A Comparison of Various Classical Optimizers for a Variational Quantum Linear Solver

Aidan Pellow-Jarman · Ilya Sinayskiy · Anban Pillay · Francesco Petruccione

Received: date / Accepted: date

Abstract  Variational Hybrid Quantum Classical Algorithms (VHQCAs) are a class of quantum algorithms intended to run on noisy intermediate-scale quantum (NISQ) devices. These algorithms employ a parameterized quantum circuit (ansatz) and a quantum-classical feedback loop. A classical device is used to optimize the parameters in order to minimize a cost function that can be computed far more efficiently on a quantum device. The cost function is constructed such that finding the ansatz parameters that minimize its value, solves some problem of interest. We focus specifically on the Variational Quantum Linear Solver (VQLS), and examine the effect of several gradient-free and gradient-based classical optimizers on performance. We focus on both the average rate of convergence of the classical optimizers studied, as

A. Pellow-Jarman  
University of Kwa-Zulu Natal, University Road, Chiltern Hill, Westville, 3629  
Tel.: +27 84 628 0070  
E-mail: aidanpellow@gmail.com

I. Sinayskiy  
University of Kwa-Zulu Natal, University Road, Chiltern Hill, Westville, 3629  
National Institute for Theoretical Physics (NITheP), KwaZulu-Natal, 4001, South Africa  
E-mail: sinayskiy@ukzn.ac.za

A. Pillay  
University of Kwa-Zulu Natal, University Road, Chiltern Hill, Westville, 3629  
Centre for Artificial Intelligence Research (CAIR), cair.org.za  
E-mail: pillayw4@ukzn.ac.za

F. Petruccione  
University of Kwa-Zulu Natal, University Road, Chiltern Hill, Westville, 3629  
National Institute for Theoretical Physics (NITheP), KwaZulu-Natal, 4001, South Africa  
School of Electrical Engineering, KAIST, Daejeon, 34141, Republic of Korea  
E-mail: petruccione@ukzn.ac.za
well as the distribution of their average termination cost values, and how these are affected by noise. Our work demonstrates that realistic noise levels on NISQ devices present a challenge to the optimization process. All classical optimizers appear to be very negatively affected by the presence of realistic noise. If noise levels are significantly improved, there may be a good reason for preferring gradient-based methods in the future, which performed better than the gradient-free methods with only shot-noise present. The gradient-free optimizers, Simultaneous Perturbation Stochastic Approximation (SPSA) and Powell’s method, and the gradient-based optimizers, AMSGrad and BFGS performed the best in the noisy simulation, and appear to be less affected by noise than the rest of the methods. SPSA appears to be the best performing method. COBYLA, Nelder-Mead and Conjugate-Gradient methods appear to be the most heavily affected by noise, with even slight noise levels significantly impacting their performance.

Keywords variational algorithm · classical optimizer · near-term quantum device · NISQ · VQLS

1 Introduction

Since the advent of quantum computing, many promising quantum algorithms have been developed, offering up to exponential speed-up over their classical analogues. These include Shor’s Algorithm \[1\] for factoring integers, the Quantum Fourier Transform \[2\] for the Discrete Fourier Transform, and the Harrow-Hassidim-Lloyd algorithm \[3\] for solving the quantum linear systems problem. While these algorithms demonstrate the potential of quantum computing on fault-tolerant devices, they are not yet realisable in practice, as current quantum devices are too noisy and have too few qubits available.

Variational Hybrid Quantum Classical Algorithms (VHQCAs) were proposed, initially with the Variational Quantum Eigensolver \[4\], as quantum algorithms that may be successfully implemented on noisy intermediate-scale quantum \[5\] (NISQ) devices. VHQCAs employ a parameterized quantum circuit (ansatz) that computes a cost function, as part of a quantum-classical feedback loop, which optimizes the selection of the ansatz parameters. The cost function is constructed such that finding the ansatz parameters which minimize its value, finds the solution to some problem of interest; this corresponds to finding the ground-state of some Hamiltonian in question. The Variational Quantum Linear Solver \[12\] \[13\] (VQLS) is a proposed VHQCA intended to solve the quantum linear systems problem on near-term quantum devices.

The standard implementation of a VHQCA requires the selection of three components: the ansatz circuit, the cost function and the classical optimizer. The choice of classical optimizer is of interest in this paper. This choice is especially important for NISQ devices with high noise levels and shallow ansatz circuits with a limited overlap with the corresponding Hilbert space, which makes the optimization very difficult.
Several other papers have done various different comparisons of the performance of classical optimizers in VHQCAs [6–10]. Two new surrogate model-based optimization methods (Model Gradient Descent and Model Policy Gradient) are introduced in [6] and compared with Nelder-Mead, Bounded Optimization By Quadratic Approximation (BOBYQA) and Simultaneous Perturbation Stochastic Approximation (SPSA) in the context of the VQE. This paper [6] highlights the need to use relevant cost models to make fair comparisons between optimization techniques, and the need to optimize hyper-parameters of existing classical algorithms to maximize performance. [7] compares Broyden–Fletcher–Goldfarb–Shanno (BFGS), ADAM and Natural Gradient Descent in the context of the VQE. In [8], a comparison is made between 4 gradient-free optimizers, Nonlinear Optimization by Mesh Adaptive Direct Search (NOMAD), Implicit Filtering (ImFil), Stable Noisy Optimization by Branch and FIT (SnobFit) and BOBYQA in the context of the VQE. These results also highlight the need for hyper-parameter optimization. Two new optimization techniques are introduced in [9], combining randomized perturbation gradient estimation with adaptive momentum gradient updates to create the AdamSPSA and AdamRSGF algorithms. Single qubit optimal control is used as a test of the optimization algorithms and it is shown that these new algorithms perform exceptionally well. A comparison between Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS), Constrained Optimization by Linear Approximation (COBYLA), RBFOpt, Modified Powell’s and SPSA using a noise-free implementation of the VQE to solve 6 combinatorial optimization problems is presented in [10]. The results here indicate global optimization methods are more successful than local optimization methods, due to their avoidance of local optima.

This work differs significantly from the above mentioned in a variety of ways:

– by collectively considering the effect of noise on the optimizer, and the choice of either gradient-based vs gradient-free optimizers,
– by using a practical cost comparison between optimizers based on the resources needed to perform cost function evaluations,
– by the selected set of classical optimizers being compared, and
– by selecting the variational quantum linear solver as the problem of interest.

The aim of the present study is to make a practical attempt to draw a meaningful realistic comparison between a large selection of classical optimizers. The rest of the paper is structured as follows: Section 2 introduces the variational quantum linear solver and the available classical optimizers, and Section 3 describes the experimental setup of the comparison. Section 4 provides the results and a discussion, and Section 5 concludes.

2 Variational Quantum Linear Solver

Solving a system of $N$ linear equations with $N$ unknowns, expressible as $A\vec{x} = \vec{b}$, involves finding the unknown solution vector $\vec{x}$, satisfying the equation.
The quantum linear systems problem \([11]\) (QLSP) is the quantum analogue to solving a system of linear equations on a classical computer, defined as follows: Let \(A\) be an \(N \times N\) hermitian matrix (however this algorithm is not limited to a hermitian matrix; the algorithm can easily be adapted to handle the non-hermitian case) and let \(\vec{x}\) and \(\vec{b}\) be \(N\) dimensional vectors, satisfying \(A\vec{x} = \vec{b}\), having corresponding quantum states \(|x\rangle\) and \(|b\rangle\), such that

\[
|x\rangle := \frac{\sum_i x_i |i\rangle}{||\sum_i x_i |i\rangle||_2},
\]

\[
|b\rangle := \frac{\sum_i b_i |i\rangle}{||\sum_i b_i |i\rangle||_2}.
\]

Given access to the matrix \(A\) by means of an oracle and a unitary gate \(U\) such that \(U|0\rangle = |b\rangle\), output a quantum state \(|x'\rangle\) such that \(|||x\rangle - |x'\rangle||_2 \leq \epsilon\), where \(\epsilon\) is the error-bound of the approximate solution. This QLSP is the form of the problem solved by the HHL algorithm.

The input to the Variational Quantum Linear Solver are the matrix \(A\) and vector \(\vec{b}\). \(A\) is given in a slightly different manner than in the QLSP stated above. Instead of requiring oracle access to a hermitian matrix \(A\), \(A\) is only required to be given as a linear combination of unitary matrices \(A_i\) implemented as unitary gates, such that \(A = \sum_{i=1}^n c_i A_i\), \(c_i \in \mathbb{C}\). This is equivalent to requiring that \(A\) be given as a linear combination of Pauli operators. The value of \(\vec{b}\) is given as a unitary gate \(U\), such that \(U|0\rangle = |b\rangle\) as per the QLSP.

The VQLS algorithm runs in a simple feedback loop [Figure 4], whereby a classical optimizer finds the optimal parameters for the ansatz circuit, by iteratively evaluating the cost function on the quantum device and updating the parameters, until a minimum cost value is achieved. Let the parameterized ansatz be denoted by \(V(\alpha)\), and let the optimal parameters be denoted by \(\alpha^*\). At the algorithms termination, \(V(\alpha^*) |0\rangle = |x'\rangle\), where \(|||x\rangle - |x'\rangle||_2 \leq \epsilon\), with \(|x\rangle\) being the exact solution to the linear system, and \(\epsilon\) being the error-bound of the approximate solution.

The choice of classical optimizer, ansatz circuit and cost function are free to be decided in the implementation of the VQLS algorithm. The choice of classical optimizer is discussed at length and is the main focus of this paper. The ansatz choice falls into either the category of hardware-efficient (agnostic) ansatze, and problem specific ansatze. Hardware-efficient ansatze make no use of the known structure of the problem, in this case being \(A\) and \(\vec{b}\), and instead are optimized for the specific quantum hardware available. They therefore have greater noise-resistance, making them the more realistic near-term option. Their shortfall is that there is usually no guarantee that they contain any close-approximation of the solution, due to their limited span of the relevant Hilbert space. Problem specific ansatze are designed to specifically take the structure of the problem into account, making them more suited to
A Comparison of Classical Optimizers for a VQLS

Fig. 1: Basic Variational Quantum Linear Solver Schematic. The algorithm runs in a simple feedback loop, whereby a classical optimizer finds the optimal parameters for the ansatz circuit, by iteratively evaluating the cost function on the quantum device and updating the parameters, until a minimum cost value is achieved. $V(\alpha)$ denotes the parameterized ansatz, $\alpha^*$ denotes the optimal parameters. At termination, $V(\alpha^*)\ket{0} = \ket{x'}$ and $||\ket{x} - \ket{x'}||_2 \leq \epsilon$, with $\ket{x}$ being the exact solution to the linear system, and $\epsilon$ being the error-bound of the approximate solution.

finding the solution to a specific problem. An example of one such ansatz is the QAOA ansatz [12,13], which involves performing repeated Hamiltonian application of Hamiltonian simulations of a driver and mixer Hamiltonian constructed specifically from the linear systems problem in question. These ansatze are limited by the Hamiltonian simulation which is not as near-term as the implementation of the hardware-efficient ansatz.

2.1 Classical Optimizers

There are many different classical optimizers available, both gradient-free and gradient-based, that can be used in VHQCAs for ansatz parameter optimization. Gradient-based methods require gradient information during the optimization process, which can either be obtained analytically by a quantum gradient function, or estimated by means of repeated cost
Gradient-Free
Constrained Optimization by Linear Approximation\(^\alpha\) (COBYLA) [14]
Nelder-Mead\(^\alpha\) [15]
Modified Powell’s method\(^\alpha\) [16]
Simultaneous Perturbation Stochastic Approximation\(^\beta\) (SPSA) [17]

Gradient-Based
Broyden–Fletcher–Goldfarb–Shanno algorithm\(^\alpha\) (BFGS) [18]
Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm\(^\alpha\) (L-BFGS) [19]
Conjugate Gradient method\(^\alpha\) (CG) [20]
Sequential Least Squares Programming\(^\alpha\) (SLSQP) [21]
ADAM\(^\beta\) [22]
AMSGrad\(^\beta\) [23]

Table 1: Table of Classical Optimizers Considered - The implementation of all classical optimizers are taken from the Scipy\(^\alpha\) and Qiskit\(^\beta\) Python libraries, under the respective functions `scipy.optimize.minimize` and `qiskit.aqua.components.optimizers`. (References given in the far-right column)

function evaluations. Gradient-free methods operate as black-box optimizers, which require no extra information besides the values of the cost function evaluations. The common classical optimizers to be compared are given in Table 1.

In this paper, the gradient-based optimizers all use an analytic gradient calculated by a quantum gradient function. It must also be noted that the hyper-parameter settings of all classical optimizers are left as standard in their implementations, except for the ‘rhobeg’ value of the COBYLA optimizer, which is set to \(\pi\), and the learning rate of ADAM and AMSGrad, which is set to 0.1.

3 Comparison Setup

The classical optimizer comparison is explained below. The test problem instances are introduced and then the selection of the ansatz and cost function are discussed. Next the effect of noise taken into account in the simulations is explained, and finally a method for resource comparison between classical optimizers is proposed. All quantum simulations are implemented using the qiskit python module.

3.1 Test Problem Instances

Here three test instances of the VQLS algorithm are presented by matrices \(A_1\), \(A_2\) and \(A_3\), of size \(8 \times 8\), \(16 \times 16\) and \(32 \times 32\) respectively.

\[
A_1 = 1 + 0.25 \cdot Z_2 + 0.15 \cdot H_3,
\]
\[
A_2 = Z_1 + 0.15 \cdot Z_3 + 0.5 \cdot H_4,
\]
Fig. 2: Ansatz Circuit Diagrams: 1, 2, and 3 label the 3, 4, and 5 Qubit ansatze respectively. Each ansatz consists of parameterized $R_y$ single qubit rotation gates, and 2-qubit controlled-X gates

$$A_3 = H_1 + 0.25 \cdot Z_3 + 0.5 \cdot X_4,$$

where $Z_i, (i = 1, 2, 3)$ indicates the matrix formed by the tensor product, with gate $Z$ applied to qubit $i$ and the identity gate applied to the remaining qubits. Notation is similarly defined with $H$ and $X$. $I$ indicates an $N \times N$ identity matrix.

In all problem instances, the state $|b\rangle = H^\otimes n |0\rangle^n$, for $n = 3, 4, 5$ respectively. The problems are real-valued, minimizing the complexity of the ansatz choice.

3.2 Test Ansatz and Cost Function Choice

For each of these three test instances, a very simple hardware-efficient ansatze, consisting of parameterized $R_y$ single qubit rotation gates, and 2-qubit controlled-X gates, has been used. These ansatze are designed specifically to deal with real-valued problems, such as the ones under consideration. This is not a shortfall of the VQLS algorithm, and were the problem in question complex-valued, the appropriate gates could be added to the ansatz.

The ansatze are also not optimized in any way specific to these problem instances, since only the relative performance of the different classical optimizers is of interest. They are instead designed to simulate hardware-efficient ansatze, adding some level of noise resistance. They do this by making limited use of 2-qubit gates, with assumed limited backend connectivity between qubits. Here no standard backend configuration is
employed; instead it is assumed only qubits between 2-qubit gates are connected.

It is known that the initialization parameters of the ansatz can have a great impact on the optimization of its parameters, and accordingly the initial parameters have been set randomly for each of the 100 runs for the 3 problem instances, but uniformly across all of the classical optimizers and all noise levels, giving a fair comparison of the optimizers and noise levels from identical initialization parameters. Any two classical optimizers on the same problem instance, will have the same 100 random initializations used across all noise levels.

The cost function used for these VQLS instances was proposed in [12] as a local version of the global cost function, intended to improve ansatz optimization by avoiding the exponentially vanishing gradients that global cost functions experience as the number of qubits increases. This local cost function is given by,

\[
C_L = \frac{\langle x | H_L | x \rangle}{\langle x | A^\dagger A | x \rangle},
\]

with the effective Hamiltonian \( H_L \) given by,

\[
H_L = A^\dagger U(1 - \frac{1}{n} \sum_{j=1}^{n} |0_j\rangle \langle 0_j| \otimes 1_j) U^\dagger A.
\]

This cost function is also selected here because it does not require controlled application of the ansatz, making it more resistant to noise. In depth analysis of this cost function as well as the computation of analytic gradients for this cost function are given by Bravo-Prieto et al. in [12].

3.3 Effect of Noise

To demonstrate the effect of noise on classical optimizers, three different quantum simulations were used. The first was a state vector simulation, having zero noise, implemented using qiskit’s Statevector Simulator. The second simulation was a perfect fault-tolerant quantum device, using qiskit’s Qasm Simulator having only shot-noise (that is the noise occurring through the sampling of expectation values in the Hadamard Test, with number of shots being set to 10 000). The amount of shot-noise is inversely proportional to the number of shots, with the state vector simulation being equivalent to a perfect fault-tolerant quantum device with an infinite number of shots. The third simulation is that of a realistic near-term quantum device having shot-noise as well as realistic noise model sampled from the IBM Vigo quantum device, and added to qiskit’s Qasm Simulator. This simulation is intended to provide the most difficult and realistic optimization challenge for the classical optimizers.
3.4 Resource Comparison

The above mentioned classical approaches were compared in terms of the number of evaluations of the cost function and, for optimizers making use of gradient function calls, the number of evaluations of the gradient function. The cost of the quantum resources required for one gradient function evaluation can be equivalently expressed in terms of a number of cost function evaluations, with one gradient function evaluation being equivalent to two evaluations of the cost function per parameter in the ansatz, as per the analytic gradient function procedure [12, 24]. Comparing classical optimizers by the number of cost function calls gives an objective way to measure the amount of quantum resources required by a classical optimizer during optimization. For the 3, 4 and 5 qubit problem instances, the optimizers are limited in this comparison to 600, 800 and 1000 evaluations of the cost function. Classical optimizers were terminated if they reach this cost function evaluation limit.

4 Results and Discussion

Box plots [Fig: 3, 4, 5] and line graphs [Fig: 6, 7, 8] record the final distribution of the cost value and average rate of convergence respectively, for the top 50 best performing optimizations out of the 100 for each problem instance and noise level. The top 50 were used in order to exclude any of the non-converging optimization attempts that commonly occurred. These non-converging attempts failed to converge almost at all from their initial cost values. This does not present so much of an issue for optimizers in general as these non-converging optimization attempts are clearly visible after the first few iterations, and so the optimization process can just be restarted from another random initialization of the ansatz parameters.

Near uniform trends are observed across all test problem instances with regards to the final distribution of the cost value at termination [Fig: 3, 4, 5], and therefore these results are discussed in general with respect to the trends observed in performance across problem instances as noise levels increase, from statevector simulation, to include shot-noise, and finally to include a noise model sampled from a real quantum device.

In the statevector simulation, for all problem instances, almost all classical optimizers performed similarly well, with ADAM and AMSGrad slightly under performing, and the gradient-based BFGS, L-BFGS and SLSQP, and the gradient-free COBYLA, slightly out performing the rest. There was not much difference between the gradient-free and gradient-based approached in general in a completely noise-free simulation.

Once shot-noise is taken into account, the gradient-based optimizers appear to outperform the gradient-free optimizers on average for all problem instances, with the exception of SPSA, which performed the best on average in all problem instances. The remaining gradient-free optimizers performance got noticeably worse in the presence of shot-noise alone. Again BFGS, L-BFGS
Fig. 3: 3 Qubit Problem Instance - Termination Value Box plot: This box plot captures the final distribution of the cost function value at termination, for the 50 best runs, for all optimizers, for all noise-levels. The lower cost value is better (SV - Statevector, Q - Qasm, QN - Qasm + Noise)

and SLSQP performed the best for the gradient-based optimizers, which all in general were less affected by the shot noise. ADAM and AMSGrad remain the under-performers in the gradient-based optimizers.

Finally when a realistic noise model is added to the simulation, there is a clear shift in the results. While all of the classical optimizers perform significantly worse under the increased noise levels, some optimizers handle the increased noise levels much better than others. The gradient-based optimizers may perform worse under realistic noise levels when compared to their performance under shot-noise because it may become increasingly difficult for the gradient-based optimizers to calculate analytic gradients with the increased noise levels, as calculating analytic gradients requires very many circuit evaluations (2 per ansatz parameter). Again SPSA appears to be the best performing classical optimizer overall, with Powell’s method also performing comparatively well within the gradient-free optimizers. AMSGrad and BFGS also perform exceptionally well under the increased noise-levels within the gradient-based optimizers. COBYLA, Nelder-Mead
and Conjugate-Gradient, which had both performed well in the statevector simulation, perform the worst in the noisy simulation.

Looking at the average convergence rate of the classical optimizers [Fig: 6, 7, 8] there is a general trend of near equally fast convergence between all approaches, in each respective problem instance and noise level. The relative performance of the optimizers is also uniform across the problem instances. The decision to use gradient-free or gradient-based approaches appears to be of little consequence with respect to convergence speed as well. ADAM, AMSGrad and COBYLA serve as exceptions to the general trend, with ADAM and AMSGrad consistently converging the slowest, and COBYLA consistently converging the fastest. COBYLA also terminates early in the presence of any noise.
5 Conclusion

It is clear that the realistic noise levels on NISQ devices present a big challenge to the optimization process. In very low noise levels, gradient-based approaches appear to outperform gradient-free optimizers. The performance of classical optimizers under realistic noise levels varies heavily within either of the sets of gradient-free or gradient-based approaches. There is no clear winner in general between gradient-free and gradient-based approaches under realistic noise levels. SPSA appears to perform the best under realistic noise levels, with Powell’s method also performing relatively well in the gradient-free optimizers, and AMSGrad and BFGS performing the best in the gradient-based optimizers. Some optimizers, notably COBYLA, Nelder-Mead and Conjugate Gradient, which performed well in the noise-free environment, perform exceptionally worse in the noisy environment.

It must be noted that these results are simply an initial investigation into the relative performance of various classical optimizers available. Not much focus was paid to the setting of the hyper-parameters of the various algorithms,
and as such it may be possible to greatly improve the performance of those optimizers which are heavily dependant on these by selecting appropriate hyper-parameter settings. This is intended for a future work. These results show the relative performance of the various algorithms under differing noise levels while keeping hyper-parameters constant. As this work only employed quantum simulators, confirming these results on real quantum hardware has also been left aside for a future work. The results of this work are based upon the performance of the various classical optimizers in a variational quantum linear solver; it is left to a future work to investigate whether similar results are observed in other variational algorithms.

Acknowledgements This work is based upon research supported by the South African Research Chair Initiative, Grant No. UID 64812 of the Department of Science and Innovation of the Republic of South Africa and National Research Foundation of the Republic of South Africa.
Fig. 7: 4 Qubit Problem Instance - Convergence Rate: The average rate of convergence across the 50 best attempts, for all classical optimizers, at each noise level
Fig. 8: 5 Qubit Problem Instance - Convergence Rate: The average rate of convergence across the 50 best attempts, for all classical optimizers, at each noise level.
References

1. Peter W. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,” SIAM Journal on Computing, Vol. 26, No. 5 (1997)
2. Don Coppersmith, “An Approximate Fourier Transform Useful in Quantum Factoring,” arXiv:quant-ph/0201067
3. Aram W. Harrow, Avinatan Hassidim and Seth Lloyd, “Quantum algorithm for linear systems of equations,” Phys. Rev. Lett., Vol 15, No. 103, Pp. 150502 (2009)
4. Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik and Jeremy L. O’Brien, “A Variational Eigenvalue Solver on a Photonic Quantum Processor,” Nature Communications, Vol. 5:4213 (2014)
5. John Preskill, “Quantum Computing in the NISQ era and beyond”, Quantum 2, Vol. 79 (2018)
6. Kevin J. Sung, Jiahao Yao, Matthew P. Harrigan, Nicholas C. Rubin, Zhang Jiang, Lin Lin, Ryan Babbush and Jarrod R. McClean, “Using models to improve optimizers for variational quantum algorithms,” Quantum Sci. Technol., Vol. 5, No. 4:044008 (2020)
7. David Wiecherts, Christian Gogolin and Michael Kastoryano, “Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer,” Phys. Rev. Research 2, 043246 (2020)
8. Wim Lavrijsen, Ana Tudor, Juliane Müller, Costin Iancu and Wibe de Jong, “Classical Optimizers for Noisy Intermediate-Scale Quantum Devices,” 2020 IEEE International Conference on Quantum Computing and Engineering, Pp. 267-277 (2020) arXiv:2004.03004
9. Zhaoqi Leng, Pranav Mundada, Saeed Ghadimi and Andrew Houck, “Robust and efficient algorithms for high-dimensional black-box quantum optimization” (2019) arXiv:1910.03591
10. Giacomo Nannicini, “Performance of hybrid quantum/classical variational heuristics for combinatorial optimization,” Phys. Rev. E 99, 013304 (2019)
11. Danial Dervovic, Mark Herbst, Peter Mountney, Simone Severini, Nairi Usher and Leonard Wossnig, “Quantum linear systems algorithms: a primer” (2018) arXiv:1802.08227
12. Carlos Bravo-Prieto, Ryan LaRose, M. Cerezo, Yiğit Subaşı, Lukasz Cincio and Patrick J. Coles, “Variational Quantum Linear Solver: A Hybrid Algorithm for Linear Systems” (2019) arXiv:1909.06820
13. Hsin-Yuan Huang, Kishor Bharti and Patrick Rebentrost, “Near-term quantum algorithms for linear systems of equations” (2019) arXiv:1909.07344
14. M. J. D. Powell, “A direct search optimization method that models the objective and constraint functions by linear interpolation,” Advances in Optimization and Numerical Analysis, Pp. 51-67 (1994)
15. J. A. Nelder and R. Mead, “A simplex method for function minimization,” The Computer Journal, Vol. 7, Iss. 4, Pp. 308–313 (1965)
16. M. J. D. Powell, “An efficient method for finding the minimum of a function of several variables without calculating derivatives,” Computer Journal, Vol. 7, Iss. 2, Pp. 155–162 (1964)
17. J. C. Spall. “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,” IEEE Transactions on Automatic Control, Vol. 37, No. 3, Pp 332–341, (1992)
18. R. Fletcher, “A new approach to variable metric algorithms,” The Computer Journal, Vol. 13, Iss. 3, Pp 317–322 (1970)
19. Richard H. Byrd, Peihuang Lu, Jorge Nocedal and Ciyou Zhu, “A Limited Memory Algorithm for Bound Constrained Optimization,” SIAM Journal on Scientific Computing, Vol. 16, Pp. 1190-1208 (1995)
20. Magnus R. Hestenes and Eduard Stiefel, “Methods of Conjugate Gradients for Solving Linear Systems,” Journal of Research of the National Bureau of Standards, Vol. 49, No. 6 (1952)
21. Paul T. Boggs, Jon W. Tolle, “Sequential Quadratic Programming,” Acta Numerica, Vol. 4, Pp 1-51 (1996)
22. Diederik P. Kingma and Jimmy Lei Ba, “ADAM: A Method for Stochastic Optimization” (2017) arXiv:1412.6980
23. Sashank J. Reddi, Satyen Kale and Sanjiv Kumar, “On the Convergence of ADAM and Beyond” (2018) arXiv:1904.09237
24. K. Mitarai, M. Negoro, M. Kitagawa and K. Fujii, “Quantum Circuit Learning,” Phys. Rev. A 98, 032309 (2019)