Efficient protocol for solving combinatorial graph problems on neutral-atom quantum processors

Wesley da Silva Coelho*, Mauro D'Arcangelo*, Louis-Paul Henry*

* PASQAL SAS, France
7, rue Léonard de Vinci, 91300, Massy - France
{wesley.coelho, mauro.darcangelo, louis-paul.henry}@pasqal.com

1 Introduction

A lot of effort is currently being put into designing quantum algorithms and hardware that could provide an advantage over classical computers. This advantage can take the form of more accurate results, a faster convergence, or even a lower energy consumption. These solutions are developed on very different platforms, using a wide range of technologies. The most prominent ones are based on trapped ions [1, 2], Josephson junctions [3, 4] and Rydberg neutral atoms [5, 6]. In each case, the information is stored in a two-level system constituting the qubits. Different sets of quantum gates [7] can then be implemented and, for a given quantum algorithm, the effective quantum circuits can vary significantly across platforms. Additionally, there are problems for which even the Noisy Intermediate Scale Quantum (NISQ) processors [8] are expected to provide an advantage. This could be obtained from an *analog* approach where, as opposed to the case of *digital* quantum computing, the quantum operations are not divided into discrete consecutive steps (gates) but are rather the result of a time-dependent control of the Hamiltonian acting upon the qubits. This solution will be very intrinsically problem- and platform-specific, further complicating any comparisons.

Neutral atom quantum processors are well suited to solving combinatorial graph problems. In fact, the Ising Hamiltonian describing the dynamics of the qubits is closely related to the cost function to be minimized. Solving the problems is then equivalent to finding the ground state of the system, which can be achieved by adiabatic annealing, as it has been shown in the case of the Maximum Independent Set (MIS) problem [5]. Although most QAOA applications focus on gate-based models of quantum computing, a promising avenue for noisy devices is represented by analog variational algorithms. The analog mode of operation involves the evolution of a quantum system under a continuously controllable resource Hamiltonian rather than the discrete application of a fixed set of quantum gates.

Whereas the successful implementation of a gate-based algorithm is limited by the absence of error correction on current devices, an analog algorithm is intrinsically more resilient to noise [5]. In this framework, the role of Rydberg atom arrays is recognized as a prominent example of how the ground state of a quantum Hamiltonian directly maps to the solution of a hard

combinatorial graph problem, MIS on unit-disk graphs for instance.

On neutral atom platforms, preparing specific quantum states is usually achieved by pulse shaping, i.e., by optimizing the time-dependence of the Hamiltonian related to the system. This process can be extremely costly, as it requires sampling the final state in the quantum processor many times. Hence, determining a good pulse is one of the most important bottlenecks of the analog approach. In this work, we propose a novel protocol for solving hard combinatorial graph problems that combines variational analog quantum computing and machine learning. Our numerical simulations show that the proposed protocol can reduce dramatically the number of iterations to be run on the quantum device. Finally, we assess the quality of our approach by estimating the related Q-score, a recently proposed metric aimed at benchmarking QPUs.

2 Methodology

To the best of our knowledge, only two machine learning techniques were proposed in order to accelerate Quantum Approximate Optimization Algorithms (QAOAs). To solve combinatorial problems, Khairy et al [9, 10] propose two different machine learning-based approaches to find optimal QAOA parameters: a kernel density estimator-based model [9] that learns generative models of optimal circuit parameters, and a reinforcement learning-based model [10] that can learn different policies to predict (near-)optimal QAOA parameters. Comparing both proposed approaches and the optimization loop under limited runtime constraints, the authors showed that the optimality gap could be considerably reduced. Even though different machine learning-based approaches were proposed in order to find near-optimal parameters for circuit-based QAOA algorithms, no attention has been given to analog quantum processing on neutral-atom QPUs.

The main objective of our supervised machine learning-based approach is to automatically provide: i) the Rabi frequency and detuning values on different instants of the pulse, and ii) the total duration of the pulse. Hence, the out-coming pulse is specifically tailored to evolve the system to states that represent (near-)optimal solutions for a given combinatorial graph problem instance. In what follows, we detail each step of the proposed machine learning algorithm.

In this work, we focus on the Maximum Cut (MaxCut) and Maximum Independent Set (MIS) problems. While the MaxCut problem is equivalent to minimizing the Hamiltonian of a spin glass, the solutions of the MIS problem on unit-disk graphs can be encoded as the ground state of the Hamiltonian describing neutral-atoms devices [5].

2.1 Pulse prediction

The way combinatorial graph problems are usually solved with quantum hardware involves the optimal tuning of a set of parameters. This is usually done via an optimization loop that is

applied to each instance of the problem, which is time and resource consuming. To overcome both time and resource limitations, we propose a new supervised machine learning- based approach that automates the parameter choices and creates pulse sequences for analog quantum processes. Our model is based on the Chained Multi-Target Regression Algorithm (CMTRA) [11], which is generally used to predict multiple target values that are dependent upon the input and upon each other.

By predicting essential pulse parameters, one can considerably scale up quantum algorithms and, hence, solve bigger instances of complex combinatorial problems without dedicated optimization loops. The main objective of our supervised machine learning-based approach is to automatically provide: i) the Rabi frequency and detuning values on different instants of the pulse, and ii) the total duration of the pulse. Hence, the out-coming pulse is specifically tailored to evolve the system to states that represent (near-)optimal solutions for a given combinatorial graph problem instance.

2.2 Q-Score

The Q-score metric [12] was developed to benchmark Quantum Processing Units at a time when commercially viable NISQ applications are becoming a reality. It is application-centric, hardware-agnostic, and can be applied equally effectively on current machines as well as future large-scale devices. For these reasons, the Q-score represents to date one of the best attempts at establishing a practical standardized benchmark that can be monitored over time to assess the evolution of quantum computers in solving real problems. Essentially, the Q-score is the largest number of qubits for which a solution to the problem is at least 20% better than the average random solution.

3 Numerical Results

n our results, the score obtained stayed above the 20%, even in the presence of noise, up to the largest graphs we were able to simulate with noise in a reasonable amount of time. In order to determine the Q-score of the method and platform we need to extrapolate the results to larger problem sizes. To this end, we fit an exponential decay on the tail of the size dependence of the score $\beta(n) = \beta_0 e^{-n/n0}$. The Q-score is then given by $Q_{score} = n_0 \log(5\beta_0)$. The results are summarized in 1. For both problems, the Q-score is of the order of 80 (except for MIS of non-UD graphs on noisy devices), to be compared with the Q-score determined in [12] for QAOA on state-of-the-art gate-based QC platforms. In particular, the presence of noise does not seem to significantly lower the score. Indeed, this specific behavior of analog quantum computing is very different from what was observed in the digital quantum circuits [12], where the score degrades faster for larger circuit depths. The comparison between the two approaches

is not easy, as there is no equivalent to the circuit depth here. However, this example highlights the resilience to noises of the analog approach. For an in-deph overview of our work, one may refer to [13].

Table 1: Estimated Q-scores for MIS and MaxCut problems on Unit-Disk and non-UD graphs and in a noisy and noiseless settings

		Noiseless	Noisy
MIS	UD graphs	74±5	80±7
	Non-UD graphs	80±10	63±4
MaxCut	UD graphs	79±11	75±7
	Non-UD graphs	80±6	91±16

3 Concluding remarks

In this study, we demonstrated that it is possible, thanks to machine learning, to develop an efficient way of solving combinatorial graph problems on analog quantum processing units such as neutral atom platforms. Determining a good pulse, as well as a good embedding, to solve the problem on a given graph is one of the most important bottlenecks of the analog approach. By providing directly a good pulse, our method allows restricting the runs on the Quantum device to the sampling of a given final state, reducing dramatically the number of shots and hence the time-to-solution. We showed that it is possible to train a model to predict a pulse that prepares a final state with a sufficient overlap of (near-) optimal solutions of the problem's instances.

In this study, we chose a training set that was not fully optimized, so that its generation would not take too long. If one would be to improve the performance, one step would be to improve the training set, both by pushing further optimization for each of its instances, as well as increasing the largest graph order (i.e., number of nodes) it contains. Also, the quality of the result depends on the number of shots of the final state one allows taking (the time budget). One could also try to specifically train the model for a fixed number of shots. Alternatively, one could try a reinforcement learning scheme. In that case, the training is expected to take longer, but one would spare the generation of the training data set. A similar approach could be applied to other combinatorial graph problems.

Furthermore, each model we trained in this study had its own embedding strategy. It is worth mentioning that, because of the difficulty of embedding a generic graph, it may be more efficient to use an alternative representation of the initial problem. For example, instead of solving the MIS problem on a graph, one could solve the equivalent Maximum Clique problem on the complement graph.

Our results highlight the potential of NISQ-era, analog quantum computing. Even though the need to develop problem-specific frameworks may seem to contradict the goal of speeding-up classical calculations, it could lead to the first quantum advantageous solution.

References

[1] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Höfer, C. Wächter, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz, "Compact ion-trap quantum computing demonstra- tor," *PRX Quantum*, vol. 2, p. 020343, Jun 2021.

[2] J. Benhelm, G. Kirchmair, C. F. Roos, and R. Blatt, "Towards fault-tolerant quantum computing with trapped ions," *Nature Physics*, vol. 4, pp. 463–466, apr 2008.

[3] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting qubits : Current state of play," *Annual Review of Condensed Matter Physics*, vol. 11, no. 1, pp. 369–395, 2020.

[4] M. H. Devoret, A. Wallraff, and J. M. Martinis, "Superconducting qubits : A short review," 2004.

[5] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, "Quantum computing with neutral atoms," *Quantum*, vol. 4, p. 327, 2020.

[6] J. Wurtz, P. Lopes, N. Gemelke, A. Keesling, and S. Wang, "Industry applications of neutral-atom quantum computing solving independent set problems," 2022.

[7] C. P. Williams, *Quantum Gates*, pp. 51–122. London : Springer London, 2011.

[8] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79,

aug 2018.

[9] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Learning to optimize variational quantum circuits to solve combinatorial problems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 2367–2375, apr 2020.

[10] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Reinforcement-learning-based variational quantum circuits optimization for combinatorial problems," *arXiv preprint arXiv :1911.04574*, 2019.

[11] K. C. Demirel, A. Sahin, and E. Albey, "Ensemble learning based on regressor chains : A case on quality prediction.," in *DATA*, pp. 267–274, 2019.

[12] S. Martiel, T. Ayral, and C. Allouche, "Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way," *IEEE Transactions on Quan- tum Engineering*, vol. 2, pp. 1–11, 2021.

[13] W. d. S. Coelho, M. D'Arcangelo, and L.-P. Henry, "Efficient protocol for solving combinatorial graph problems on neutral-atom quantum processors," 2022, DOI : 10.48550/ARXIV.2207.13030.