Algorithm selection and configuration for Noisy Intermediate Scale Quantum methods for industrial applications

Moussa, C.

Chapter 6

Performance comparison of optimization methods on variational quantum algorithms

While for QAOA one can leverage concentration properties to avoid relying heavily on a classical optimizer as discussed in Chapter 4, this is not the case for other VQAs. In the case of VQE for quantum chemistry applications presented in chapter 2.3, fast and reliable classical optimization algorithms are required. Hence, understanding and optimizing how off-the-shelf optimization methods perform is an important research area. In this chapter[1], we study the performance of two commonly used gradient-free optimization methods: CMA-ES, and SPSA. We do so on the task of finding ground-state energies of a range of small chemistry and material science problems. SPSA was used frequently as an optimizer for VQE while CMA-ES is a state-of-art optimizer for difficult optimization problems in continuous search spaces. Hence, the underlying motivation was to benchmark both optimizers. We find that, with proper hyperparameter tuning, CMA-ES is competitive with and sometimes outperforms SPSA.

[1]Contents of this chapter are based on [24];Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E. O'Brien. Performance comparison of optimization methods on variational quantum algorithms. *Phys. Rev. A*, 107:032407, Mar 2023.

6.1 Introduction

The performance of VQAs is dependent on the ability of classical optimization algorithms to solve such tasks. Finding their limitations for different VQA tasks is very important in research and industry applications. To this end, optimization algorithms can be benchmarked on a wide variety of systems. However, when we worked on the topic, no extensive performance comparison of the most common optimization methods existed for chemistry and material science problems yet. The Simultaneous perturbation stochastic approximation algorithm (SPSA) [182, 183] seemed most often used though as it was designed for optimization on noisy functions. As stated previously, our goal was to benchmark CMA-ES, a state-of-art optimizer for difficult optimization problems in continuous search spaces, against the more common SPSA for VQA on several chemistry and material science problems.

We first carried out hyperparameter tuning for CMA-ES and SPSA from which we conclude that the two methods are comparable in performance across many problems. One can outperform the other depending on the task. Additionally, the accuracy of the optimized parameters is investigated. For this purpose, we define a 'sampling noise floor': a limit on the accuracy that an optimizer can achieve when the optimal parameters correspond to the best-ever function evaluation. We demonstrated numerically that CMA-ES can outperform this 'sampling noise floor'.

The structure of the chapter is as follows. Section 6.2 presents the settings of running VQE. We present the optimizers and the systems considered in Section 6.3. Section 6.4 presents the results of hyperparameter tuning while Section 6.5 concerns the accuracy of the optimized parameters. Finally, we conclude this chapter in Section 6.6.

6.2 VQE methods

We have seen in Chapter 2 the typical VQA workflow and how they are applied for chemistry under the naming VQE in Chapter 2.3. We remind here how VQE works. Given a PQC preparing the state $|\Psi(\boldsymbol{\theta})\rangle$ and an observable O specifying the problem, the cost function to optimize is:

$$\mathcal{C}(\boldsymbol{\theta}) = \langle O \rangle = \langle \Psi(\boldsymbol{\theta}) | O | \Psi(\boldsymbol{\theta}) \rangle.$$

In order to measure the expectation value of O on a quantum computer, it is

typical to write O as a linear combination of easy-to-measure operators, i.e., the Pauli operators $\hat{P}_i \in \{\mathbb{I}, X, Y, Z\}^{\otimes N}$. By linearity of the expectation operator, we get:

$$O = \sum_i c_i \hat{P}_i \rightarrow \mathcal{C}(\boldsymbol{\theta}) = \langle O \rangle = \sum_i c_i \langle \hat{P}_i \rangle. \tag{6.1}$$

Such a cost function is estimated by measuring many times many circuits for each Pauli-based operator above-mentioned. Such estimation is then passed to a classical optimization algorithm to find the set of parameters minimizing $\mathcal{C}(\boldsymbol{\theta})$.

The total number of samples (also called shots) to be measured on quantum devices is usually of the order of $\sim 10^9$. Such limit calls for a balance between exploration and estimation of the cost function such that the optimal parameters are reliably obtained. In naive settings, the total number of shots per Pauli operator is fixed. We refer to this approach as one-stage optimization. Cade et al. [36] split the total shot budget between three stages, showing improved performance with VQAs.

Our numerical experiments are performed with different shot budgets of 10^7, 10^8, and 10^9. On the one-stage method, the total number of function evaluations is fixed to 10^4 and 10^3, 10^4, and 10^5 shots per Pauli operator per function call used respectively. Within the three-stage procedure, for a fair comparison, the function calls are fixed at $7150 - 2145 - 715$, and the shots per Pauli operator at every stage are $10^2 - 10^3 - 10^4$, $10^3 - 10^4 - 10^5$, and $10^4 - 10^5 - 10^6$, respectively.

6.3 Optimizers and systems considered

In this work, two gradient-free optimization algorithms are compared across multiple problems of different sizes:

1. Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [78] is a state-of-art population-based optimization algorithm using self-adaptation of its internal variables to the energy landscape.

2. Simultaneous perturbation stochastic approximation algorithm (SPSA) [182, 183] employs a stochastic perturbation vector to compute simultaneously an approximate gradient of the objective and performs well on noisy functions.

As for their implementation, we use PyCMA [76] for CMA-ES and a modified version of SPSA based on the code in [124].

Our numerical results use different chemistry and material science problems. For VQE, we require their hamiltonians (generated via the open-source electronic structure

H_2O	Equilibrium	Stretched
O	(0.0, 0.0, 0.1173)	(0.0, 0.0, 0.0)
H	(0.0, 0.7572, -0.4692)	(0.0, 1.8186, 1.4081)
H	(0.0, -0.7572, -0.4692)	(0.0, -1.8186, 1.4081)

Table 6.1: Table describing the configurations of the atoms for the two water molecule problems used in this work.

System	# Parameters
H_4 chain	14
H_4 square	10
H_2O eq.	26
H_2O stret.	26
Hub. 1x6	15
Hub. 2x2	6
Hub. 2x3	16

Table 6.2: Number of parameters of the ansatze for each target problem.

package OpenFermion [126]) and the ansatz. The systems are presented in Table 6.2 with the respective number of parameters of the ansatz used.

The first kind of hamiltonians are Fermi-Hubbard Hamiltonians which describe the behavior of fermions (fundamental particles such as electrons) on a lattice of $n_x \times n_y$ sites. The variational circuit for this task is from Cade et al. [36].

The second kinds are molecular systems in different configurations (specified by $3D$ coordinates): H_4 and H_2O. For the H_4 in the chain configuration, the first hydrogen atom is located at 0.0 in all coordinates, then every atom is separated in the x-direction by 1.5Å. In the square configuration, we fix the hydrogen atoms in 2-dimensions. The positions of the atoms are parametrized by their polar coordinates with $R = 1.5$Å and $\theta = \frac{\pi}{4}$, and we locate them at $(x, y, 0), (x, -y, 0), (-x, y, 0), (-x, -y, 0)$ with $x = R\cos(\theta)$ and $y = R\sin(\theta)$. For the water molecule problems, the (x, y, z)-coordinates of the atoms are given in Table 6.1. Concerning the ansatz, we use the Unitary Coupled-Cluster ansatz [152, 56, 160], state-of-art for chemistry applications.

6.4 Hyperparameter tuning results

After presenting in the previous sections how a VQE is run and the problems considered, we start benchmarking under optimal hyperparameters found. In this work, we

use the iterated racing for automatic algorithm configuration or shortly IRACE [121], to tune the settings of SPSA and CMA-ES for the molecular systems. Additionally, we perform hyperparameter tuning of CMA-ES for the Hubbard model on three different configurations; 1×6, 2×2, and 2×3. For SPSA, however, we take the results of [36] where its hyperparameters were optimized.

For comparison, we use the relative energy error,

$$\Delta_r E = \left| \frac{\mathcal{C}(\boldsymbol{\theta}_{\text{opt}}) - E_0}{E_0 - c_0} \right|, \tag{6.2}$$

where $\mathcal{C}(\boldsymbol{\theta}_{\text{opt}})$ is the noiseless cost function evaluated at the optimized parameters $\boldsymbol{\theta}_{\text{opt}}$ obtained from a noisy optimization. E_0 is the lowest eigenvalue of the problem, computed exactly. c_0 is the coefficient of the identity operator, which is the largest term of the Hamiltonian and can be measured with exact precision. The results of these numerical simulations are shown in figure 6.1.

SPSA performs better in the weakly-correlated problems H_4 chain, H_2O equilibrium, and 2×2 Hubbard model. For the more interesting strongly-correlated systems (categorized as more challenging in chemistry), CMA-ES slightly outperforms with the mean values following mostly within error bars (orange ticks in fig. 6.1). Finally, we observe that CMA-ES starts to outperform SPSA when the system size and number of parameters increase. Such scenarios are more interesting for quantum simulations. We leave the study of larger systems for future work.

6.5 The sampling noise floor

In VQA, given the real objective $\mathcal{C}(\boldsymbol{\theta})$, one obtains optimal parameters returned dealing with a sampled version $\bar{\mathcal{C}}$ with variance $\text{Var}[\bar{\mathcal{C}}]$. By construction, it is possible that the evaluation of the outputted parameters is lower than its corresponding noiseless evaluation due to statistical fluctuations. Thus, by considering the parameters achieving the best-ever function evaluation, we can obtain worse results than the global minimum (assuming its existence).

Let us assume $\mathcal{C}(\boldsymbol{\theta})$ has a global minimum $\boldsymbol{\theta}_g$ with noiseless value $\mathcal{C}_g$. Then, under sampling noise, by evaluating the cost over multiple realizations of $\boldsymbol{\theta}$, including one at $\boldsymbol{\theta}_g$, we end up with a confidence interval for the evaluation at $\boldsymbol{\theta}_g$ with probability $1 - p$:

$$\Delta_p = [\mathcal{C}_g + m(p)\sqrt{\text{Var}[\bar{\mathcal{C}}]}, -\inf), \tag{6.3}$$

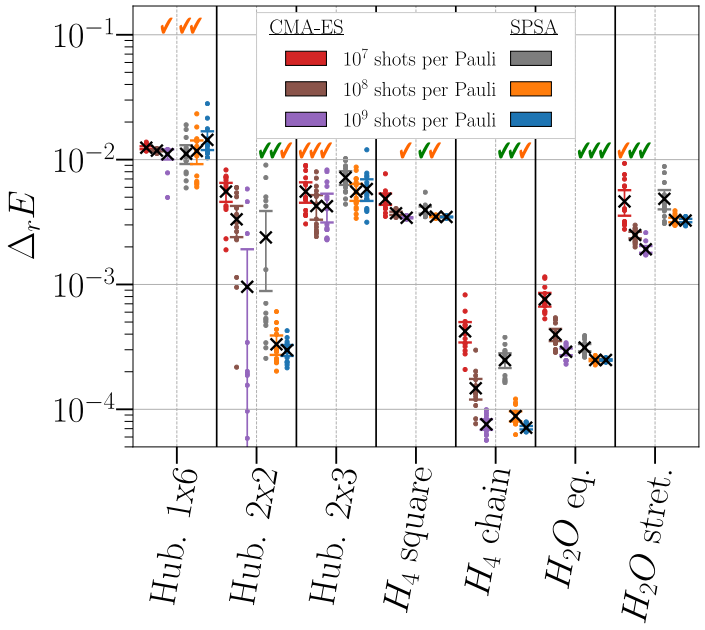

Figure 6.1: Comparison of optimized hyper-parameters of CMA-ES (red, brown, and purple dots) and SPSA (grey, orange, and blue dots). The black cross depicts the mean value, with the error bar showing the 95% confidence interval of 15 independent runs. The green ticks indicate that the optimization wins overall with a better mean and without overlapping in standard error. The orange ticks indicate the optimization wins in mean value, but its standard error overlaps with one or more optimization methods.

where $m(p) \sim \log(p)$ defines the size of the interval for the distribution of $\bar{\mathcal{C}}$. Assuming this distribution is symmetric, for $\boldsymbol{\omega} \neq \boldsymbol{\theta}_g$ satisfying $\mathcal{C}(\boldsymbol{\omega}) - m(p)\sqrt{\mathrm{Var}[\bar{\mathcal{C}}]} \notin \Delta_p$, $\mathcal{C}(\boldsymbol{\omega})$ will lie outside Δ_p with probability strictly greater than $1-p$. Thus, with probability strictly greater than $1-p^2$, $\boldsymbol{\theta}_g$ can be correctly identified as optimal parameters. However, when the conditions are not met, the true minimum cannot be determined and alternative candidates can be drawn with probability p from the region:

$$\Omega(p) = \{\boldsymbol{\omega} \colon \mathcal{C}(\boldsymbol{\omega}) < \mathcal{C}(\boldsymbol{\theta}_g) + 2m(p)\sqrt{\mathrm{Var}[\bar{\mathcal{C}}]}\}, \tag{6.4}$$

True cost values an optimizer returning the best-measured candidate in this region lie in:

$$\mathbb{C}_p = \left[\mathcal{C}(\boldsymbol{\theta}_g), \mathcal{C}(\boldsymbol{\theta}_g) + 2m(p)\sqrt{\mathrm{Var}[\bar{\mathcal{C}}]}\right], \tag{6.5}$$

The quantity $2m(p)\sqrt{\mathrm{Var}[\bar{\mathcal{C}}]}$ is defined as the *sampling floor*. To be completely defined, the value of p should be set but it is not accessible as it depends upon the optimizer's convergence. Yet, one can demonstrate numerically the effect of the *sampling floor* on the candidates returned by optimizers.

CMA-ES returns two different candidates; the best-ever measured and a so-called favourite which uses all accumulated prior information during optimization. Such information contains many more shots than a single function call. In principle, the sampling noise can be averaged out and beat the sampling noise floor. We investigate such a phenomenon in this section. In figure 6.2, we present the results of the sampling noise floor on the optimization performance. For every system, we obtained the evaluations: $\bar{\mathcal{C}}(\boldsymbol{\theta}_{\mathrm{best}}), \mathcal{C}(\boldsymbol{\theta}_{\mathrm{best}})$ and $\mathcal{C}(\boldsymbol{\theta}_{\mathrm{fav}})$. The energy error is then computed as:

$$\Delta E = \frac{\mathcal{C}(\boldsymbol{\theta}_{\mathrm{opt}}) - E_0}{|E_0 - c_0|}. \tag{6.6}$$

Firstly, we observe that the best function evaluation (orange points) is often below the true energy due to sampling noise. The comparison should be done between $\mathcal{C}(\boldsymbol{\theta}_{\mathrm{best}})$ (red points) and $\mathcal{C}(\boldsymbol{\theta}_{\mathrm{fav}})$ (purple points). The average value of $\mathcal{C}(\boldsymbol{\theta}_{\mathrm{best}})$ estimates the sampling floor. In all cases, we observe the average evaluation of the favourite candidate is below the best candidate's. By using the favourite, CMA-ES overcomes the sampling floor. For the Hubbard model and the H_4 systems, this is not significant (up to a 95% confidence interval), but for the two geometries of the water molecule, the difference is much larger (up to a 3-fold reduction of error). Such a difference between problems may come from the different optimization landscapes

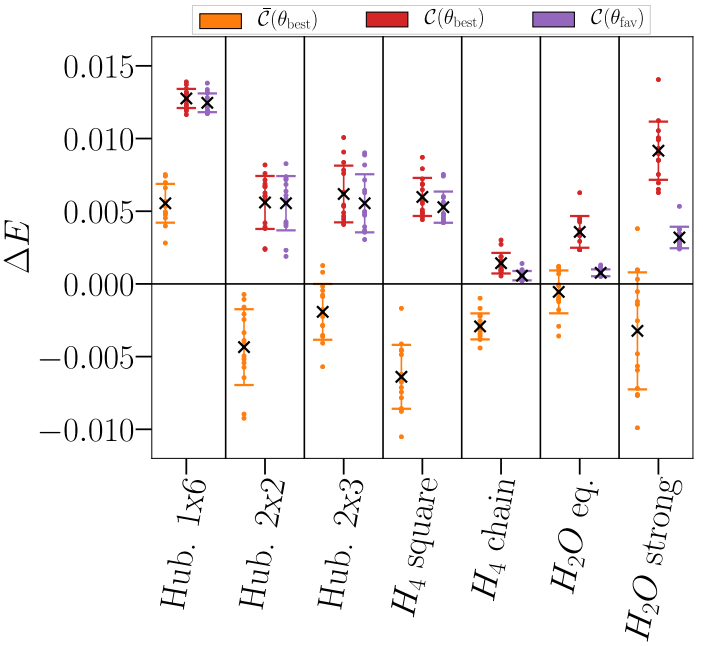

Figure 6.2: Comparison of the cost function evaluated at the best-ever measured and favourite candidate given by CMA-ES optimization under noisy optimization for the problems considered in this work. Each optimization uses 10^7 shots per Pauli over the course of the entire experiment: individual estimations of $\bar{\mathcal{C}}(\boldsymbol{\theta})$ are made using only 10^4 shots per Pauli. On the y-axis, we depict relative energy error where the column shows the mean and the overlaying points are the values of these energies of 15 independent runs. For each problem, from left to right we plot: (orange) the best-ever measured function evaluation during the CMA-ES optimization, (red) the best-ever candidate evaluated without noise, and (purple) the favourite candidate guessed by CMA-ES, evaluated without noise.

of the different problems. We leave this study for future work.

6.6 Conclusion

VQA algorithms require careful investigation of the performances of the underlying optimizers they depend on. Such optimizers may have been benchmarked using other types of problems. However quantum applications can be inherently different and optimizers will require to be adapted or discarded for tackling such settings. In this chapter, we benchmarked two state-of-art gradient-free optimizers on several chemistry and material science problems: SPSA and CMA-ES.

The optimizers were compared under a one-stage and three-stage sampling method from Ref. [36]. Hyperparameter optimization is performed showing comparable performances between SPSA and CMA-ES. The latter obtains better results on the more challenging systems. Additionally, we study the effect of sampling noise on the optimization performance using CMA-ES. Indeed, as the evaluation is not exact, using the best-ever evaluation out of the optimizer can be misleading. To overcome such a problem that we named *sampling noise floor*, using the favourite candidate yielded by CMA-ES was a better strategy. We leave as future work more benchmarking using larger systems of interest and against other VQA-tailored optimizers.