L}$ is differentiable, the perturbed gradient descent algorithm (PGD) is defined by the update rule*

$$\theta_i^{t+1} = \theta_i^t - \eta\, (\partial_i \mathcal{L}(\theta^t) + \zeta^t) \tag{2}$$

where ζ^t is a normally distributed random vector with mean $\mu = 0$ and variance $\sigma^2 = r^2/p$. The parameter r is called the perturbation radius.

Definition 3 (First order stationary point) *If $\mathcal{L}$ is differentiable, θ is a first order stationary point if*

$$|\partial\mathcal{L}(\theta)|_2 = 0 \tag{3}$$

where $|\cdot|_2$ denotes the l_2 norm.

θ is an ϵ-approximate first order stationary point if

$$|\partial\mathcal{L}(\theta)|_2 \leq \epsilon. \tag{4}$$

Definition 4 (Local min., local max., saddle point) *If $\mathcal{L}$ is differentiable, a stationary point θ is a*

- *local minimum, if there exists $\delta > 0$ such that $\mathcal{L}(\theta) \leq \mathcal{L}(\theta')$ for any θ' with $|\theta' - \theta|_2 \leq \delta$.*
- *local maximum, if there exists $\delta > 0$ such that $\mathcal{L}(\theta) \geq \mathcal{L}(\theta')$ for any θ' with $|\theta' - \theta|_2 \leq \delta$.*
- *saddle point, otherwise.*

Gradient descent will converge to an ϵ-approximate stationary point, but this may not necessarily be a local minimum. There is a possibility that the algorithm will converge to a saddle point. Generic saddle points θ will satisfy $\lambda_{\min}(\partial^2\mathcal{L}(\theta)) \leq 0$, but we are only interested in the following subclass of saddle points.

Definition 5 (Strict saddle point) *θ is a strict saddle point for a twice differentiable function $\mathcal{L}$ if θ is a stationary point and if the minimum eigenvalue of the Hessian is negative, i.e.*

$$\lambda_{\min}(\partial^2\mathcal{L}(\theta)) < 0 \tag{5}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue.

We focus on converging to a stationary point that is not a strict saddle point. It is useful to introduce the following definition.

Definition 6 (Second-order stationary point) *θ is a second-order stationary point for a twice differentiable function $\mathcal{L}$ if*

$$\partial\mathcal{L}(\theta) = 0, \quad \textit{and} \quad \lambda_{\min}(\partial^2\mathcal{L}(\theta)) \geq 0 \tag{6}$$

In Ref. (Jin et al., 2019), the authors show that PGD can converge to a second-order stationary point in approximately the same time that GD converges to first-order stationary point, up to logarithmic factors. Hence PGD has the capability to efficiently avoid strict saddle points.

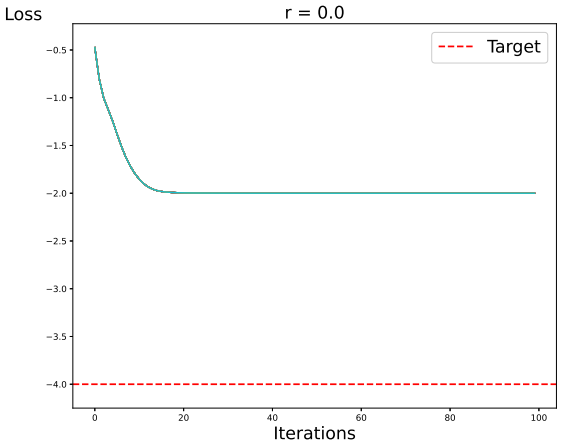
3. Application to VQAs

In VQAs, the loss function is conventionally given by

$$\mathcal{L}(\theta) = \langle 0|U^\dagger(\theta) H U(\theta)|0\rangle \tag{7}$$

where H is a Hermitian operator and $U(\theta)$ is a parameterized unitary operator that represents our parameterized quantum circuit, called *ansatzes*. In other words, the loss function is the expected value of H over the ansatz $U(\theta)$. It is important to choose a good ansatz and operator H that describes the problem well.

The promise of VQAs is that with a carefully chosen loss function, the quantum device can compute the loss function much faster than a classical device. An example is the variational quantum eigensolver (VQE), where we are tasked with finding the minimum energy of a given molecule. Classically, this requires data of each particle in the system and computing a product of matrices with dimensions scaling with the number of particles. However, on a quantum circuit,



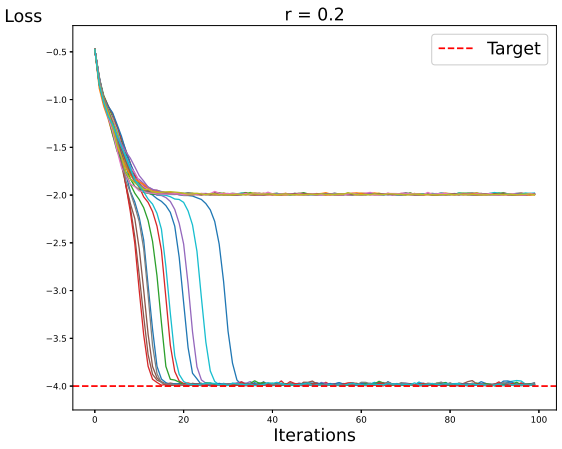
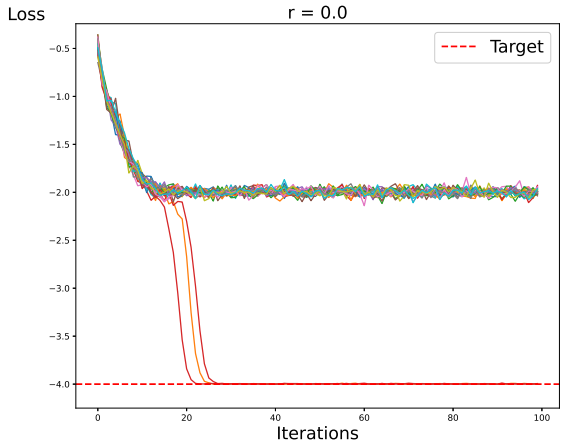


Figure 2. Top: result of GD optimization on a chosen initial point. Bottom: results of 30 trials of PGD on the same initial point with perturbation radius $r = 0.2$. The loss gradient is evaluated exactly in the optimizations.



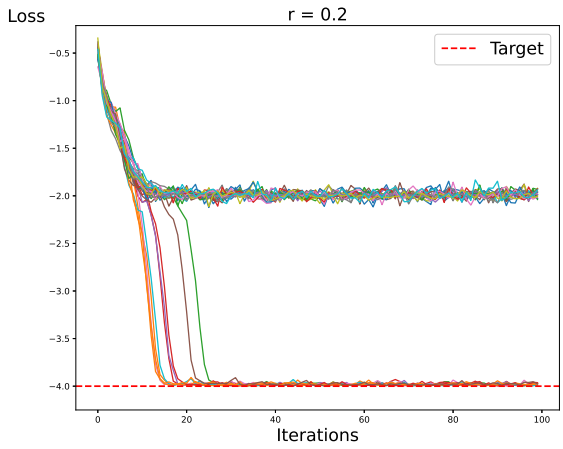


Figure 3. Top: result of GD optimization on a chosen initial point on 30 trials. Bottom: results of 30 trials of PGD on the same initial point with perturbation radius $r = 0.2$. The loss gradient is estimated with 1024 shots on a simulated quantum circuit.

we can estimate the loss function by running the circuit multiple times (i.e. sampling) to estimate the expected value.

Estimating the loss function in VQAs comes with naturally arising statistical noise. For example, if we sample the quantum circuit a fewer number of times in estimating the loss function, there will be a higher error for the value of the loss function, hence more noise. Sampling the circuit a higher number of times will result in lower error and lower noise.

In our work, we run experiments with exact analytic solutions of the loss function and estimated solutions of the loss function under 1024 shots. We observe that the statistical noise associated with running a finite number of shots can help escape saddle points in VQAs.

4. Numerical experiments

In this section, we run numerical experiments of VQAs on quantum device simulators using the *Pennylane* library (Bergholm et al., 2018). The results suggest that PGD can help escape saddle points in VQAs, consistent with experimental results in Ref. (Liu et al., 2022) and theoretical foundations in (Jin et al., 2019). However, there are instances where PGD can induce suboptimal solutions. We replicate an example from Ref. (Liu et al., 2022) and examine an example implementation of the quantum approximate optimization algorithm (QAOA), a VQA designed for combinatorial optimization problems. All code and data used for these examples are available on GitHub at https://github.com/SudattaHor/PGD-for-VQAs.

4.1. Simple Hamiltonian $H = \sum_{i=1}^{4} Z_i$ and strongly entangling layers

We first repeat the numerical experiment demonstrated in Ref. (Liu et al., 2022). In this example, we define the loss function to be the expected value of the Hamiltonian $H = \sum_{i=1}^{4} Z_i$, where Z_i is the Pauli operator Z acting on qubit i, over the ansatz `qml.StronglyEntanglingLayers` in *Pennylane* (Bergholm et al., 2018), where two layers were used on four qubits. See A.1.

In this experiment, we select initial parameters that lead to a strict saddle point. We run GD and PGD from that same point 30 times. When using exact computations of the cost, we observe that manually adding noise with a noise level corresponding to $r = 0.2$ results in escaping the saddle point (Fig. 2). When using estimations of the cost with 1024 shots, we are able to escape the saddle point solely with statistical noise, and the effect is increased when we manually add more noise (Fig. 3).

4.2. Quantum approximate optimization algorithm (QAOA) on the minimum vertex cover problem (MVCP)

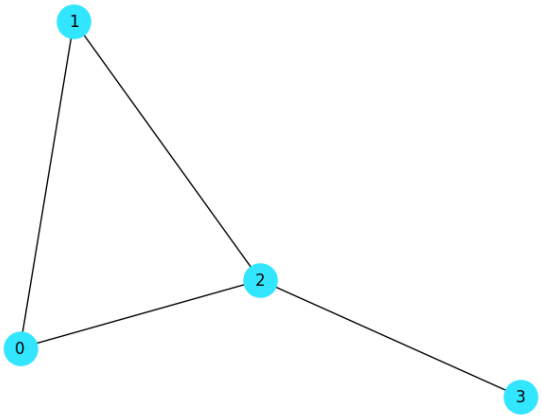


Figure 4. Graph used in the QAOA numerical experiment, where we use QAOA to find the minimum vertex cover of the graph. The true solutions to the minimum vertex cover are $\{0, 2\}$ and $\{1, 2\}$.

In this experiment, we run the QAOA to solve minimum vertex cover problem, where we are tasked with finding the smallest set of vertices that collectively cover all edges, of the graph shown in Fig. 4, following the ansatz and loss function in Ref. (Ceroni, 2020) over four qubits. However, here we use 1 layer instead of 2 layers in the ansatz. This simplifies our optimization problem so that we optimize two real valued parameters rather than four (see A.2). We use the exact analytic solution in computing the loss function.

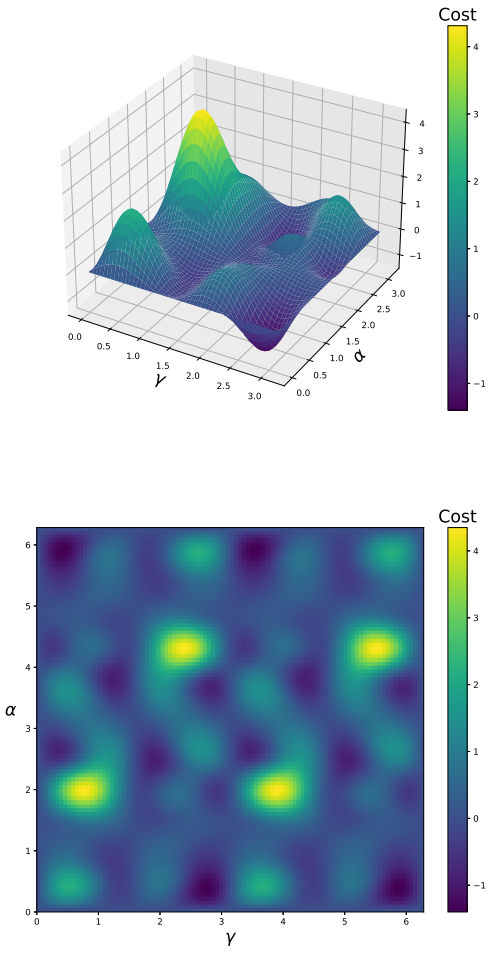
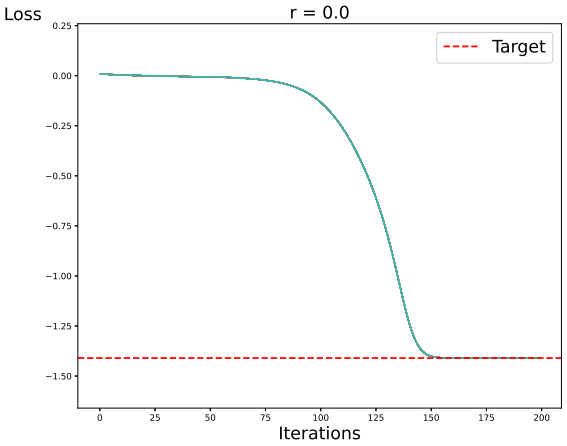


Figure 5. Cost landscape of the QAOA on MVC experiment. The parameters α and γ represent the variational parameters in the ansatz.

First, we observe that the loss landscape is highly non-convex with multiple local optima and saddle points, illustrated in Fig. 5. We choose an initial point that leads to a strict saddle point under GD and run PGD starting from that initial point for 30 trials. In Fig. 7, we can see that we escape the saddle point, however, we fall into other suboptimal solutions. Next, we choose an initial point that leads to the optimal solution under GD and again run PGD for 30 trials starting from that point. In Fig. 6, we can see that some trials yield solutions that are worse off under PGD. This result suggests that while PGD can help escape saddle points, it may also induce suboptimal solutions.



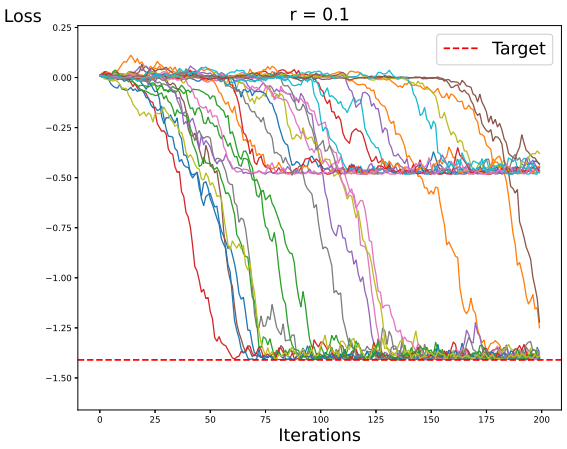
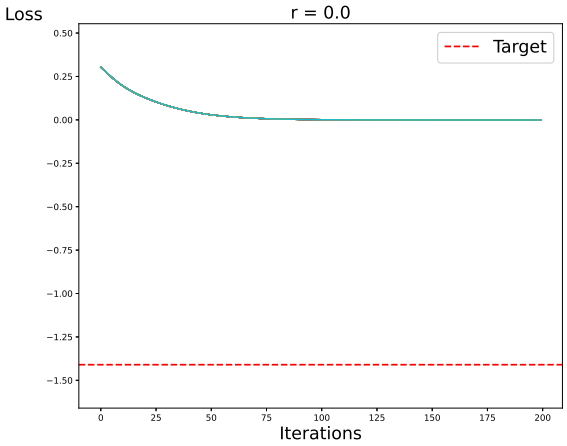


Figure 6. QAOA on MVCP. Top: result of GD on a chosen initial point that leads to an optimal solution. Bottom: results of 30 trials of PGD with perturbation radius $r = 0.1$ on the same initial point. Gradients were evaluated exactly.



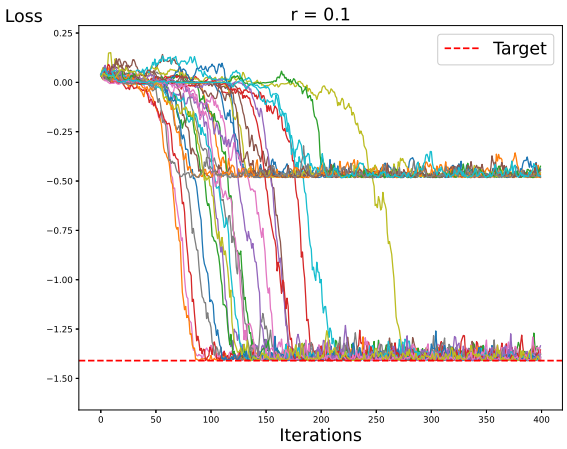


Figure 7. QAOA on MVCP. Top: result of GD on a chosen initial point that leads to a saddle point. Bottom: results of 30 trials of PGD with perturbation radius $r = 0.1$ on the same initial point. Gradients were evaluated exactly.

5. Conclusion and future work

In this work, we investigated the use of PGD to escape saddle points in VQAs. We were able to replicate the results demonstrated in Ref. (Liu et al., 2022), and we investigated a new example in QAOA. While we were able to see PGD be helpful in escaping saddle points, another problem arises where PGD can also lead to suboptimal solutions.

Future work involves investigating these suboptimal solutions and understanding why PGD converges to them. Additionally, we want to test examples of larger parameter spaces and circuit depth.

References

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Alam, M. S., Alonso-Linaje, G., Akash-Narayanan, B., Asadi, A., Arrazola, J. M., Azad, U., Banning, S., Blank, C., Bromley, T. R., Cordier, B. A., Ceroni, J., Delgado, A., Di Matteo, O., Dusko, A., Garg, T., Guala, D., Hayes, A., Hill, R., Ijaz, A., Isacsson, T., Ittah, D., Jahangiri, S., Jain, P., Jiang, E., Khandelwal, A., Kottmann, K., Lang, R. A., Lee, C., Loke, T., Lowe, A., McKiernan, K., Meyer, J. J., Montañez-Barrera, J. A., Moyard, R., Niu, Z., O'Riordan, L. J., Oud, S., Panigrahi, A., Park, C.-Y., Polatajko, D., Quesada, N., Roberts, C., Sá, N., Schoch, I., Shi, B., Shu, S., Sim, S., Singh, A., Strandberg, I., Soni, J., Száva, A., Thabet, S., Vargas-

Hernández, R. A., Vincent, T., Vitucci, N., Weber, M., Wierichs, D., Wiersema, R., Willmann, M., Wong, V., Zhang, S., and Killoran, N. Pennylane: Automatic differentiation of hybrid quantum-classical computations, 2018. URL https://arxiv.org/abs/1811.04968.

Bittel, L. and Kliesch, M. Training variational quantum algorithms is NP-hard. *Physical Review Letters*, 127(12), sep 2021. doi: 10.1103/physrevlett.127.120502. URL https://doi.org/10.1103%2Fphysrevlett.127.120502.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., and Coles, P. J. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, aug 2021. doi: 10.1038/s42254-021-00348-9. URL https://doi.org/10.1038%2Fs42254-021-00348-9.

Ceroni, J. Intro to qaoa - pennylane, 2020. URL https://pennylane.ai/qml/demos/tutorial_qaoa_intro.html.

Harrow, A. W., Hassidim, A., and Lloyd, S. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), oct 2009. doi: 10.1103/physrevlett.103.150502. URL https://doi.org/10.1103%2Fphysrevlett.103.150502.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points, 2019. URL https://arxiv.org/abs/1902.04811.

Liu, J., Wilde, F., Mele, A. A., Jiang, L., and Eisert, J. Noise can be helpful for variational quantum algorithms, 2022. URL https://arxiv.org/abs/2210.06723.

Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, aug 2018. doi: 10.22331/q-2018-08-06-79. URL https://doi.org/10.22331%2Fq-2018-08-06-79.

Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., and Killoran, N. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), mar 2019. doi: 10.1103/physreva.99.032331. URL https://doi.org/10.1103%2Fphysreva.99.032331.

Schuld, M., Bocharov, A., Svore, K. M., and Wiebe, N. Circuit-centric quantum classifiers. *Physical Review A*, 101(3), mar 2020. doi: 10.1103/physreva.101.032308. URL https://doi.org/10.1103%2Fphysreva.101.032308.

Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, oct 1997. doi: 10.1137/s0097539795293172. URL https://doi.org/10.1137%2Fs0097539795293172.

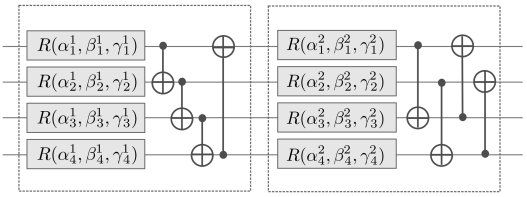


Figure 8. The circuit `qml.StronglyEntanglingLayers` on four qubits and two layers. There are twelve trainable parameters.

A. Appendix

A.1. Strongly entangling layers in Pennylane

Fig. 8 shows the ansatz in the experiment performed in 4.1, which is given by `qml.StronglyEntanglingLayers` from *Pennylane* (Bergholm et al., 2018). The circuit designed is inspired by the work in (Schuld et al., 2020). Each rotation operator given in the diagram contains three trainable parameters. Eq. 8 shows the matrix representation of these operators.

$$R(\phi,\theta,\omega) = R_z(\omega)R_y(\theta)R_z(\phi) = \begin{pmatrix} e^{-i(\phi+\omega)/2}\cos(\theta/2) & -e^{i(\phi-\omega)/2}\sin(\theta/2) \\ e^{-i(\phi-\omega)/2}\sin(\theta/2) & e^{i(\phi+\omega)/2}\cos(\theta/2) \end{pmatrix} \tag{8}$$

A.2. Quantum approximate optimization algorithm ansatz and Hamiltonian

In 4.2, we follow the example laid out in Ref. (Ceroni, 2020). Let n be a positive integer denoting the number of layers being used, and let $\boldsymbol{\gamma}, \boldsymbol{\alpha} \in \mathbb{R}^n$. The ansatz used in general QAOA algorithms is given by

$$U(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = e^{-i\alpha_n H_M} e^{-i\gamma_n H_C} \dots e^{i\alpha_1 H_M} e^{-i\gamma_1 H_C} \tag{9}$$

where the cost Hamiltonian H_C and mixer Hamiltonian H_M are operators that properly encode the problem we want to solve. For the MVCP, we use `qml.qaoa.cost.min_vertex_cover` to retrieve the desired Hamiltonians.

A.3. Additional numerical results

In 4.2, we noticed that PGD will "ruin" an optimal initial point by increasing the chance of converges into a suboptimal solution. In Fig. 9, we can also see that PGD can also ruin a suboptimal initial point. In Fig. 10, we plot the probability distribution of the suboptimal ansatz from Fig. 7. Contrary to Fig. 11, this does not give a good solution to the MVCP.

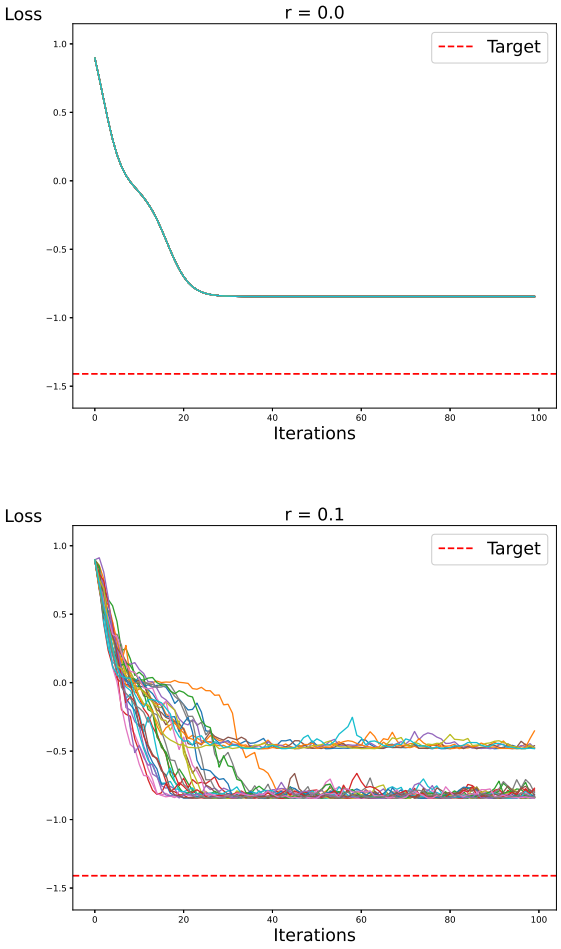


Figure 9. QAOA on MVCP. Top: result of GD on an initial point that leads to a suboptimal local optima. Bottom: results of 30 trials of PGD with perturbation radius $r = 0.1$ on the same initial point. Gradients were evaluated exactly.

In Fig. 12, we plot the solutions given by GD and PGD on the trials from Fig. 7 onto the cost landscape to gain a better visual understanding of the result.

Additional layers in the ansatz also has significant effect. In Fig. 11, we show the probability distributions of the outputs of the optimal ansatz with 1 and 2 layers.

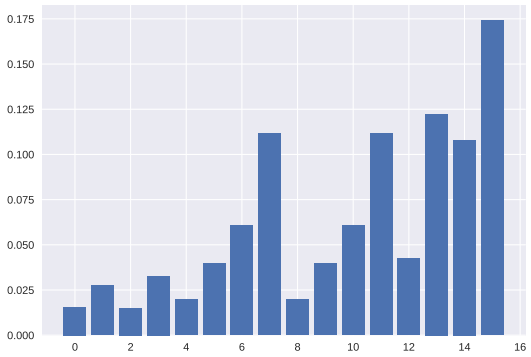


Figure 10. Probability distribution of the optimized ansatz in section 4.2. The most likely outcome is output 15, which corresponds to the set of all vertices.

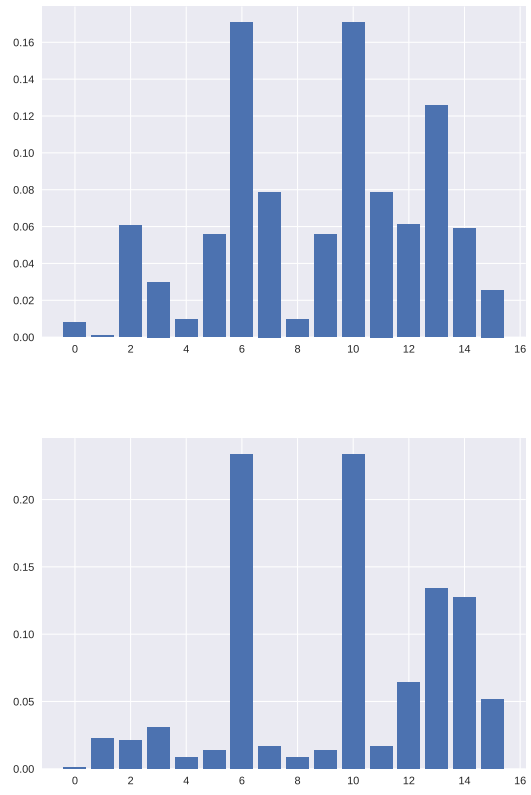


Figure 11. Probability distributions of the optimized ansatz in section 4.2. Top: probability distribution of the optimized ansatz with 1 layer. Bottom: probability distribution of the optimized ansatz with two layers. Outputs 6 (i.e. $|0110\rangle$ corresponding to the set of vertices $\{1, 2\}$) and 10 (i.e. $|1010\rangle$ corresponding the set of vertices $\{0, 2\}$) are the most likely outcomes, and consequently the solutions to the MVCP.

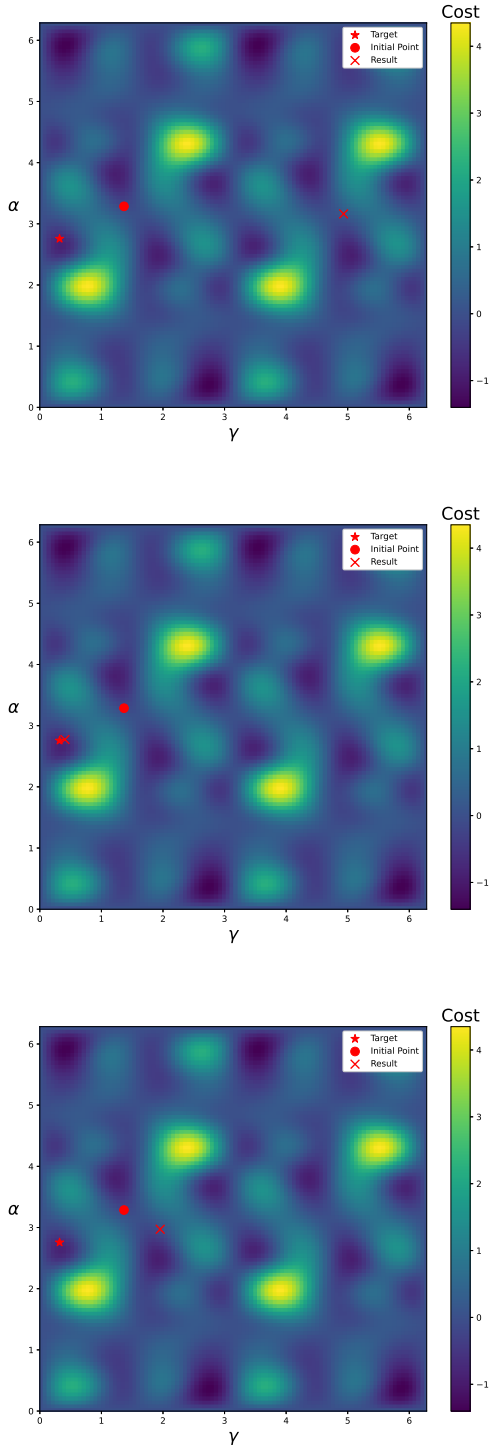


Figure 12. PGD outcomes from the result in Fig. 7. Top: result of GD that converges to a saddle point. Middle: result of PGD that converges to the optimal solution. Bottom: result of PGD that converges to a suboptimal solution.