Workshop on Quantum Artificial Intelligence 2023

27 – 28 July 2023, Naples, Italy

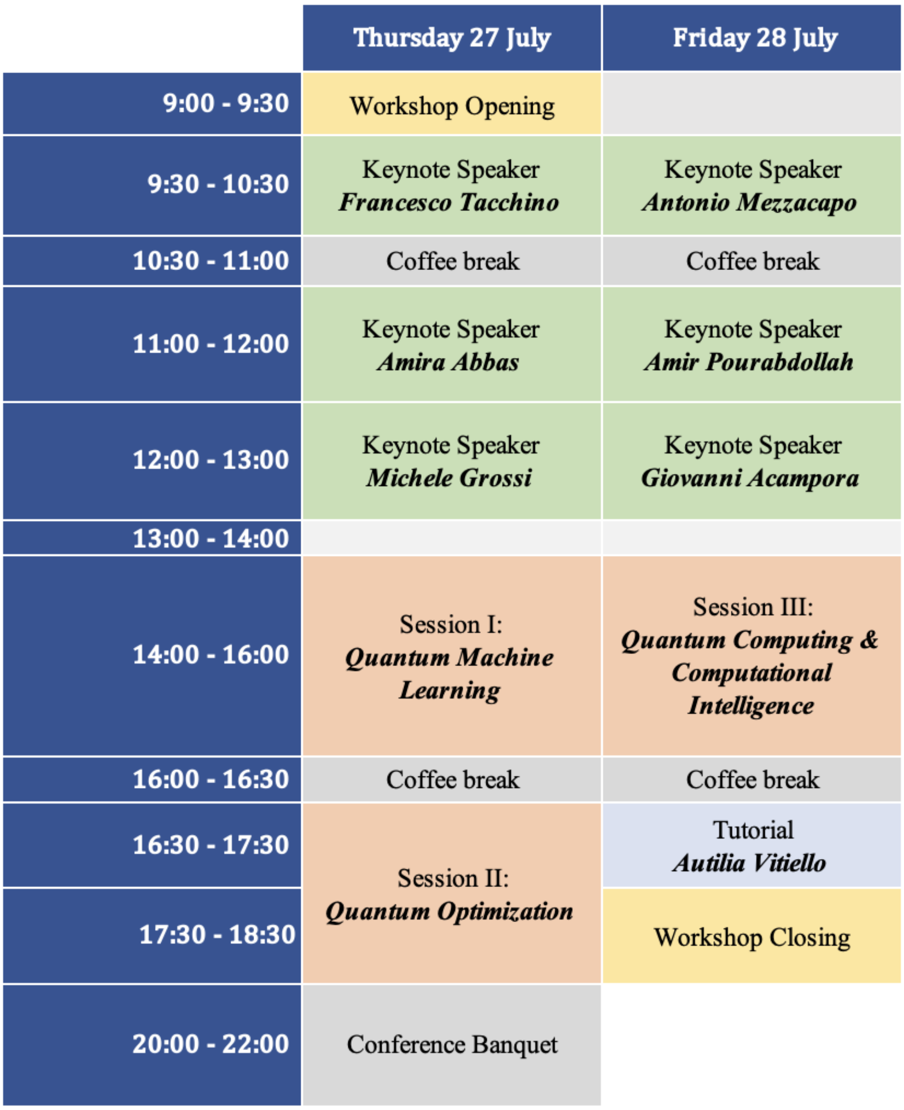
Program and Short Papers

Università degli Studi di Napoli
Federico II

Università degli Studi di Napoli Federico II - Dipartimento di
Fisica "Ettore Pancini"

QUASAR
Quantum Computing
and Smart Systems
Laboratory

QAI 2023 - Workshop on Quantum Artificial Intelligence



	Thursday 27 July	Friday 28 July
9:00 - 9:30	Workshop Opening	
9:30 - 10:30	Keynote Speaker ***Francesco Tacchino***	Keynote Speaker ***Antonio Mezzacapo***
10:30 - 11:00	Coffee break	Coffee break
11:00 - 12:00	Keynote Speaker ***Amira Abbas***	Keynote Speaker ***Amir Pourabdollah***
12:00 - 13:00	Keynote Speaker ***Michele Grossi***	Keynote Speaker ***Giovanni Acampora***
13:00 - 14:00		
14:00 - 16:00	Session I: ***Quantum Machine Learning***	Session III: ***Quantum Computing & Computational Intelligence***
16:00 - 16:30	Coffee break	Coffee break
16:30 - 17:30	Session II: ***Quantum Optimization***	Tutorial ***Autilia Vitiello***
17:30 - 18:30		Workshop Closing
20:00 - 22:00	Conference Banquet	

Contents

Presentation

Organizing Committee

General Chair
Giovanni Acampora
University of Naples Federico II
Italy

Local Arrangement Chair
Roberto Schiattarella
University of Naples Federico II
Italy

Local Arrangement Chair
Alfredo Massa
QuantumNet
Italy

General Chair
Autilia Vitiello
University of Naples Federico II
Italy

Local Arrangement Chair
Angela Chiatto
University of Naples Federico II
Italy

Publicity Chair
Zakaria Abdelmoiz DAHI
University of Malaga
Spain

Session I: Quantum Machine Learning

Using quantum supervised and unsupervised learning for binary classification in online setting

Corrado Loglisci
Department of Computer Science
University of Bari
Bari, Italy
corrado.loglisci@uniba.it

Donato Malerba
Department of Computer Science
University of Bari
Bari, Italy
donato.malerba@uniba.it

***Abstract*—Quantum machine learning recently gained prominence due to the promise of quantum computers in solving machine learning problems that are intractable on a classical computer. Nevertheless, several studies on problems which remain challenging for classical computing algorithms are emerging. One of these is classifying continuously incoming data instances in online fashion, which is studied in this paper through a hybrid computational solution that combines classical and quantum techniques. Hybrid approaches represents one of the current ways for the use of quantum computation in practical applications.**

In this paper, we show how typical issues of online learning can be equally addressed with the properties of quantum mechanics, until to offer often better results. We propose the combined use of quantum supervised algorithms (variational quantum circuits) and techniques of quantum unsupervised algorithms (distance estimation). We aim at keeping the classification capabilities, which have learned on previously processed data instances, preserved as much as possible, and then acquiring new knowledge on new data instances. Experiments are performed on real-world datasets with quantum simulators.

***Index Terms*—Quantum supervised learning; Quantum unsupervised learning; Hybrid quantum-classical framework; Online learning**

I. Introduction

Quantum machine learning has been introduced with the promise to handle machine learning problems that are intractable on a classical computer, especially those characterized by huge amounts of data. In the research on Quantum computing technologies, the current status sees the era of noisy intermediate scale quantum (NISQ) computers [1], which are devices able to deal with low-middle size data problems. An approach which seems bringing practical advantages is instead the one of *hybrid* frameworks [2] that combine classical and quantum methods and allow to exploit quantum physics properties while limiting the impact of the existing restrictions of the quantum devices.

One of the categories of data-intensive problems in which the research on classical computing dedicates still many efforts is learning of models from continuously incoming sequential data. Even the accurate solutions of Deep Learning find challenging working on that data scenario. This is demonstrated by the quite recentest studies to address the so-called *catastrophic forgetting* [3], which is the tendency of an artificial neural network to abruptly and drastically forget previously learned information upon learning new information. In those cases, it is not important designing algorithms for massive computation, but keeping the quality of the models high over unbounded sequences of data.

We investigate these points through a quantum-classical framework which build a classification model in the supervised setting and works in online learning [4] by acquiring continuously incoming data instances. The framework adapts continually a classification model and keeps on learning over time. More precisely, it trains and updates a classifier on (sub-) sequences of incoming data instances (data blocks) marked as labelled. Then, the classifier is used to estimate the class-value of unlabelled incoming data instances. The update is performed only when the properties and distribution of the labelled data changes. To detect such changes, we rely on a quantum centroid distance estimation technique, often used in quantum clustering. Indeed, the clusters gather the labelled data for each class that are therefore synthesized in the respective centroids. So, the changes are detected when the properties of the clusters change and this happens when the labelled data (assigned to the respective clusters) change. The classification model needs to be updated on those data.

The framework has been tested on the binary classification task by using two real-world datasets. The data size of these experiments is of the same order of the magnitude, or even higher, of the one used in the related works [5], [6]. It has been also compared against a classical computing algorithm working in online setting. The experimental results are encouraging and show the potential superiority in terms of accurate estimations over different experimental configurations.

II. Quantum-classical framework for binary classification

The overall framework (illustrated in Figure 1) relies on the classical computing techniques of feature selection, data sampling, normalization and model optimization. On the quantum side, it integrates quantum encoding techniques, quantum neural networks in the form of variational quantum circuits for the problem of binary classification and quantum distance estimation.The framework faces a binary classification problem, which can be formulated as follows. We have data

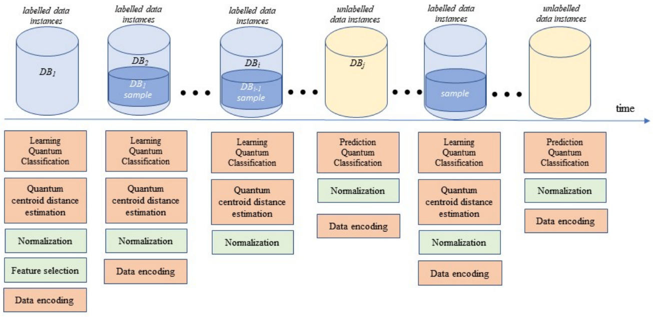

labelled data instances
labelled data instances
labelled data instances
unlabelled data instances
labelled data instances
unlabelled data instances
DB1
DB2
DB1 sample
DBi
DBi-1 sample
DBj
sample
time
Learning Quantum Classification
Quantum centroid distance estimation
Normalization
Feature selection
Data encoding
Learning Quantum Classification
Quantum centroid distance estimation
Normalization
Data encoding
Learning Quantum Classification
Quantum centroid distance estimation
Normalization
Prediction Quantum Classification
Normalization
Data encoding
Learning Quantum Classification
Quantum centroid distance estimation
Normalization
Data encoding
Prediction Quantum Classification
Normalization
Data encoding

Fig. 1. The components of the proposed hybrid framework as they run in online learning setting for binary classification.

instances described by $\mathbf{X} \cup y$, $\mathbf{X}$ are values of the set $\mathcal{X}$ of descriptive attributes/features, while $y \in \{-1, 1\}$ denotes the class label. The framework operates in online learning, which leads to alternate training sessions, where we have labelled data blocks, to prediction sessions, where the data instances have no class label. The succession of training sessions and prediction sessions is not predefined, coherently with the realistic assumption according to which the distribution of labelled and unlabelled data instances is not previously established and therefore not all the data instances are labelled.

In the following, we first provide a short description of each component and then describe how these work in the whole framework.

Feature selection operates only at the beginning and selects the subset of descriptive features which we will consider afterwards. It exploits a technique based on the mutual information between the class labels.

Normalization scales the values of the previously selected features within the range of [0,1] by using the standard min-max function on the original ranges. It is performed for each incoming data block, both those of training and those of prediction.

Data sampling selects a subset of the labelled data instances within the previously data block of training session. The samples will contain data instances of both the class labels and, for each class label, the component takes data instances with simple random techniques without replacement. The sample size is fixed.

Quantum Centroid Distance Estimation operates only on the labelled data instances and allows us to detect drift within data. Distance estimation is a standard operation in the blueprint of the distance-based clustering, and, in this work, is used specifically to build centroids. In this work, there is no clustering procedure as it is typically defined, but we use two centroids to synthesize the properties of the data instances of the two class labels and distance estimation to compute the distance between the centroids and the labelled data instances. More precisely, the centroids represent class prototypes and are used to identify the data instances which present concept (class) drift. The two centroids will be recomputed dependently on the drifts present in the incoming data instances. To implement these operations we resort to the design decisions proposed in [7] which implements the centroid distance estimation through SWAP gates, arranges centroids over quantum RAM structures and offers computations of the distances in superposition.

Quantum Classification works in two modalities, training and prediction. It is implemented through two quantum circuits with a number of qubits determined by the number features selected by Feature selection.

The first quantum circuit takes the classical data and represents them as quantum states to be assigned to the qubits. More precisely, this circuit implements a feature mapping operation $\mathcal{F}$ which encodes real-valued data instances $\mathbf{X}$ into quantum states spanning d qubits: $|\psi(\mathbf{X})\rangle = \mathcal{F}(\mathbf{X})|0\rangle^{\otimes d}$, where, $|0\rangle^{\otimes d}$ denotes the register with d-qubits at the state $|0\rangle$ ($|0\rangle \otimes \ldots \otimes |0\rangle$). In this work, $\mathcal{F}$ has been implemented as follows: $R_z{}^{\otimes d}(\mathbf{X}) H^{\otimes d} |0\rangle^{\otimes d}$, where, the parameter for each gate R_z is the normalized real-valued of of the feature (corresponding to the qubit on which R_z works). The term $H^{\otimes d}$ denotes the tensor product H $\otimes \ldots \otimes$H over d occurrences (that is, the number of selected features) of the gate H (the same holds for R_z).

The second circuit is variational and manipulates the quantum states returned by the first circuit. It implements a quantum neural network composed of layers of entangled rotation gates. Generally, entangled rotation gates are matrix operations which combine the gates Hadamard, CNOT and Rotation under the quantum physics effect of the entanglement [8]. The second circuit with the first completes the structure of gates which builds the classifier ϕ: $|\phi(\mathbf{X}, \theta)\rangle = \mathcal{V}(\theta)|\psi(\mathbf{X})\rangle$, where, $\mathcal{V}$ is the variational circuit, θ denotes the parameters of the parameterized gates that being optimized. In this work, $\mathcal{V}$ has been implemented as follows: $R_y{}^{\otimes d} CX^{\otimes d} R_y{}^{\otimes d}$, where, each occurrence of the two-qubit gate CX takes one pair of qubits (over the d-qubit register) composed by the consecutive qubits indexed as i and $i+1$.

Finally, we perform measurements on the qubits and the measured state is recorded. So, we can estimate the expectation value of the circuit on x and θ, by measuring the state over multiple runs, with the following $|\mathcal{E}(\mathbf{X}, \theta) = \langle\phi(\mathbf{X}, \theta)|\sigma_z^{\otimes d}|\phi(\mathbf{X}, \theta)\rangle$, where, $\sigma_z^{\otimes d}$ is the tensor product of the single qubit gate σ_z over d occurrences. The gate σ_z has the interesting property that if the measured quantum state has odd parity, it returns -1 (as eigenvalue), while, if the measured quantum state has even parity, it returns 1. This implies that the expectation value of the circuit will always be within the interval $[-1, 1]$. We can use this property to relate the expectation value to the probability that a data instance $\mathbf{X}$ being assigned to a class label y, that is: $P(y|\mathbf{X}) = \frac{y\mathcal{E}(\mathbf{X}, \theta)+1}{2}$.

The probability P(y|$\mathbf{X}$) is exploited in the optimization process concerning the parameters θ. In particular, the optimizer iteratively updates the circuit parameters by minimizing a cost function, which accounts for the negative log-likelihood of the probabilities P(y|$\mathbf{X}$) computed on the current labelled datablocks, that is: $-\frac{1}{size}\sum_{i=1}^{size} log(P(y_i|\mathbf{X}_i))$, where, $size$ is the number of data instances of the data block.

The cost function is minimized by a classical computing optimizer based on gradient descent. The derivative concerns the expectation value $\mathcal{E}()$ with respect to the current values of θ and is computed by means of the parameter shift rule [9]: $\frac{d\mathcal{E}}{d\theta} = \frac{\mathcal{E}(\theta+\epsilon)-\mathcal{E}(\theta_k-\epsilon)}{2}$

The gradient value is the difference between the two output values of the circuit: the first value is the output of the circuit with the parameter θ_k increased by a value ϵ, and the second value is the parameter θ_k decreased by ϵ.

Online learning. Learning classification models on continuously incoming data can be faced with time-windows models [10] in online setting. Time-windows models allows us to handle data instances by equally-sized blocks on which we train, update and apply the predictive capabilities of the classifier. During a training session, the training modality of Quantum classification is activated, which implies the execution the feature mapping $\mathcal{F}$ on the data instances of the current data block and optimization process of the parameters θ of the variational circuit $\mathcal{V}$. During a prediction session, the Quantum classification only estimates the class labels on the current data block by using the classification model up there updated.

To keep the classifier updated, we have to deal with the catastrophic forgetting effect raising when updating neural networks. In the literature, three alternatives are mainly suggested, *replay methods*, *regularization-based methods*, *parameter isolation methods* [11]. Considering that the replay methods represent the solution which asks for less and leaves unchanged the number of hyper-parameters of the neural network, we lean for this approach.

The framework operates in three steps, namely *initialization*, *update*, *prediction*. Training sessions are performed at the initialization and update. In the initialization step, the classifier is trained from scratch on the first data block DB_1 (Figure 1). The operation of *Feature selection* is used only at the initialization step, so the other steps of the framework work on the features before selected. Still at the initialization, two centroids, one for each class label, are determined from the labelled data instances of the data block DB_1. As new labelled data blocks will be acquired, the centroids will be recomputed. It should be clarified that the data represented used for quantum distance is amplitude encoding, which is different from the feature mapping used for the Quantum classification. The rationale behind is to use a different representational space in order to capture a different characteristics of the data from those expressed by the feature mapping of the Quantum classification.

Next, the framework prepares the steps of update and prediction by collecting labelled data instances and unlabelled data instances for two different data blocks. Afterwards, we will term them as DB_i and DB_j respectively. As soon as one of the two data blocks is being filled (the number of collected data instances is equal to the predefined size), the respective step is performed. By supposing the data block DB_i of labelled data instances is completed for first, the update step will be performed, otherwise it will be the turn of prediction step to work on DB_j. Clearly, the data block DB_j contains (unlabelled) data instances, in the order they arrive. When the update step starts, it first checks for possible concept drifts within the current data block, and, if any is present, it updates the classification model. To check the presence of drifts, we rely on classical computing technique, that is Page-Hinkley test [12], which, in this work, detects changes over time in the characteristics of the two clusters. As indicator of the characteristics of the clusters, we use sum of the squared errors $SSE : \sum_{instance_h \in DB_i} distance(c_k, instance_h)$ (DB_i current data block, c_k centroids), where the distance is the one introduced in Quantum Centroid Distance Estimation. This way, the Page-Hinkley test reveals the presence of drift if the SSE on the current data instances greatly differs from the one computed on the data instances previously processed. When this happens the two centroids are recomputed considering the new data instances and the classification model is updated. The training set used for the current learning round is composed of data instances of the current data block and those provided by the component of *Data sampling*. As explained above, this is done to mitigate the effect of catastrophic forgetting.

III. Experiments on real-world datasets

We implemented the proposed framework in IBM Qiskit [13] and run experiments by using simulators on two real-world datasets, more precisely *Ozone level detection* [1] (having 2536 data instances, 73 features) and *Spambase* (having 4600 data instances, 57 features) [2]. Data blocks have been partitioned so as having a portion of 75% of the dataset as labelled data instances (training sessions) and the remaining 25% as unlabelled data instances (prediction sessions and testing sets of the evaluation). The classical computing components described in Section II are those available in the toolkit Scikit-learn [14]. The number of runs of the classification model to estimate the expectation values is 1024, while the number of iterations (epochs) to optimize the parameters is 20. The number of layers for the variational quantum circuit is 3. The sample size of Data sampling is 30% the data-block size.

Preliminary experiments have been performed to emphasize the impact of the technical configuration of the framework on the accuracy, namely number of qubits (corresponding to the features selected) and size of the data blocks (number of data instances in each training/prediction session). In Table I, we report the accuracy of the proposed framework ($HYQOL$) compared to i) a classical computing solution (CC, originally designed for data stream learning) [15] and ii) a baseline of the framework that works on the whole dataset (FQC). The values illustrated have been computed as the average computed over the data blocks. As we can see, except two trials, HYQOL does not never worst than CC, even when the number of qubit is the higher (i.e., 8). Also, we note that the configurations of HYQOL with smallest set of qubits (i.e., 2) are better than

[1] https://archive.ics.uci.edu/ml/datasets/ozone+level+detection

[2] https://archive.ics.uci.edu/ml/datasets/Spambase

those with largest set (i.e., 8), without, however, particular discrepancy between the two endpoints. The size of the data blocks seems not be determinant for the accuracies, but, it is evident that online learning can be beneficial for quantum-based classifiers compared to the version that works on the whole dataset (FQC).

TABLE I
Accuracy (in [0,1] of the proposed framework against a classical computing solution and a hybrid solution without online learning. Dataset *seismic-bumps* at the top. Dataset *thyroid* at the bottom.)

data-block size		#qubits		
		2	5	8
50	HYQOL	0,9	0,9	0,9
	CC	0,9	0,9	0,9
100	HYQOL	0,93	0,89	0,86
	CC	0,87	0,87	0,87
200	HYQOL	0,94	0,94	0,91
	CC	0,92	0,92	0,92
400	HYQOL	0,92	0,91	0,9
	CC	0,9	0,9	0,9
	FQC	0,9	0,87	0,81

data-block size		#qubits		
		2	5	8
50	HYQOL	0,88	0,84	0,84
	CC	0,82	0,82	0,82
100	HYQOL	0,87	0,87	0,85
	CC	0,82	0,82	0,82
200	HYQOL	0,9	0,89	0,8
	CC	0,89	0,89	0,89
400	HYQOL	0,84	0,84	0,84
	CC	0,86	0,9	0,84
	FQC	0,83	0,83	0,8

IV. Conclusions

In this paper, we investigated the viability of quantum machine learning solutions to work on the realistic scenarios of changeability of the statistical properties of the data, which often implies the variability of the performances of the model. We conjecture this can be a machine learning problem in which the quantum solutions can lead innovation. On simulated hardware, the hybrid quantum-classical proposal offers encouraging results, in terms of accuracy, often better than a classical computing solution working on data stream and hybrid solution working in batch mode (no online learning). As our opinion, three take-home messages can be identified from this paper. The first one is methodological, in that the online learning opens to practical applications able to combine quantum computing and classical computing techniques, which is likely the only way to concretely use current quantum technologies. The second one is experimental, in that it provides arguments on the fact that stable quantum devices could even do better in terms of performances and quality of the results, when used in predictive tasks. The third one tell us that, although the high-performance computation and tractability of hard problems are the promises of quantum computing which, with the current devices, often are not kept, the research on the lifelong computation can be a field in which quantum computing can already bring interesting results.

References

[1] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-08-06-79

[2] A. Callison and N. Chancellor, "Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond," *Phys. Rev. A*, vol. 106, p. 010101, Jul 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.106.010101

[3] J. Peng, B. Tang, H. Jiang, Z. Li, Y. Lei, T. Lin, and H. Li, "Overcoming long-term catastrophic forgetting through adversarial neural pruning and synaptic consolidation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 9, pp. 4243–4256, 2022. [Online]. Available: https://doi.org/10.1109/TNNLS.2021.3056201

[4] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, 2021. [Online]. Available: https://doi.org/10.1016/j.neucom.2021.04.112

[5] D. Arthur and P. Date, "Hybrid quantum-classical neural networks," in *IEEE International Conference on Quantum Computing and Engineering, QCE 2022, Broomfield, CO, USA, September 18-23, 2022*, 2022, pp. 49–55.

[6] A. Chalumuri, R. Kune, and B. S. Manoj, "A hybrid classical-quantum approach for multi-class classification," *Quantum Inf. Process.*, vol. 20, no. 3, p. 119, 2021. [Online]. Available: https://doi.org/10.1007/s11128-021-03029-9

[7] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "q-means: A quantum algorithm for unsupervised machine learning," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 4136–4146.

[8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[9] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, "General parameter-shift rules for quantum gradients," *Quantum*, vol. 6, p. 677, Mar. 2022. [Online]. Available: https://doi.org/10.22331/q-2022-03-30-677

[10] J. Gama and M. G. (Eds), *Learning from Data Streams – Processing techniques in Sensor Networks*. Springer, 2007.

[11] M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3057446

[12] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954. [Online]. Available: http://www.jstor.org/stable/2333009

[13] M. S. Anis and et al, "Qiskit: An open-source framework for quantum computing," 2021.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018. [Online]. Available: http://jmlr.org/papers/v19/18-251.html

Implementation of Markov Decision Processes into quantum algorithms for reinforcement learning

1st M.P. Cuéllar
Dept. Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
manupc@decsai.ugr.es

2nd M.C. Pegalajar
Dept. Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
mcarmen@decsai.ugr.es

3rd L.G.B. Ruiz
Dept. Software Engineering
University of Granada
Granada, Spain
bacaruiz@ugr.es

4th G. Navarro
Dept. Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
gnavarro@ugr.es

5th C. Cano
Dept. Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
ccano@decsai.ugr.es

6th L. Servadei
School of Computation, Information and Technology
Technical University of Munich
Munich, Germany
lorenzo.servadei@tum.de

***Abstract*—In this work, we propose a methodology to implement classic Markov Decision Processes in a Quantum Computing paradigm, as a first step to achieve systems running Quantum Reinforcement Learning where both agent and environment are expressed as quantum programs. To do so, we analyze the interaction cycle between the agent and the environment in classic reinforcement learning and create a method to map a Markov Decision Process with discrete state space, action set, and rewards, into a quantum program.**

***Index Terms*—Reinforcement Learning, Quantum Computing, Quantum Machine Learning, Markov Decision Process**

I. Introduction

Classic Reinforcement Learning (RL) [1] is a type of Machine Learning where an intelligent agent learns by interaction with an unknown environment. A cycle of a RL agent/environment interaction (see Figure 1) is as follows: At each time step t the environment is in a state $s(t)$ which is fully observable by the agent. Then, the agent selects an action $a(t)$ over an available action set and performs the action over the environment, which evolves from state $s(t)$ to $s(t+1)$ and returns a reward scalar value $r(t+1)$ to the agent as a feedback of its performance. The goal in RL is to learn a policy $\pi(a|s)$ to select the best action a at every state s that maximizes the long term accumulated reward, i.e. $\sum_t \gamma r(t)$, where $\gamma \in [0,1]$ is a *discount factor*. Applications of RL are varied and some are described in [1].

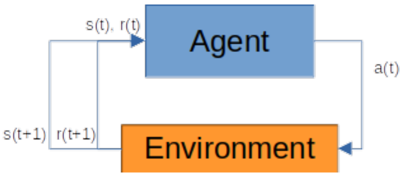


Fig. 1. Agent/environment interaction in a Reinforcement Learning cycle.

Classic RL is built upon the first-order Markov assumption, so that the environment can be modelled with a Markov Decision Process (MDP) [2]. An MDP is defined as a tuple $\langle S, A, P, R\rangle$, where $S = \{s_1, s_2, ..., s_{|S|}\}$ is a set of states, $A = \{a_1, a_2, ..., a_{|A|}\}$ is a set of actions, $P : S \times A \to \mathcal{P}(S)$ is a probabilistic transition function where $P(s_i, a_k, s_j)$ is the probability to evolve from state s_i to state s_j after executing action a_k, and $R : S \times A \times S \to \mathbb{R}$ is a reward function where $R(s_i, a_k, s_j)$ is a scalar value containing the reward obtained after reaching state s_j by means of executing action a_k in state s_i.

Quantum Reinforcement Learning (QRL) [3] attempts to adapt classic Reinforcement Learning methods, or to develop new techniques, for Quantum Computing (QC). As in Quantum Machine Learning [4], four different scenarios combining quantum and classical cases are studied, that arise from the possibilities of a combination of classical/quantum agents operating in classical/quantum environments. Nowadays, the QRL literature has been mostly focused on the case where the agent is implemented as a quantum program (usually by

This work was supported by the project QUANERGY (Ref. TED2021-129360B-I00), Ecological and Digital Transition R&D projects call 2022, Government of Spain.

means of Variational Quantum Circuits, VQC) working over a classical environment as in [3], [5]–[8] to mention just a few, since real quantum environments are difficult to find. However, in the last few years, there have been a few proposals to create quantum environments such as the quantum Tic-Tac-Toe [9].

In this work, we propose a method to migrate a classic MDP to a QC paradigm, as a first step to create true quantum environment simulators that enable the execution of quantum agents in quantum environments. To do so, we first analyze the components of the underlying MDP that rules the environment evolution according to an agent action, and propose a methodology to create quantum programs that emulate the agent/environment interaction cycle.

This work is structured as follows: Section II describes the approach. After that, Section III show an example of the method as a proof of concept, and Section IV concludes.

II. Implementation of the MDP cycle in Quantum Computing scenarios

Our starting hypothesis is that all the state space S, action space A, and possible rewards in R, can be expressed as discrete sets. In this context, our goal is to find a deterministic policy $\pi(s)$ that returns the best action $a = \pi(s)$ for each environment state. The proposed general method to implement the RL cycle in QC, containing the underlying MDP, can be devised as follows (Figure 2):

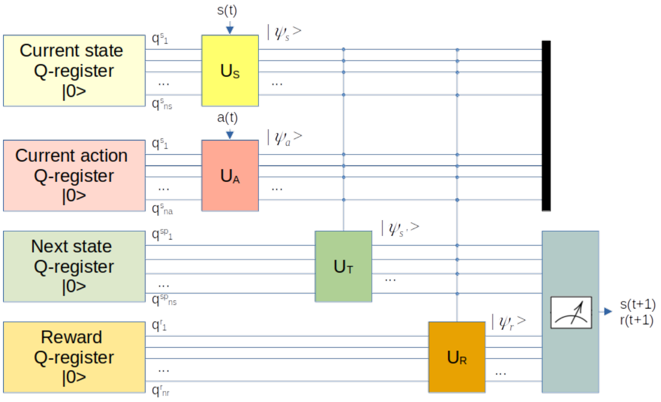


Fig. 2. Scheme of the implementation of an MDP with a quantum program.

- At time t, the environment is at a given state $s(t) = s_i \in S$, provided as an input. Therefore, a quantum state preparation method is required to encode the cycle's initial state s_i into its corresponding quantum state $|\psi_s\rangle$ in a quantum register with n_s qubits. The operator to create $|\psi_s\rangle$ from $|0\rangle$ is U_S, i.e. $|\psi_s\rangle = U_S\,|0\rangle$.
- A deterministic action $a(t) = a_k \in A$, selected by either a classical or quantum agent, is provided as input and applied over the environment. The action a_k is encoded into the state of a quantum register with n_a qubits $|\psi_a\rangle$ using the unitary transformation U_A so that $|\psi_a\rangle\,|\psi_s\rangle = U_A\,|0\rangle\,|\psi_s\rangle$.
- Once the input state s_i and action a_k are encoded into quantum states $|\psi_s\rangle, |\psi_a\rangle$, the transition function $P(s_i, a_k)$ is executed to evolve the environment from the known state $s(t) = s_i$ to a new state $s(t+1) = s_j$ with probability $P(s_i, a_k, s_j)$. To do so, a target quantum register to store the next state, $|\psi_{s'}\rangle$, is required. A controlled unitary transformation U_T is in charge of calculating the superposition of target environment states as $|\psi_{s'}\rangle$, so that $|\psi_{s'}\rangle\,|\psi_a\rangle\,|\psi_s\rangle = U_T\,|0\rangle\,|\psi_a\rangle\,|\psi_s\rangle$.
- The reward $r(t+1) = R(s_i, a_k, s_j)$ is finally calculated. We could think of the reward function as a controlled quantum operator U_R that evolves a quantum register $|0\rangle$ of n_r qubits allocated for rewards to a quantum state $|\psi_r\rangle$ encoding $r(t+1)$. Formally, it is written as $|\psi_r\rangle\,|\psi_{s'}\rangle\,|\psi_a\rangle\,|\psi_s\rangle = U_R\,|0\rangle\,|\psi_{s'}\rangle\,|\psi_a\rangle\,|\psi_s\rangle$.
- Return of $s(t+1), r(t+1)$ to the agent. In the case the agent is classical, this can be performed with measurement operators over the target state quantum register $|\psi_{s'}\rangle$ and reward register $|\psi_r\rangle$.

A. Current state and action encoding

To achieve an optimal use of qubit resources using a dense encoding technique, all environment states, actions and rewards are stored in the amplitudes of their corresponding quantum registers. This means that the size of n_s qubits used for representing the environment states, n_a qubits to represent the available actions, and n_r qubits to represent rewards can hold $2^{n_s}, 2^{n_a}$ and 2^{n_r} different possible values, respectively.

Since environment states are fully observable in a MDP, the quantum state representing $|\psi_s\rangle$ contains an amplitude whose squared value equals 1.0, and therefore a simple basis encoding technique [4] containing X gates can be used as the encoding operator U_S. The same situation occurs for actions and U_A. Assuming a mapping from environment states, $s_i \in S$ to basis quantum states, i.e. $s_i \mapsto |i\rangle$, and also for actions $a_k \mapsto |k\rangle$, the unitary operators U_S and U_A can be generalized and implemented using parameterized circuits containing $R_x(\theta)$ gates, where $\theta \in \{0, \pi\}$ depending on the binary representation of s_i, a_k.

B. Implementation details of transition probabilities

The unitary transformation U_T is in charge of calculating the superposition of target states $s_j \in S$ into $|\psi_{s'}\rangle$, so that the register of the next state $|\psi_{s'}\rangle$ contains the probabilities of measurement $P(s_i, a_k, s_j)$. Thus, U_T can be implemented as a sequence of instances of the Q-Sample encoding method [4] controlled by the values of the quantum registers $|\psi_a\rangle\,|\psi_s\rangle$. Section III provides an example of such instances.

C. Implementation details of reward computation

Similarly to states and actions, we assume a mapping from the set of rewards $r_l \in R$ to basis states as $r_l \mapsto |l\rangle$ in the reward quantum register. After all values s_i, a_k, s_j are known, the reward function $R(s_i, a_k, s_j)$ is implemented into U_R as a controlled operation that sets a probability amplitude of a basis vector in the reward quantum register to value 1.0. This is implemented in a circuit as a sequence of multiple controlled NOT gates that set the correct amplitude. The idea behind this operation is to entangle the reward quantum state with $|\psi_{s'}\rangle |\psi_a\rangle |\psi_s\rangle$ so that the correct reward value is obtained once $|\psi_{s'}\rangle$ collapses after measurement.

III. Proof of Concept and Implementation

Due to space limitations, we limit the experimentation in this work to show the RL cycle mapping from the MDP in Figure 3 to a quantum program. We also performed a toy example experiment to train a classic agent using the Q-Learning procedure with $\gamma = 0.99$ in a maximum number of $T = 200$ steps and learning rate $\alpha = 0.2$ to test convergence. The source code for this experimentation is freely available at *https://github.com/manupc/MDPQuantum*.

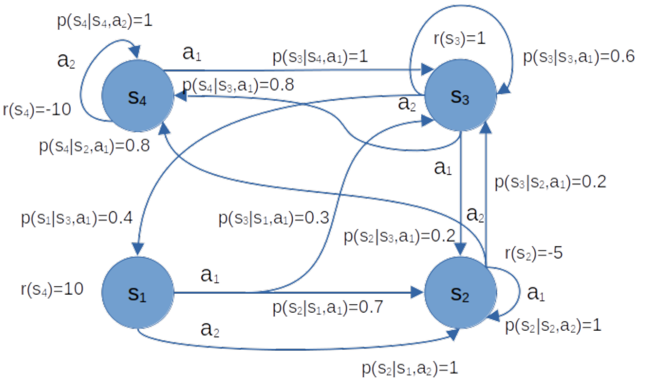


Fig. 3. Example MDP.

The MDP in Figure 3 contains four states $S = \{s_1, s_2, s_3, s_4\}$, two possible actions for each state $A = \{a_1, a_2\}$, and a discrete set with four rewards that depend on the final transition state only $\{r(s_1) = 10, r(s_2) = -5, r(s_3) = 1, r(s_4) = -10\}$. Therefore, the number of qubits required to represent the states and rewards is $n_s = n_r = 2$. The proposed mapping from states and rewards to basis vectors is $s_1 \mapsto |00\rangle, s_2 \mapsto |01\rangle, s_3 \mapsto |10\rangle, s_4 \mapsto |11\rangle$ and $r(\cdot,\cdot,s_1) \mapsto |00\rangle, r(\cdot,\cdot,s_2) \mapsto |01\rangle, r(\cdot,\cdot,s_3) \mapsto |10\rangle, r(\cdot,\cdot,s_4) \mapsto |11\rangle$, respectively. It is assumed that the initial state to execute the environment is s_1. With respect to actions, a number of $n_a = 1$ qubit is required under the mapping $a_1 \mapsto |0\rangle, a_2 \mapsto |1\rangle$.

The optimal deterministic policy was obtained using the Value Iteration method in classic RL, obtaining the policy $\pi(s_1) = a_1, \pi(s_2) = a_2, \pi(s_3) = a_2, \pi(s_4) = a_1$. Running the value iteration over the classical environment required 0.02 sec. on a desktop computer Intel(R) Core(TM) i5-9600K CPU at 3.70GHz with 32GB RAM. On the other hand, the Q-Learning algorithm over the classical environment took 0.012 sec. and obtained the optimal policy.

The quantum MDP was implemented in IBM's Qiskit with the QASM noise-free simulator. An example of a sub-circuit containing the transition from s_3 and action a_2 to states s_0, s_3 with probabilities 0.4 and 0.6, respectively, is depicted in Figure 4. The Q-Learning method was applied to learn the optimal policy, using a computational time of 226.68 sec. under ideal conditions.

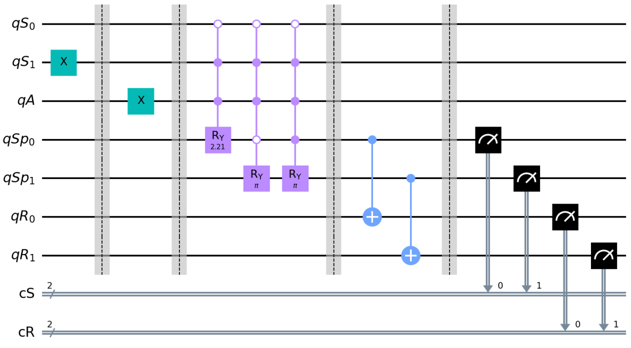


Fig. 4. Example circuit to represent a single transition $P(s_2, a_2)$.

IV. Conclusions

In this work, we have shown a preliminary study towards the creation of quantum environments for reinforcement learning, inspired in the hypotheses of classic RL using MDPs as a first step. We have developed a method to implement a reinforcement learning agent/environment interaction cycle in a quantum computer containing the full transition and reward functions of the MDP. Future works will consider extending the implementation to models with partial observability, and general methods to implement quantum agents working in quantum environments.

Acknowledgements

This article was supported by the project QUANERGY (Ref. TED2021-129360B-I00), Ecological and Digital Transition R&D projects call 2022 funded by MCIN/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR.

References

[1] P. Winder, *Reinforcement Learning: Industrial Applications with Intelligent Agents*. O'Reilly, 2020.

[2] M. Putterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[3] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1207–1220, 2008.

[4] A. Zeguendry, Z. Jarir, and M. Quafafou, "Quantum machine learning: A review and case studies," *Entropy*, vol. 25, no. 2, 2023. [Online]. Available: https://www.mdpi.com/1099-4300/25/2/287

[5] A. Skolik, S. Jerbi, and V. Dunjko, "Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning," *Quantum*, vol. 6, p. 720, 2022.

[6] S. Jerbi, C. Gyurik, S. C. Marshall, H. J. Briegel, and V. Dunjko, "Parametrized quantum policies for reinforcement learning," 2021.

[7] N. D. Pozza, L. Buffoni, S. Martina, and F. Caruso, "Quantum reinforcement learning: the maze problem," *Quantum Mach. Intell.*, vol. 4, no. 2, pp. 1–10, 2022. [Online]. Available: https://doi.org/10.1007/s42484-022-00068-y

[8] E. Andrés, M. P. Cuéllar, and G. Navarro, "On the use of quantum reinforcement learning in energy-efficiency scenarios," *Energies*, vol. 15, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/16/6034

[9] S. Sagole, A. Dey, B. Behera, and P. Panigrahi, "Quantum tic-tac-toe: A hybrid of quantum and classical computing," Tech. Rep., 12 2019.

Quantum Based Outlier Detection

Anna Bernasconi
Dept. of Computer Science
University of Pisa, Italy
0000-0003-0263-5221

Alessandro Berti
Dept. of Computer Science
University of Pisa, Italy
0000-0001-9144-9572

Gianna M. Del Corso
Dept. of Computer Science
University of Pisa, Italy
0000-0002-5651-9368

Alessandro Poggiali
Dept. of Computer Science
University of Pisa, Italy
0000-0002-1591-7925

***Abstract*—The growing interest in quantum computation is driven by its potential to surpass classical computation speed. This interest has led to an increasing focus on developing new quantum algorithms for data analysis tasks, including outlier detection, which is crucial in identifying dataset abnormalities. Some quantum outlier detection techniques have been proposed in the past, demonstrating the potential of quantum algorithms to enhance outlier detection. This study introduces a novel Quantum Outlier Detection Algorithm (QODA) that leverages variance estimation. We evaluate the performance of QODA on multiple datasets, and our results demonstrate that it enhances the efficiency of classical outlier detection while maintaining high accuracy.**

***Index Terms*—Quantum Computing, Quantum Artificial Intelligence, Quantum Outlier Detection**

I. Introduction

Detecting outliers is crucial in many fields, including finance, healthcare, and engineering. Outliers can significantly impact data analysis results, leading to incorrect conclusions or decisions. Hence, developing effective outlier detection algorithms is critical. Recently, quantum computing has shown great potential to solve computationally expensive problems for classical computers, including data analysis tasks. In the literature, various techniques for detecting outliers in quantum computing have been suggested [1], [3]. However, our paper introduces a novel approach, a hybrid quantum algorithm inspired by the classical Angle-Based Outlier Detection (ABOD) algorithm [2], an unsupervised distance-based method for detecting outliers. Our proposed hybrid quantum version of ABOD leverages the power of quantum computing to enhance the algorithm's performance.

We begin by providing an overview of the ABOD algorithm and of the notation used in the paper. Next, we propose a hybrid algorithm inspired by ABOD, which leverages a quantum subroutine to enhance the efficiency of outlier detection. Finally, we evaluate the performance of our algorithm on some synthetic and benchmark datasets and compare it with the classical technique. The experiments indicate that our hybrid quantum algorithm yields results comparable to those of the classical ABOD algorithm. Overall, our algorithm offers a promising approach for outlier detection tasks, demonstrating the potential of quantum computing to enhance data analysis in various fields.

II. Outlier Detection

A. Classical Approach

ABOD (Angle-Based Outlier Detection) is a distance-based unsupervised outlier detection method that measures the abnormality of each data point by computing the variance of the angles between the difference vectors of the given point to the other points in the dataset. Indeed, the variance of these angles depends on how different a point is from the others: for points within a cluster, the angles differ widely, while outliers are expected to present a smaller variance because they are further away from the rest of the dataset. Hence, the points with smaller ABOD values are considered to be more likely to be outliers. ABOD has several advantages over other outlier detection techniques, including detecting outliers in high-dimensional datasets. However, it may not perform well on datasets with a high degree of overlap between classes or when the dataset contains clusters of outliers. Moreover, it has a cost of $O(M^3N)$ where M is the number of records, and N is the dimension of each record; hence the cost is very high.

The classical algorithm works as follows. Assume we have M records, each with N-dimensions, stored in a matrix $X = [x_1|x_2|\cdots|x_M] \in \mathbb{R}^{N\times M}$, so that each record can be seen as belonging to a vector space of dimension N. We iterate over all records, considering each one as a pivot. Thus, let x_p, $1 \leq p \leq M$, the current pivot. We denote by $\theta_{ij}^{(p)}$ the angle between the vectors x_i and x_j observed from the pivot x_p, i.e., the angle between the difference vectors $\tilde{x}_i = x_i - x_p$ and $\tilde{x}_j = x_j - x_p$. Mathematically this can be written as:

$$\theta_{ij}^{(p)} = \arccos\left(\frac{\tilde{x}_i^T\tilde{x}_j}{\|\tilde{x}_i\|\|\tilde{x}_j\|}\right),$$

where all norms are 2-norms. For each $p \in \{1,\ldots,M\}$, we compute the variance v_p of $\theta_{ij}^{(p)}$, with $1 \leq i < j \leq M$ and $i,j \neq p$, i.e., $v_p = \text{Var}\left(\theta_{ij}^{(p)}\right) = \mathbb{E}[(\theta_{ij}^{(p)})^2] - \mathbb{E}^2[\theta_{ij}^{(p)}]$, where we denoted by $E[\cdot]$ the expected value. Once the vector $V = [v_1, v_2, \ldots, v_M]$ is ready, the algorithm detects the outliers as the records with variance below a given threshold. The algorithm is usually implemented by computing the variance directly on the inner products, i.e., on the cosine of $\theta_{ij}^{(p)}$ [2].

B. Quantum Outlier Detection

In this section, we design a hybrid quantum version of ABOD called Quantum Outlier Detection Algorithm (QODA).

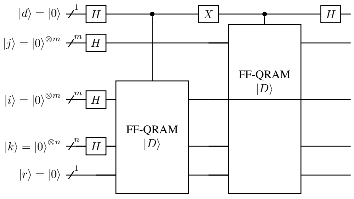

$|d\rangle = |0\rangle$
$|j\rangle = |0\rangle^{\otimes m}$
$|i\rangle = |0\rangle^{\otimes m}$
$|k\rangle = |0\rangle^{\otimes n}$
$|r\rangle = |0\rangle$
1
m
m
n
1
H
X
FF-QRAM
$|D\rangle$
FF-QRAM
$|D\rangle$

Fig. 1. Computation of differences.

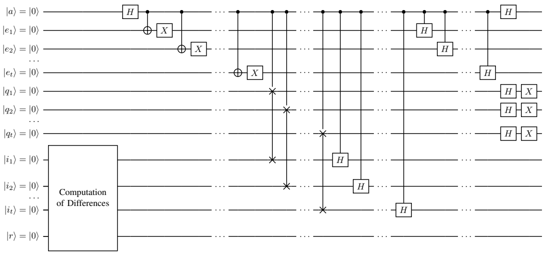

$|a\rangle = |0\rangle$
$|e_1\rangle = |0\rangle$
$|e_2\rangle = |0\rangle$
$|e_t\rangle = |0\rangle$
$|q_1\rangle = |0\rangle$
$|q_2\rangle = |0\rangle$
$|q_t\rangle = |0\rangle$
$|i_1\rangle = |0\rangle$
$|i_2\rangle = |0\rangle$
$|i_t\rangle = |0\rangle$
$|r\rangle = |0\rangle$
Computation of Differences
H
X

Fig. 2. Quantum Outlier Factor Oracle

QODA iterates over all records in a given dataset and calls a quantum subroutine to test whether the current record is an outlier on a single quantum circuit. Alternatively, we could test each record in the dataset by employing distinct quantum circuits that can run in parallel. QODA mimics the classical ADOB approach, exploiting the variance as a statistical measure to detect outliers, to reduce the computational complexity of the classical version thanks to quantum phenomena. A crucial subroutine of QODA is, therefore, an efficient quantum heuristic for estimating the variance of all angles $\theta_{ij}^{(p)}$, for a given sample x_p, taken as a pivot. We call this subroutine *Quantum Outlier Factor* (QOF). This heuristic is based on the idea of approximating the angles $\theta_{ij}^{(p)}$ with the differences $\Delta_{ij}^{(p)}$ between the vectors translated by the pivot x_p and normalized. The variance is then computed directly over the approximations $\Delta_{ij}^{(p)}$ within the same quantum circuit. More precisely, let x_p be the pivot. For all $1 \leq i \leq M$, we (classically) compute $\tilde{x}_i = x_i - x_p$. Then, we normalize all vectors adopting as default data preprocessing the Inverse Stereographic Projection (ISP), in order to keep the clusters separated, as discussed in [5]. Recall that the ISP maps N-dimensional data into the surface of a unit sphere in the $(N+1)$-dimensional space and is computed as follows. Given a vector $v = (v_1, v_2, \dots, v_N) \in \mathbb{R}^n$, then:

$$\mathrm{ISP}(v) = \left(\frac{2v_1}{\|v\|^2+1}, \frac{2v_2}{\|v\|^2+1}, \dots, \frac{2v_N}{\|v\|^2+1}, \frac{\|v\|^2-1}{\|v\|^2+1}\right),$$

where all norms are 2-norms. Notice that $\|\mathrm{ISP}(v)\| = 1$. Thus, let $\hat{x}_i = \mathrm{ISP}(\tilde{x}_i) = \mathrm{ISP}(x_i - x_p)$ denote the normalized vectors, translated by x_p. We encode all these vectors in a quantum circuit and then use a Hadamard gate to compute the differences $\Delta_{ij}^{(p)} = \frac{1}{2(M-1)\sqrt{N+1}}(\hat{x}_i - \hat{x}_j)$ in the amplitudes, appropriately normalized, as shown in Figure 1. Note that the dependence from the chosen pivot x_p is kept within $\hat{x}_i$ and $\hat{x}_j$, indeed we have

$$(\hat{x}_i - \hat{x}_j)_k = \begin{cases} \frac{2(x_{ki}-x_{kp})}{\|x_i-x_p\|^2+1} - \frac{2(x_{kj}-x_{kp})}{\|x_j-x_p\|^2+1} & \text{for } k = 1, \dots, N \\ \frac{2(\|x_i-x_p\|^2-\|x_j-x_p\|^2)}{(\|x_i-x_p\|^2+1)(\|x_j-x_p\|^2+1)}, & \text{for } k = N+1. \end{cases}.$$

These differences are then used, instead of the angles $\theta_{ij}^{(p)}$, for the variance computation in the same circuit, as shown in Figure 2. Indeed, the variance of the differences provides an approximated lower bound for the variance of the angles, as stated in the following theorem.

Theorem 1. *Let $\Delta^{(p)}$ be the $(M-1)^2(N+1)$-dimensional vector obtained appending all the $\Delta_{ij}^{(p)}$, for $1 \leq i, j \leq M, (i, j \neq p)$, and $\theta^{(p)}$ be the $(M-1)^2$-dimensional vector whose components are the angles $\hat{\theta}_{ij}^{(p)}$ between $\hat{x}_i$ and $\hat{x}_j$, for $1 \leq i, j \leq M, (i, j \neq p)$. Then:*

$$\mathit{Var}(\Delta^{(p)}) \;\dot{<}\; \frac{1}{N+1}\mathit{Var}(\hat{\theta}^{(p)}) - \frac{1}{N}\mathbb{E}^2[\hat{\theta}^{(p)}]\,. \tag{1}$$

Proof. (Sketch) Follows applying the second order Taylor expansion of the cosine function to approximate the scalar product between the vectors $\hat{x}_i$ and $\hat{x}_j$.

Using vector differences instead of angles is convenient as we can maintain the superposition of all differences in the QOF oracle and perform the variance computation directly on them within the same circuit (see Figure 2). Thus, this approach requires only one data encoding step and one final measurement. An alternative oracle could be derived by computing the inner products between all pairs of vectors with a technique inspired by [4], that does not require an intermediate measurement as the standard method based on fidelity computation. This alternative approach is currently under experimental evaluation.

The quantum subroutine for computing the QOF employs the Amplitude Estimation (AE) algorithm, using as oracle the one of Figure 2. In this way, we can estimate the amplitude of the target configuration containing the variance of the differences by measuring the m additional qubits of AE. The measurement output represents an approximation of the variance in the computational basis.

The complexity of the quantum oracle QOF is $O(\log M^2N)$, i.e., $O(\log MN)$, assuming available efficient methods for encoding classical data. The overall hybrid algorithm QODA to solve the outlier detection problem is summarized by the pseudocode in Algorithm 1. The complexity of QODA is $O(M(MN+\log MN)) = O(M^2N)$,

Algorithm 1 QODA

Input: X - input data, t - threshold
Output: O - list of outliers

```
$O \leftarrow []$                                   // list of outliers initially empty
for $p \in \{1, 2, \dots, M\}$ do                     // for each record
    $\tilde{X} \leftarrow X - |x_p\rangle\langle\mathbf{1}|$ ;              // scale the dataset w.r.t. $x_p$
    $\hat{X} \leftarrow ISP(\tilde{X})$ ;                      // normalize $X_p$ with ISP
    $v_p \leftarrow QOF(\hat{X})$                       // Quantum Outlier Factor of $x_p$.
    if $v_p \leq t$ then                          // check if $x_p$ is an outlier
        $O.append(x_p)$ ;                  // append $x_p$ to the list of outliers
return $O$ ;                                 // return the list of outliers
```

where the term MN is due to the data preprocessing. Due to the space limitation, all details, together with a formal proof of the correctness of the quantum oracle QOF, will be given in the extended version of the paper.

III. Experiments

The following section presents the numerical experiments conducted to evaluate the effectiveness of the Quantum Outlier Detection Algorithm (QODA) on a series of benchmark datasets. The initial assessment involves testing the effectiveness of the heuristic used by QODA. The second experimental evaluation aims at assessing the accuracy of QODA using the QASM SIMULATOR of Qiskit.

A. Heuristic Evaluation

The following datasets are used as benchmarks to evaluate the heuristic:

- synth3: M=500 synthetic data with dimension N=20 whose 2% are outliers.
- synth4: M=500 synthetic data with dimension N=30 whose 2% are outliers.
- synth1: M=500 synthetic data with dimension N=20 whose 10% are outliers.
- synth2: M=500 synthetic data with dimension N=30 whose 10% are outliers.
- lympho: real dataset with M=148, N=18.
- glass: real dataset with M=214, N=9.
- wbc: real dataset with M=278, N=30.

Our experiments can be divided into two types. The first type is conducted to verify the accuracy of the lower bound given by Theorem 1. In Table I, we present the approximation errors, measured in terms of Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), introduced by the relation (1). Our findings indicate that the lower bound (1) holds for all datasets considered.

TABLE I
Approximation errors of relation (1)

dataset	**MSE**	**MAE**	**RMSE**
synth1	2.53e-8	1.30e-4	1.60e-4
synth2	1.57e-9	3.37e-5	3.96e-5
synth3	3.02e-8	1.45e-4	1.73e-4
synth4	1.92e-9	3.85e-5	4.38e-5
lympho	3.83e-7	5.27e-4	6.19e-4
glass	7.51-3	5.98e-2	8.66e-2
wbc	3.09e-5	5.21e-3	5.55e-3

The second type of experiment aims to evaluate the effectiveness of outlier detection leveraging the variance of the differences $\Delta_{ij}^{(p)}$ instead of the variance of the angles $\theta_{ij}^{(p)}$. We compare the outlier rankings generated by two methods to achieve this objective. Table II presents the precision-at-n (P@n) metric, which we used to conduct this comparison. Specifically, we calculated P@n by determining the number of records in the top n ranks identified as outliers using the differences and angles, respectively. We can see that accuracy is very high, especially for synthetic datasets. Furthermore, we utilized the Rank Biased Overlap (RBO) [6] measure to evaluate the overall outlier ranking of our heuristic, taking as ground truth the classical algorithm working with the angles. This metric considers the ranking similarity on the top ranks, with the weighting determined by a parameter p. A higher value of p assigns greater importance to the top ranks in the final similarity calculation. The table indicates that the outlier rankings obtained by the method leveraging the variance of differences are similar to the ground truth, particularly for the highest-ranked outliers.

TABLE II
Precision-at-n (P@n) of the algorithm taking as ground truth the classical algorithm working with the angles.

dataset	**n=5**	**n=10**	**n=15**	**n=20**	**n=25**	**n=30**
synth1	0.80	0.90	0.93	0.90	0.96	0.97
synth2	1.0	1.0	1.0	0.95	0.96	0.93
synth3	1.0	1.0	1.0	0.90	0.84	0.93
synth4	1.0	1.0	0.93	0.95	0.84	0.90
lympho	0.80	0.60	0.67	0.70	0.76	0.77
glass	0.70	0.80	0.80	0.90	0.84	0.87
wbc	0.60	0.70	0.80	0.70	0.60	0.57

TABLE III
RBO measure varying the parameter p

dataset	**p=0.70**	**p=0.75**	**p=0.80**	**p=0.85**	**p=0.90**	**p=0.95**
synth1	0.65	0.69	0.74	0.78	0.83	0.88
synth2	0.67	0.71	0.76	0.81	0.86	0.91
synth3	0.88	0.89	0.89	0.90	0.90	0.91
synth4	0.65	0.69	0.74	0.79	0.84	0.88
lympho	0.52	0.54	0.57	0.59	0.62	0.66
glass	0.51	0.55	0.59	0.63	0.68	0.74
wbc	0.60	0.63	0.66	0.68	0.68	0.67

B. Quantum Simulation

Currently, the availability of quantum hardware is limited, which means that quantum computation can only be simulated using classical hardware. Nevertheless, simulating quantum computation on classical hardware is expensive, so we can only execute basic versions of our algorithm. For this reason, we run QODA on two simple synthetic datasets with a single outlier each. The first has $M = 5$ records with dimension $N = 1$, and the second has $M = 5$ records with dimension $N = 3$. By running the experiments with different random seed values for generating datasets five times, we found that QODA successfully identifies the sole outlier in both datasets with a 100% accuracy rate.

References

[1] M. Guo, H. Liu, Y. Li, W. Li, F. Gao, S. Qin, and Q. Wen, "Quantum algorithms for anomaly detection using amplitude estimation," *Physica A: Statistical Mechanics and its Applications*, vol. 604, 2022.

[2] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 444–452.

[3] J.-M. Liang, S.-Q. Shen, M. Li, and L. Li, "Quantum anomaly detection with density estimation and multivariate gaussian distribution," *Physical Review A*, vol. 99, no. 5, 2019.

[4] V. Markov, C. Stefanski, A. Rao, and C. Gonciulea, "A generalized quantum inner product and applications to financial engineering," 2022.

[5] A. Poggiali, A. Berti, A. Bernasconi, G. M. D. Corso, and R. Guidotti, "Quantum clustering with k-means: a hybrid approach," 2022.

[6] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 4, pp. 1–38, 2010.

Parametrized quantum circuits for anomaly detection and generative tasks

1st Simone Bordoni
Sapienza Università di Roma
Technology Innovation Institute (Abu Dhabi)
Rome, Italy
simone.bordoni@uniroma1.it

2nd Andrea Cacioppo
Sapienza Università di Roma
Rome, Italy
andrea.cacioppo@uniroma1.it

3rd Lorenzo Colantonio
Sapienza Università di Roma
Rome, Italy
colantonio.1796146@studenti.uniroma1.it

4th Stefano Giagu
Sapienza Università di Roma and INFN
Rome, Italy
stefano.giagu@uniroma1.it

***Abstract*—We investigate the possibility to apply parametrized quantum circuits, in particular quantum autoencoders, for different machine learning tasks. The first application is for anomaly detection in handwritten digits as well as more complex structures like anomalous patterns in the particle detectors. This algorithm has been trained on a classical computer and tested with simulations and on real quantum hardware. Tests on NISQ devices have been performed with IBM quantum computers. For the execution on quantum hardware specific hardware driven adaptations have been devised and implemented. We also present a preliminary study about the possibility of applying parametrized quantum circuits for generative tasks. In this study, the quantum circuit has been used in the denoising steps of a quantum diffusion model.**

***Index Terms*—Quantum machine learning, Quantum autoencoder, Parametrized quantum circuit, Anomaly detection, Diffusion model**

I. Introduction

In this work we propose two different applications of the quantum autoencoder (QAE) circuit ansatz for QML [3]. The first application is quantum anomaly detection. We test this algorithm on an easier task involving a standard benchmark dataset in machine learning, the handwritten digits MNIST dataset. We then apply the technique to a more complex and interesting use-case, the identification of anomalous signatures inside a particle detector due to the decay of long-lived particles. Some of the quantum circuits developed in this work are simple enough to be tested on Noisy Intermediate Scale Quantum (NISQ) computers [2]. The tests on real quantum hardware have been implemented on IBM quantum computers.The second application of the QAE is for a diffusion model. Classical diffusion models are showing great performances in generative tasks, and usually are more stable than the previous generation of generative algorithms. In a quantum diffusion model a QAE is applied to perform consecutive denoising steps. An algorithm of this kind would be able, in principle, to transform an initial random quantum state into a quantum state with defined characteristics. The possibility of generating quantum states similar to a defined set of samples may be an important resource in quantum computing [11].

II. Background on quantum autoencoders

Autoencoders are a class of machine learning algorithms that aim at compressing and reconstructing data. These algorithms con be implemented using two artificial neural networks: the encoder and the decoder. The encoder compresses initial data down to a small dimension (latent dimension). The decoder inverts the process to reconstruct the original data from the compressed representation. The parameters of the neural network are trained in order to minimize the difference between the initial and reconstructed data. A quantum autoencoder keeps the same structure of the classical counterpart but replaces the artificial neural networks with parametrized quantum circuits (PQC) [5]. A PQC is a quantum circuit that depends on free (trainable) parameters. Information is stored in the state of the qubits, in this work we use the state amplitudes (amplitude encoding) [10]. The initial state is transformed using rotation gates and entangling gates. These gates can be organized in layers; in our circuit architecture one layer is composed of rotation gates (R_x, R_y, R_z) acting on all qubits followed by a series of C-NOT gates coupling neighboring qubits. The trainable parameters are the angles of rotation gates and can be trained using the conventional stochastic gradient descent techniques via backpropagation adopted in the training of artificial neural networks. A quantum circuit implements a unitary, thus invertible, transformation on the initial state. This represents a great advantage for the

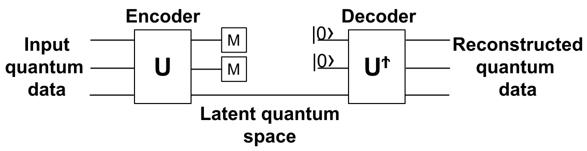


Fig. 1. Schematic representation of a quantum autoencoder.

autoencoder architecture, as the decoder can be taken as simply the inverse of the encoder quantum circuit (Fig. 1). In order to compress information, the encoder circuit has to disentangle and set to the ground state a given number of qubits [3]. The loss function is thus taken as the expected measurement values of these qubits. In this way, for the training of the circuit, it is necessary only the encoder. For the simulation and training of the PQC we have used the QIBO [6] library that can be easily integrated with Tensorflow

III. Quantum anomaly detection

Anomaly detection describes a class of algorithms that aims at the identification of rare events, which deviate significantly from the majority of the data. For the anomaly detection task, an autoencoder is trained only on data samples belonging to the normal event class. When the trained model is applied to new samples we expect the loss function to have different values for normal and anomalous data. With quantum anomaly detection we mean the anomaly detection task performed with a QAE.

A. Simulation on classic hardware

In order to find the best parameters of the proposed quantum anomaly detection algorithm the first tests have been carried out with simulations on classical hardware. Two different use cases have been considered, a simple application to handwritten digits and an application to high-energy physics.
Quantum anomaly detection on handwritten digits has been carried out on the MNIST dataset with ”zero” digits as normal data, and ”one” digits as anomalous data. The original MNIST images have been compressed down to 8×8 pixels, in this way one image can be encoded in the state amplitudes of 6 qubits. An high-energy physics application of an anomaly detection algorithm is the identification of anomalous patterns in the triggers system of a collider experiment [1]. In this study two datasets are generated, one corresponding to short decay length (standard) and one corresponding to very displaced decays (anomalous). Data are conveniently represented in the form of images of dimension 100×20 pixels, the initial state is stored in 11 qubits. The best quantum circuit ansatz has been found to be with six layers for the handwritten digits case and eight layers for the high-energy physics application. Fig. 2 shows the loss distribution for the two test datasets in the handwritten digit application and the high-energy physics application.

B. Quantum hardware implementation

The execution of quantum circuits on NISQ devices is difficult even on state-of-the-art quantum devices. The main problems come from the amplitude encoding and from the connectivity in the architecture of quantum computers. In order to make the algorithm work with these limitations, some changes and a careful adaptations have to be implemented on the PQC. Given the consequent reduction in the expressive power of the model, we decided to focus only on the simplest use-case of the handwritten digits for the quantum hardware test. To solve the connectivity problem we removed some C-NOT gate in order to map directly the circuit on the chip without adding any SWAP gate. The quantum chip employed in this work is IBM_hanoi. Moreover we decided to use only four layers for the encoder circuit. Amplitude encoding is a state preparation procedure that requires a number of C-NOT gates that grows exponentially in the number of qubits [8]. To overcome the problem we developed another PQC designed to provide a good approximation of the exact amplitude encoding while using a reduced number of gates. The final tests on quantum hardware have been carried out using circuits composed of two parts, the approximated amplitude encoding circuit and the encoder. Each circuit has been executed with 2048 shots. The distributions for normal and anomalous data for this loss function are reported in Fig. 3 for simulated circuits with no noise and real quantum circuit. It is possible to observe a significant separation between normal and anomalous data, although with a clear degradation in the case of the execution on real quantum hardware due to the high level of noise.

IV. Quantum diffusion model

Diffusion models are state of the art generative algorithms, they are used to generate samples with the same characteristics of a defined dataset. A diffusion model is trained by combining two opposite processes. In the diffusion process, noise is progressively added to the initial data. In the reverse diffusion process a neural network is trained to progressively denoise the noisy data. After training, the reverse process can be used to generate new data by starting from pure noise [4]. In the proposed quantum diffusion model the neural network has been replaced with a parametrized quantum circuit. It is important to notice that, in order to avoid mixed quantum states, only coherent noise can be added during the diffusion process.

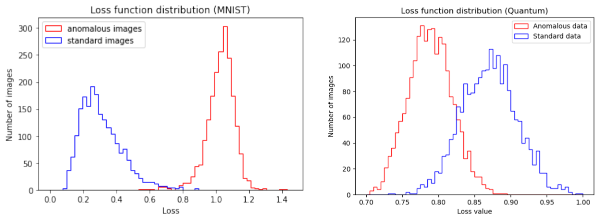


Fig. 2. Quantum autoencoder loss function values distribution for handwritten digits application (left) and for the high-energy physics application (right).

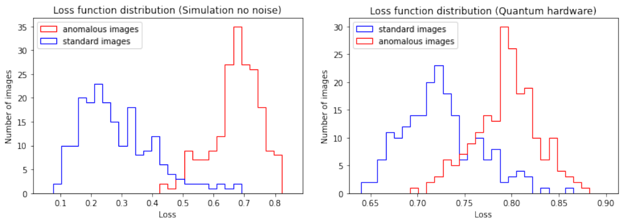


Fig. 3. Quantum autoencoder loss function values distribution. Simulated circuits with no noise (left) and a noisy quantum circuits (right).

A. Denoising circuit

In the classical diffusion model the denoising steps are usually performed using an autoencoder with a U-Net architecture [9]. This kind of neural networks have a high expressive power, that is not possible to reach with small size quantum circuits. However, a QAE can still be used to perform simple denoising steps. The simple generative task that we have tested is the generation of handwritten digits of the MNIST dataset. In order to make the quantum diffusion model work it is necessary to tune a high number of hyperparameters of the QAE and of the diffusion process. For the diffusion process the most important hyperparameters have been found to be the number of steps in the diffusion process and the schedule of the noise added at each step. In a classical diffusion model the number of diffusion steps is of the order of 10^3, as we are testing the algorithm on a simple case we have significantly reduced the number of steps, varying them from 5 to 50. A higher number of steps helps the generation of images with a better resolution but increase the chance of collapsing the model into generating a single state. For the noise addition schedule we found the better performance using uncorrelated Gaussian noise with the variance proposed in [7]. The main hyperparameters of the quantum autoencoder regard the number of layers and the number of compressed qubits. We have tested a number of layers varying from 5 to 200. By increasing the number of layers it is possible to obtain a better expressive power that result in more definition on the final states. However, increasing the number of layers makes the training process more difficult because of barren plateaus and makes the execution on NISQ devices impossible. The number of compressed qubits in the quantum autoencoder has been found to be the most important hyperparameter of the model. By increasing the number of compressed qubits it is possible to introduce more non-linearities in the circuit, resulting in a better denoising power. However, by compressing more than one or two qubits the model collapses into generating the same state. Fig. 4 shows the generation process of a quantum state encoding a ”zero” handwritten digit. Ten denoising steps have been performed with a quantum autoencoder with six layers and one compressed qubits; only the last five steps are reported in the figure.

B. Model collapse

The main limitation of the presented quantum diffusion model is the model collapse, this means that it is not able to generate a large variety of samples and tends to generate samples very close to the average over all the training set. In order observe better this phenomena, we have increased the images dimension to 16×16 pixels and used both ”zero” and ”one” digits. To mitigate the model collapse we have added noise in the reverse diffusion process, this strategy is also used in classical diffusion models. The noise can be added directly after each denoising step with a small random quantum circuit that implements a transformation close to identity. Otherwise, we have also tried adding some small rotations on the qubits in the latent space. With both these noise injection strategies it is necessary to find a good balance between the variety of the generated samples, that can be be increased by adding more noise, and the definition of the generated states, that is reduced by adding more noise. Fig. 5 shows three states where noise has been injected after each denoising step during the generation process. The first image on the left shows a case where the model has collapsed into an averaged state.

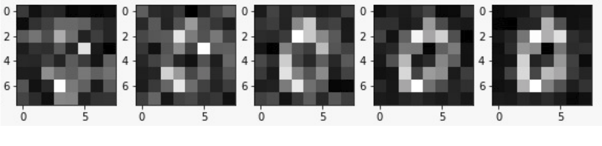


Fig. 4. Five consecutive steps of the generation process of a quantum state encoding a ”zero” handwritten digit.

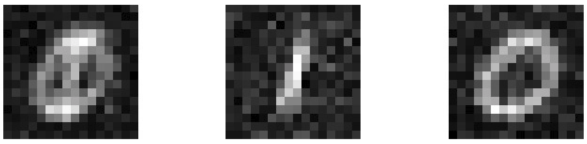


Fig. 5. Three samples generated with the quantum diffusion model by adding noise during the generation process to mitigate model collapse. In the first figure on the left the model has produced an average (collapsed) state, while in the two other cases correct samples have been generated of ”one” and ”zero” handwritten digits.

References

[1] ATLAS Collaboration. In: *Phys. Rev. D* 101 (5 2020), p. 052013.

[2] Kishor Bharti et al. In: *Rev. Mod. Phys.* 94 (1 2022), p. 015004.

[3] Carlos Bravo-Prieto. In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035028.

[4] Florinel-Alin Croitoru et al. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), pp. 1–20. DOI: 10.1109/tpami.2023.3261988. URL: https://doi.org/10.1109%5C%2Ftpami.2023.3261988.

[5] Yuxuan Du et al. In: *Phys. Rev. Research* 2 (3 2020), p. 033125.

[6] Stavros Efthymiou et al. In: *Quantum Science and Technology* 7.1 (2021), p. 015018.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. arXiv: 2006.11239 [cs.LG].

[8] Hsin-Yuan Huang et al. In: *Nature Communications* 12.1 (2021).

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. arXiv: 1505.04597 [cs.CV].

[10] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. In: *Phys. Rev. A* 103 (3 2021), p. 032430.

[11] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. In: *npj Quantum Information* 5.1 (2019).

Quantum Based Outlier Detection

Anna Bernasconi
Dept. of Computer Science
University of Pisa, Italy
0000-0003-0263-5221

Alessandro Berti
Dept. of Computer Science
University of Pisa, Italy
0000-0001-9144-9572

Gianna M. Del Corso
Dept. of Computer Science
University of Pisa, Italy
0000-0002-5651-9368

Alessandro Poggiali
Dept. of Computer Science
University of Pisa, Italy
0000-0002-1591-7925

***Abstract*—The growing interest in quantum computation is driven by its potential to surpass classical computation speed. This interest has led to an increasing focus on developing new quantum algorithms for data analysis tasks, including outlier detection, which is crucial in identifying dataset abnormalities. Some quantum outlier detection techniques have been proposed in the past, demonstrating the potential of quantum algorithms to enhance outlier detection. This study introduces a novel Quantum Outlier Detection Algorithm (QODA) that leverages variance estimation. We evaluate the performance of QODA on multiple datasets, and our results demonstrate that it enhances the efficiency of classical outlier detection while maintaining high accuracy.**

***Index Terms*—Quantum Computing, Quantum Artificial Intelligence, Quantum Outlier Detection**

I. Introduction

Detecting outliers is crucial in many fields, including finance, healthcare, and engineering. Outliers can significantly impact data analysis results, leading to incorrect conclusions or decisions. Hence, developing effective outlier detection algorithms is critical. Recently, quantum computing has shown great potential to solve computationally expensive problems for classical computers, including data analysis tasks. In the literature, various techniques for detecting outliers in quantum computing have been suggested [1], [3]. However, our paper introduces a novel approach, a hybrid quantum algorithm inspired by the classical Angle-Based Outlier Detection (ABOD) algorithm [2], an unsupervised distance-based method for detecting outliers. Our proposed hybrid quantum version of ABOD leverages the power of quantum computing to enhance the algorithm's performance.

We begin by providing an overview of the ABOD algorithm and of the notation used in the paper. Next, we propose a hybrid algorithm inspired by ABOD, which leverages a quantum subroutine to enhance the efficiency of outlier detection. Finally, we evaluate the performance of our algorithm on some synthetic and benchmark datasets and compare it with the classical technique. The experiments indicate that our hybrid quantum algorithm yields results comparable to those of the classical ABOD algorithm. Overall, our algorithm offers a promising approach for outlier detection tasks, demonstrating the potential of quantum computing to enhance data analysis in various fields.

II. Outlier Detection

A. Classical Approach

ABOD (Angle-Based Outlier Detection) is a distance-based unsupervised outlier detection method that measures the abnormality of each data point by computing the variance of the angles between the difference vectors of the given point to the other points in the dataset. Indeed, the variance of these angles depends on how different a point is from the others: for points within a cluster, the angles differ widely, while outliers are expected to present a smaller variance because they are further away from the rest of the dataset. Hence, the points with smaller ABOD values are considered to be more likely to be outliers. ABOD has several advantages over other outlier detection techniques, including detecting outliers in high-dimensional datasets. However, it may not perform well on datasets with a high degree of overlap between classes or when the dataset contains clusters of outliers. Moreover, it has a cost of $O(M^3N)$ where M is the number of records, and N is the dimension of each record; hence the cost is very high.

The classical algorithm works as follows. Assume we have M records, each with N-dimensions, stored in a matrix $X = [x_1|x_2|\cdots|x_M] \in \mathbb{R}^{N\times M}$, so that each record can be seen as belonging to a vector space of dimension N. We iterate over all records, considering each one as a pivot. Thus, let x_p, $1 \le p \le M$, the current pivot. We denote by $\theta_{ij}^{(p)}$ the angle between the vectors x_i and x_j observed from the pivot x_p, i.e., the angle between the difference vectors $\tilde{x}_i = x_i - x_p$ and $\tilde{x}_j = x_j - x_p$. Mathematically this can be written as:

$$\theta_{ij}^{(p)} = \arccos\left(\frac{\tilde{x}_i^T\tilde{x}_j}{\|\tilde{x}_i\|\|\tilde{x}_j\|}\right),$$

where all norms are 2-norms. For each $p \in \{1,\ldots,M\}$, we compute the variance v_p of $\theta_{ij}^{(p)}$, with $1 \le i < j \le M$ and $i,j \ne p$, i.e., $v_p = \mathrm{Var}\left(\theta_{ij}^{(p)}\right) = \mathbb{E}[(\theta_{ij}^{(p)})^2] - \mathbb{E}^2[\theta_{ij}^{(p)}]$, where we denoted by $E[\cdot]$ the expected value. Once the vector $V = [v_1, v_2, \ldots, v_M]$ is ready, the algorithm detects the outliers as the records with variance below a given threshold. The algorithm is usually implemented by computing the variance directly on the inner products, i.e., on the cosine of $\theta_{ij}^{(p)}$ [2].

B. Quantum Outlier Detection

In this section, we design a hybrid quantum version of ABOD called Quantum Outlier Detection Algorithm (QODA).

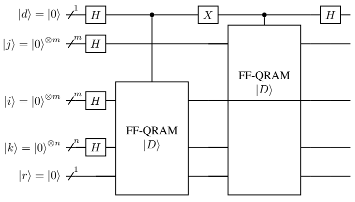

$|d\rangle = |0\rangle$
$|j\rangle = |0\rangle^{\otimes m}$
$|i\rangle = |0\rangle^{\otimes m}$
$|k\rangle = |0\rangle^{\otimes n}$
$|r\rangle = |0\rangle$
1
m
m
n
1
H
X
FF-QRAM
$|D\rangle$

Fig. 1. Computation of differences.

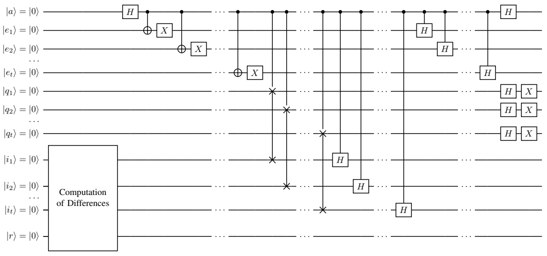

$|a\rangle = |0\rangle$
$|c_1\rangle = |0\rangle$
$|c_2\rangle = |0\rangle$
$|c_t\rangle = |0\rangle$
$|q_1\rangle = |0\rangle$
$|q_2\rangle = |0\rangle$
$|q_t\rangle = |0\rangle$
$|i_1\rangle = |0\rangle$
$|i_2\rangle = |0\rangle$
$|i_t\rangle = |0\rangle$
$|r\rangle = |0\rangle$
Computation of Differences
H
X

Fig. 2. Quantum Outlier Factor Oracle

QODA iterates over all records in a given dataset and calls a quantum subroutine to test whether the current record is an outlier on a single quantum circuit. Alternatively, we could test each record in the dataset by employing distinct quantum circuits that can run in parallel. QODA mimics the classical ADOB approach, exploiting the variance as a statistical measure to detect outliers, to reduce the computational complexity of the classical version thanks to quantum phenomena. A crucial subroutine of QODA is, therefore, an efficient quantum heuristic for estimating the variance of all angles $\theta_{ij}^{(p)}$, for a given sample x_p, taken as a pivot. We call this subroutine *Quantum Outlier Factor* (QOF). This heuristic is based on the idea of approximating the angles $\theta_{ij}^{(p)}$ with the differences $\Delta_{ij}^{(p)}$ between the vectors translated by the pivot x_p and normalized. The variance is then computed directly over the approximations $\Delta_{ij}^{(p)}$ within the same quantum circuit. More precisely, let x_p be the pivot. For all $1 \leq i \leq M$, we (classically) compute $\tilde{x}_i = x_i - x_p$. Then, we normalize all vectors adopting as default data preprocessing the Inverse Stereographic Projection (ISP), in order to keep the clusters separated, as discussed in [5]. Recall that the ISP maps N-dimensional data into the surface of a unit sphere in the $(N+1)$-dimensional space and is computed as follows. Given a vector $v = (v_1, v_2, \ldots, v_N) \in \mathbb{R}^n$, then:

$$\text{ISP}(v) = \Big(\frac{2v_1}{\|v\|^2+1}, \frac{2v_2}{\|v\|^2+1}, \ldots, \frac{2v_N}{\|v\|^2+1}, \frac{\|v\|^2-1}{\|v\|^2+1}\Big),$$

where all norms are 2-norms. Notice that $\|\text{ISP}(v)\| = 1$. Thus, let $\hat{x}_i = \text{ISP}(\tilde{x}_i) = \text{ISP}(x_i - x_p)$ denote the normalized vectors, translated by x_p. We encode all these vectors in a quantum circuit and then use a Hadamard gate to compute the differences $\Delta_{ij}^{(p)} = \frac{1}{2(M-1)\sqrt{N+1}}(\hat{x}_i - \hat{x}_j)$ in the amplitudes, appropriately normalized, as shown in Figure 1. Note that the dependence from the chosen pivot x_p is kept within $\hat{x}_i$ and $\hat{x}_j$, indeed we have

$$(\hat{x}_i - \hat{x}_j)_k = \begin{cases} \frac{2(x_{ki}-x_{kp})}{\|x_i-x_p\|^2+1} - \frac{2(x_{kj}-x_{kp})}{\|x_j-x_p\|^2+1} & \text{for } k = 1, \ldots, N \\ \frac{2(\|x_i-x_p\|^2 - \|x_j-x_p\|^2)}{(\|x_i-x_p\|^2+1)(\|x_j-x_p\|^2+1)}, & \text{for } k = N+1. \end{cases}.$$

These differences are then used, instead of the angles $\theta_{ij}^{(p)}$, for the variance computation in the same circuit, as shown in Figure 2. Indeed, the variance of the differences provides an approximated lower bound for the variance of the angles, as stated in the following theorem.

Theorem 1. *Let $\Delta^{(p)}$ be the $(M-1)^2(N+1)$-dimensional vector obtained appending all the $\Delta_{ij}^{(p)}$, for $1 \leq i,j \leq M, (i,j \neq p)$, and $\theta^{(p)}$ be the $(M-1)^2$-dimensional vector whose components are the angles $\hat{\theta}_{ij}^{(p)}$ between $\hat{x}_i$ and $\hat{x}_j$, for $1 \leq i,j \leq M, (i,j \neq p)$. Then:*

$$\mathit{Var}(\Delta^{(p)}) \mathrel{\dot{<}} \frac{1}{N+1}\mathit{Var}(\hat{\theta}^{(p)}) - \frac{1}{N}\mathbb{E}^2[\hat{\theta}^{(p)}]. \qquad (1)$$

Proof. (Sketch) Follows applying the second order Taylor expansion of the cosine function to approximate the scalar product between the vectors $\hat{x}_i$ and $\hat{x}_j$.

Using vector differences instead of angles is convenient as we can maintain the superposition of all differences in the QOF oracle and perform the variance computation directly on them within the same circuit (see Figure 2). Thus, this approach requires only one data encoding step and one final measurement. An alternative oracle could be derived by computing the inner products between all pairs of vectors with a technique inspired by [4], that does not require an intermediate measurement as the standard method based on fidelity computation. This alternative approach is currently under experimental evaluation.

The quantum subroutine for computing the QOF employs the Amplitude Estimation (AE) algorithm, using as oracle the one of Figure 2. In this way, we can estimate the amplitude of the target configuration containing the variance of the differences by measuring the m additional qubits of AE. The measurement output represents an approximation of the variance in the computational basis.

The complexity of the quantum oracle QOF is $O(\log M^2 N)$, i.e., $O(\log MN)$, assuming available efficient methods for encoding classical data. The overall hybrid algorithm QODA to solve the outlier detection problem is summarized by the pseudocode in Algorithm 1. The complexity of QODA is $O(M(MN + \log MN)) = O(M^2 N)$,

Algorithm 1 QODA

```
Input: $X$ - input data, $t$ - threshold
Output: $O$ - list of outliers
$O \leftarrow []$                                  // list of outliers initially empty
for $p \in \{1, 2, \ldots, M\}$ do                     // for each record
    $\tilde{X} \leftarrow X - |x_p\rangle\langle\mathbf{1}|$ ;                  // scale the dataset w.r.t. $x_p$
    $\hat{X} \leftarrow ISP(\tilde{X})$ ;                        // normalize $X_p$ with ISP
    $v_p \leftarrow QOF(\hat{X})$                     // Quantum Outlier Factor of $x_p$.
    if $v_p \leq t$ then                          // check if $x_p$ is an outlier
        $O.append(x_p)$ ;              // append $x_p$ to the list of outliers
return $O$ ;                                  // return the list of outliers
```

where the term MN is due to the data preprocessing. Due to the space limitation, all details, together with a formal proof of the correctness of the quantum oracle QOF, will be given in the extended version of the paper.

III. Experiments

The following section presents the numerical experiments conducted to evaluate the effectiveness of the Quantum Outlier Detection Algorithm (QODA) on a series of benchmark datasets. The initial assessment involves testing the effectiveness of the heuristic used by QODA. The second experimental evaluation aims at assessing the accuracy of QODA using the QASM SIMULATOR of Qiskit.

A. Heuristic Evaluation

The following datasets are used as benchmarks to evaluate the heuristic:

- synth3: M=500 synthetic data with dimension N=20 whose 2% are outliers.
- synth4: M=500 synthetic data with dimension N=30 whose 2% are outliers.
- synth1: M=500 synthetic data with dimension N=20 whose 10% are outliers.
- synth2: M=500 synthetic data with dimension N=30 whose 10% are outliers.
- lympho: real dataset with M=148, N=18.
- glass: real dataset with M=214, N=9.
- wbc: real dataset with M=278, N=30.

Our experiments can be divided into two types. The first type is conducted to verify the accuracy of the lower bound given by Theorem 1. In Table I, we present the approximation errors, measured in terms of Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), introduced by the relation (1). Our findings indicate that the lower bound (1) holds for all datasets considered.

TABLE I
Approximation errors of relation (1)

dataset	MSE	MAE	RMSE
synth1	2.53e-8	1.30e-4	1.60e-4
synth2	1.57e-9	3.37e-5	3.96e-5
synth3	3.02e-8	1.45e-4	1.73e-4
synth4	1.92e-9	3.85e-5	4.38e-5
lympho	3.83e-7	5.27e-4	6.19e-4
glass	7.51-3	5.98e-2	8.66e-2
wbc	3.09e-5	5.21e-3	5.55e-3

The second type of experiment aims to evaluate the effectiveness of outlier detection leveraging the variance of the differences $\Delta_{ij}^{(p)}$ instead of the variance of the angles $\theta_{ij}^{(p)}$. We compare the outlier rankings generated by two methods to achieve this objective. Table II presents the precision-at-n (P@n) metric, which we used to conduct this comparison. Specifically, we calculated P@n by determining the number of records in the top n ranks identified as outliers using the differences and angles, respectively. We can see that accuracy is very high, especially for synthetic datasets. Furthermore, we utilized the Rank Biased Overlap (RBO) [6] measure to evaluate the overall outlier ranking of our heuristic, taking as ground truth the classical algorithm working with the angles. This metric considers the ranking similarity on the top ranks, with the weighting determined by a parameter p. A higher value of p assigns greater importance to the top ranks in the final similarity calculation. The table indicates that the outlier rankings obtained by the method leveraging the variance of differences are similar to the ground truth, particularly for the highest-ranked outliers.

TABLE II
Precision-at-n (P@n) of the algorithm taking as ground truth the classical algorithm working with the angles.

dataset	n=5	n=10	n=15	n=20	n=25	n=30
synth1	0.80	0.90	0.93	0.90	0.96	0.97
synth2	1.0	1.0	1.0	0.95	0.96	0.93
synth3	1.0	1.0	1.0	0.90	0.84	0.93
synth4	1.0	1.0	0.93	0.95	0.84	0.90
lympho	0.80	0.60	0.67	0.70	0.76	0.77
glass	0.70	0.80	0.80	0.90	0.84	0.87
wbc	0.60	0.70	0.80	0.70	0.60	0.57

TABLE III
RBO measure varying the parameter p

dataset	p=0.70	p=0.75	p=0.80	p=0.85	p=0.90	p=0.95
synth1	0.65	0.69	0.74	0.78	0.83	0.88
synth2	0.67	0.71	0.76	0.81	0.86	0.91
synth3	0.88	0.89	0.89	0.90	0.90	0.91
synth4	0.65	0.69	0.74	0.79	0.84	0.88
lympho	0.52	0.54	0.57	0.59	0.62	0.66
glass	0.51	0.55	0.59	0.63	0.68	0.74
wbc	0.60	0.63	0.66	0.68	0.68	0.67

B. Quantum Simulation

Currently, the availability of quantum hardware is limited, which means that quantum computation can only be simulated using classical hardware. Nevertheless, simulating quantum computation on classical hardware is expensive, so we can only execute basic versions of our algorithm. For this reason, we run QODA on two simple synthetic datasets with a single outlier each. The first has $M = 5$ records with dimension $N = 1$, and the second has $M = 5$ records with dimension $N = 3$. By running the experiments with different random seed values for generating datasets five times, we found that QODA successfully identifies the sole outlier in both datasets with a 100% accuracy rate.

References

[1] M. Guo, H. Liu, Y. Li, W. Li, F. Gao, S. Qin, and Q. Wen, “Quantum algorithms for anomaly detection using amplitude estimation,” *Physica A: Statistical Mechanics and its Applications*, vol. 604, 2022.

[2] H.-P. Kriegel, M. Schubert, and A. Zimek, “Angle-based outlier detection in high-dimensional data,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 444–452.

[3] J.-M. Liang, S.-Q. Shen, M. Li, and L. Li, “Quantum anomaly detection with density estimation and multivariate gaussian distribution,” *Physical Review A*, vol. 99, no. 5, 2019.

[4] V. Markov, C. Stefanski, A. Rao, and C. Gonciulea, “A generalized quantum inner product and applications to financial engineering,” 2022.

[5] A. Poggiali, A. Berti, A. Bernasconi, G. M. D. Corso, and R. Guidotti, “Quantum clustering with k-means: a hybrid approach,” 2022.

[6] W. Webber, A. Moffat, and J. Zobel, “A similarity measure for indefinite rankings,” *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 4, pp. 1–38, 2010.

Latent Style-based Quantum GAN for Image Generation

Su Yeon Chang
OpenLab, IT Department
CERN
Geneva, Switzerland
su.yeon.chang@cern.ch

Michele Grossi
OpenLab, IT Department
CERN
Geneva, Switzerland
michele.grossi@cern.ch

Bertrand Le Saux
Φ-lab
European Space Agency
Frascati (RM), Italy
bertrand.le.saux@esa.int

Sofia Vallecorsa
OpenLab, IT Department
CERN
Geneva, Switzerland
sofia.vallecorsa@cern.ch

***Abstract*—In this work, we introduce the Latent Style-based Quantum GAN (LaSt-QGAN), which employs a hybrid classical-quantum approach in Generative Adversarial Networks (GANs) training for arbitrary image generation with quantum circuits. This novel approach consists of two steps: latent space embedding and hybrid data generation. We rely on auto-encoders to embed images into a low-dimensional latent space. The embedded features are then fed into the hybrid GAN for feature generation using a classical discriminator and a quantum generator, which utilizes a style-based architecture. The generated features are passed back to the classical auto-encoder to reconstruct images. Our LaSt-QGAN can be trained on realistic computer vision datasets beyond the standard MNIST, namely FashionMNIST and SAT4 (Earth Observation images). The paper provides the practical usage of quantum GAN for image generation and opens the possibility of applying it to a larger dataset in the future.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The recent advent in Quantum Computing leads to exploring possible applications in *Quantum Machine Learning* (QML) in the effort to leverage the practical quantum advantage [1], [2] in the currently existing Machine Learning (ML) techniques. It is also extended to the context of generative models, suggesting the possibility of constructing a quantum version of Generative Adversarial Networks (GANs) [3], one of the most popular ML methods to learn the implicit data distribution.

Despite recent studies on quantum GANs [4]–[6], it is yet to come that they will generate an arbitrary number of images with large sizes. Especially the *continuous* image space is difficult to be handled in quantum circuits, which are defined over *discrete* computational bases. Furthermore, images are often too large in size for the current quantum simulators and real hardware with a limited number of qubits available.

This paper proposes a hybrid GAN approach, which we call Latent Style-based Quantum GAN (LaSt-QGAN), to efficiently generate images independent of size. With respect to previous frameworks, we demonstrate the model capacity to generate various kinds of images by applying it to the MNIST, the Fashion MNIST dataset and an Earth observation image dataset, SAT4 [7]. Although the current model is limited in terms of complexity due to currently available quantum hardware, we expect to further scale up the dataset size with future hardware improvements.

II. General Framework

Our work proposes Latent Style-based Quantum GAN (LaSt-QGAN), trained in two steps: 1) Embedding images into latent space using a classical auto-encoder, 2) Generating latent representations via a quantum GAN, which consists of a quantum generator and a classical discriminator. This decomposition of the image formation process has also been formulated in classical models [8], but never tried in the hybrid quantum-classical setup.

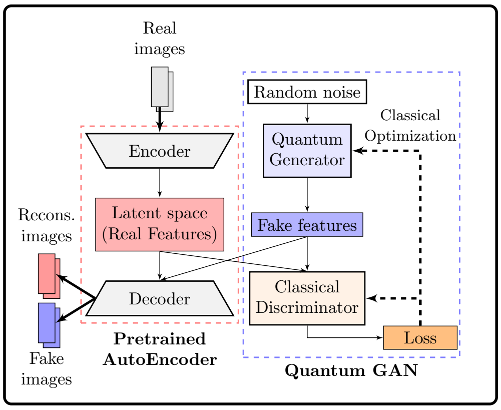
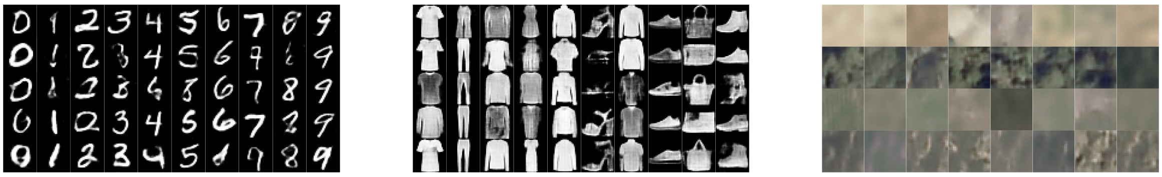


Fig. 1: Schematic diagram for LaSt-QGAN training with a convolutional auto-encoder and a quantum GAN which consists of a quantum generator and a classical discriminator.

Fig. 1 illustrates the LaSt-QGAN training schema. Essential features denoted as $\mathbf{x} \in \mathbb{R}^{\mathcal{D}_e} \sim \mathbb{P}_r$ of size $\mathcal{D}_e$, are first extracted from real images via a classical auto-encoder pre-trained on the real dataset. Those extracted features are utilized as the real dataset for the quantum GAN training. At each step, the generator reproduces fake data $\tilde{\mathbf{x}} \in \mathbb{R}^{\mathcal{D}_e} \sim \mathbb{P}_g$ from a latent noise $\mathbf{z} \in \mathbb{R}^{\mathcal{D}_\mathbf{z}}$ and the discriminator distinguishes real and fake data. At the end of the training, $\mathbb{P}_g$ should approach as close as possible to $\mathbb{P}_r$. The generated features are then passed back to the and inversely transformed into images.

The n-qubit quantum generator takes the form of a Variational Quantum Circuit (VQC), $\mathcal{U}_{\boldsymbol{\theta}}(\mathbf{z})$, to embed the classical latent noise $\mathbf{z}$ into a quantum state $\left|\Psi_{\boldsymbol{\theta}(z)}\right\rangle$ in 2^n dimensional Hilbert space $\mathcal{H}$. The particularity of the style-based architec-



(a) MNIST

(b) FashionMNIST

(c) SAT4

Fig. 3: Examples of images generated via LaSt-QGAN training. The images are presented in columns classified using a pre-trained ResNet50 for MNIST and FashionMNIST.

			MNIST		FashionMNIST	
	$G_{\boldsymbol{\theta}}$ config.	# param.	FID ↓	JSD (10^{-2}) ↓	FID ↓	JSD (10^{-2})
LaSt-QGAN	Circ. 1 ($d = 2$)	1360	17.2 ± 0.35	1.63 ± 0.09	29.42 ± 0.59	1.61 ± 0.2
	Circ. 1 ($d = 4$)	2280	14.85 ± 0.34	1.49 ± 0.18	27.59 ± 0.56	1.42 ± 0.1
	Circ. 1 ($d = 6$)	3200	14.13 ± 0.731	1.29 ± 0.1	26.89 ± 0.57	1.28 ± 0.11
	Circ. 2 ($d = 6$)	2370	14.85 ± 0.61	1.39 ± 0.11	28.1 ± 0.77	1.40 ± 0.16
	Circ. 3 ($d = 6$)	9900	$\mathbf{11.99 \pm 0.56}$	$\mathbf{1.13 \pm 0.12}$	$\mathbf{25.43 \pm 0.4}$	$\mathbf{1.17 \pm 0.23}$
Classical	$[50, 30]$	2960	18.24 ± 3.6	4.51 ± 2.0	28.32 ± 0.88	2.73 ± 0.29
	$[100, 50]$	7660	12.56 ± 0.91	1.56 ± 0.13	27.36 ± 1.51	2.81 ± 0.68

TABLE I: FID and JSD at the end of the GAN training (averaged over 10 runs), evaluated over 10k generated samples (best results in bold). With a similar model size ($\approx 3k$ parameters), LaSt-QGAN outperforms the classical GAN for all metrics. Our results are close to the result of SoTA vanilla GAN models for MNIST dataset, which have FID of 7.87 [9] and 12.88 [10].

ture is that the rotation angles in each learning layer, $U^{\ell}_{\boldsymbol{\theta}}(z)$, $\ell = 1, ..., L$, are also parameterized by the latent noises:

$$\mathcal{U}_{\boldsymbol{\theta}}(\mathbf{z}) = U^{L}_{\boldsymbol{\theta_L}}(\mathbf{z}) \cdots U^{1}_{\boldsymbol{\theta_1}}(\mathbf{z}), \quad \theta_\ell = W_\ell \mathbf{z} + \mathbf{b}_\ell \tag{1}$$

where W_ℓ is the weight matrix of size $N_{angle} \times \mathcal{D}_{\mathbf{z}}$ with N_{angle} the number of rotation angles in QNNs and $\mathbf{b}_\ell$ the bias. During the training, the model will learn $\boldsymbol{\Theta} = \{W_\ell, \mathbf{b}_\ell\}_{\ell=1,\ldots,L}$. In this paper, we test three different quantum learning layer circuits, $U^{\ell}_{\boldsymbol{\theta}}(z)$, for further simulations : Circuit1 and Circuit2 are inspired by the quantum circuit presented in Ref. [5] Ref. [4], respectively, used to learn continuous distribution, and Circuit3 consists of repeated two-qubit quantum filters which generate an arbitrary $SU(4)$ state.

At the end of the quantum circuit, we measure the expectation value of observables to retrieve the classical data. While the original architecture [5] only measures σ_z expectation, our architecture uses both σ_x and σ_z expectations, which are then concatenated to a single vector and passed to the discriminator. This way of interpreting the quantum output state allows using only n qubits for $\mathcal{D}_e = 2n$ values, also bringing an advantage in terms of quantum resources.

III. Results

This section presents the results of LaSt-QGAN trained on MNIST (28×28 pixels), Fashion MNIST (28×28 pixels) and SAT4 [7] ($28 \times 28 \times 4$ pixels), a 4-class Earth Observation images with an extra Infrared channel.

We choose the latent noise dimension $\mathcal{D}_e = 20$ for $n = 10$ qubit generator with the latent noises of $\mathcal{D}_{\mathbf{z}} = 10$, sampled randomly from a normal distribution, $\mathcal{N}(0, 1)$. The performance of LaSt-QGAN is compared with a classical GAN using the same training scheme but with a classical linear generator of two hidden layers with $[d_1, d_2]$ nodes. Furthermore, all the results are averaged over 10 runs, and the weight initialization and optimizer hyperparameters are chosen for both models to guarantee the fastest convergence. Note that the quantum circuits are simulated on the theoretical simulator.

A. *Generic results*

Fig. 3 displays the images of different datasets generated by LaSt-QGAN using the features extracted by the pre-trained convolutional auto-encoder. The images show that the model can reproduce images correctly, although further improvements are required for a higher quality of the results.

	$G_{\boldsymbol{\theta}}$ config.	FID ↓	JSD ↓
LaSt-QGAN	Circ. 3 ($d = 2$)	168.28 ± 2.06	2.07 ± 0.27
Classical	$[100, 50]$	172.6 ± 5.02	4.25 ± 0.65

TABLE II: FID and JSD at the end of the GAN training, evaluated over 10k generated SAT4 dataset images. We observe that LaSt-QGAN outperforms the classical benchmark for all metrics by using only half the number of parameters.

In Tab. I and Tab. II, we compare the best results of LaSt-QGAN and the classical GAN in terms of Fréchet Inception Distance (FID) for the quality of images, and Jensen-Shannon divergence (JSD) for the diversity. Lower the FID and JSD, the better the results. We can find that with a similar number of parameters, LaSt-QGAN outperforms the classical benchmark for all types of datasets not only in terms of quality (FID) but also in terms of diversity (JSD) in both features and

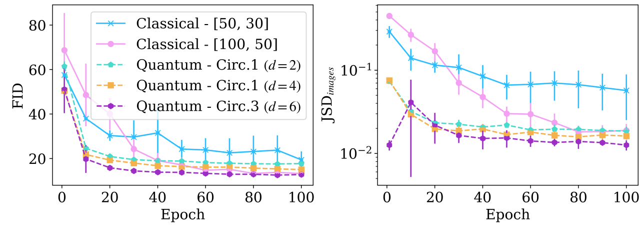
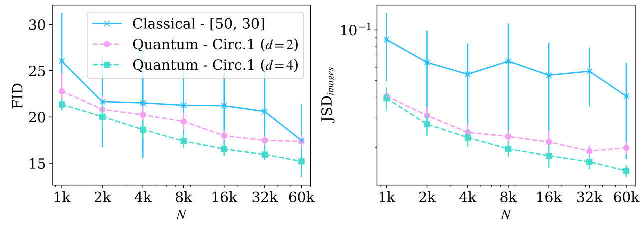


Fig. 4: Progress in FID and JSD over $10k$ MNIST samples generated during the GAN training. We observe faster convergence and higher stability with LaSt-QGAN than the classical model for all tested models. Furthermore, our model always reaches lower JSD compared to the classical model, highlighting its power to learn the hidden data distribution.

images, proving that the model can successfully learn the hidden distribution of the real data.

The convergence speed is another important factor in GAN training. On Fig. 4, the progress in different evaluation metrics is displayed for LaSt-QGAN and the corresponding classical counterpart with different architectures. As the results prove, faster convergence is exhibited for LaSt-QGAN by reaching the FID value below 20 before 20 training epochs for all depth d, which is at least twice faster than the classical one. The faster convergence is even observed using Circuit3 with depth 6 in the LaSt-QGAN , which is composed of more parameters compared to the classical GAN, proving that this advantage is independent of the number of parameters. Furthermore, small standard deviations reveal the stability of training with LaSt-QGAN during the whole training process, solving the training instability, one of the major issues in GANs.

B. Dependence on the dataset size



(a) MNIST

Fig. 5: FID and JSD obtained using LaSt-QGAN and the classical GAN with different MNIST dataset sizes, N. LaSt-QGAN has higher generalization power for both metrics even for small N compared to the classical GAN, using a less or similar number of parameters.

In this section, we train LaSt-QGAN with smaller training sets for MNIST and compare the results with the classical GAN to study the generalization power of the quantum generator. The models are trained on $N = 2^k \cdot 1000,\ k = 0, ..., 5$ samples and on the whole training set, $N = 60000$. For each N, the batch size is also adjusted to keep the same number of updates in each epoch.

Fig. 5 displays FID and JSD obtained at the end of the training with different dataset sizes N. LaSt-QGAN results in better performance in terms of both metrics compared to the classical GAN with a similar number of parameters for all N, proving higher generalization power. In particular, we reach FID less than 20 only with $N = 4k$ samples with LaSt-QGAN , while the classical model needs 32k parameters for the same value. Furthermore, lower standard deviations obtained in all cases with LaSt-QGAN emphasize once more the stability of the quantum generator compared to the classical one.

IV. Conclusion

In this paper, we introduced our novel LaSt-QGAN framework, which uses a classical latent embedding and quantum GAN for latent representation learning to investigate the advantage of using QNNs for image generation. Our results show that the model can successfully generate images with a quality comparable to the classical GAN. Especially, different evaluation metrics prove that the quantum GAN can reach, even outperform, the performance of classical GAN with a similar number of parameters in terms of both quality and diversity for all tested datasets. Furthermore, we investigated the performance of the models with varying sample numbers and found that LaSt-QGAN reaches a similar performance compared to the classical GAN with a smaller dataset size, showing the practical advantage of the model for generating images. Ultimately, this work paves the way towards efficient image generation for the era when the Fault Tolerant Quantum (FTQ) computers are available.

References

[1] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, pp. 209–212, mar 2019.

[2] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, “Generalization in quantum machine learning from few training data,” *Nature Communications*, vol. 13, p. 4919, Aug 2022.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014.

[4] J. Romero and A. Aspuru-Guzik, “Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions,” 2019.

[5] C. Bravo-Prieto, J. Baglio, M. Cè, A. Francis, D. M. Grabowska, and S. Carrazza, “Style-based quantum generative adversarial networks for Monte Carlo events,” *Quantum*, vol. 6, p. 777, Aug. 2022.

[6] M. S. Rudolph, N. B. Toussaint, A. Katabarwa, S. Johri, B. Peropadre, and A. Perdomo-Ortiz, “Generation of high-resolution handwritten digits with an ion-trap quantum computer,” *Phys. Rev. X*, vol. 12, p. 031010, Jul 2022.

[7] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. R. Nemani, “Deepsat - A learning framework for satellite imagery,” *CoRR*, vol. abs/1509.03602, 2015.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

[9] J. Wei, M. Liu, J. Luo, A. Zhu, J. Davis, and Y. Liu, “Duelgan: A duel between two discriminators stabilizes the gan training,” 2021.

[10] D. Lazcano, N. Fredes, and W. Creixell, “Hyperbolic generative adversarial network,” *CoRR*, vol. abs/2102.05567, 2021.

Unconventional Chemical Contributions to Quantum Artificial Intelligence

Pier Luigi Gentili
Department of Chemistry, Biology, and Biotechnology
Università degli Studi di Perugia
Perugia, Italy
pierluigi.gentili@unipg.it

***Abstract*—Chemistry can significantly contribute to the development of Quantum Artificial Intelligence. All the molecules abide by the quantum-mechanical laws. Hence, in principle, any chemical system can be exploited to encode and process quantum information. However, it is well-known that decoherent phenomena hinder quantum computation, and extreme conditions are usually required to manipulate the elementary units of quantum information, which are the qubits. In this work, alternative strategies to exploit the computational parallelism of the molecular quantum world are proposed. They do not require extreme experimental conditions. They only need to be known and used after formulating the appropriate algorithms.**

***Index Terms*—molecular conformations, molecular parallelism, fuzzy entropy, neuromorphic engineering.**

I. Introduction

Any molecule, being a microscopic entity, obeys the laws of quantum mechanics. Therefore, any molecule can, in principle, be used to encode quantum information. For instance, a molecule that has two accessible quantum states, labelled as $|0>$ and $|1>$, and can stay in a superposition condition for the two states can be exploited to encode the elementary unit of quantum information [1], which is the qubit $|\Psi>$:

$$|\Psi> = a|0> + b|1> \tag{1}$$

In equation (1), a and b are two complex numbers that verify the normalisation condition:

$$a^2 + b^2 = 1 \tag{2}$$

The coefficients a^2 and b^2 represent the probabilities of having the collapse of the qubit in the $|0>$ and $|1>$ states, respectively. The qubit can be described as a unit vector in a two-dimensional Hilbert space. The states $|0>$and $|1>$ are the computational basis states and form an orthonormal basis for this vector space. Therefore, the qubit can also be described by the function below:

$$|\Psi> = cos(\theta/2)|0> + e^{i\phi} sin(\theta/2)|1> \tag{3}$$

In equation (3), the two angles define a point on the unit three-dimensional sphere (see Fig.1), called the Bloch sphere [2]. Logic operations on qubits can be visualised as reversible rotations of the unit vectors on the Bloch sphere, preserving the norm of quantum states. Quantum computers promise to be immensely powerful due to their parallelism. For instance, if a quantum computer consists of n unmeasured qubits, it can be in an arbitrary superposition of up to 2^n different states simultaneously (the Hilbert space will be of 2^n dimensions), differently from a classical computer that can only be in one of the 2^n states at a time [3]. The superposition can regard the quantum states of far-apart particles if they are entangled [4]. The main difficulty in building a quantum computer comes from the fact that quantum states must constantly contend with insidious interactions with their environment (for example, a molecule colliding with another molecule, being all subjected to Brownian motion fed by thermal energy) that causes loss of quantum coherence. If so, any qubit collapses in one of the two constitutive states. The decoherent phenomena bring about transitions from quantum to classical information [5]. Nevertheless, it is still possible to exploit other forms of molecular parallelism. These alternative forms of parallelism have been proposed to implement fuzzy sets and fuzzy logic systems [6] [7] [8], and they are described in this contribution. They might blaze an innovative trail in the burgeoning field of Quantum Artificial Intelligence.

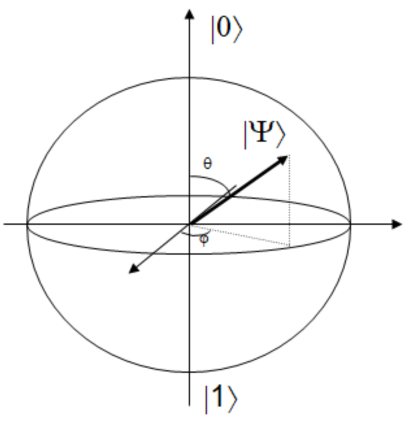


Fig. 1. The qubit $|\Psi>$ is shown in the Bloch sphere.

Identify applicable funding agency here. If none, delete this.

II. The Fuzziness of the Molecular World: An Unconventional Kind of Quantum Parallelism

A. Molecular Fuzzy Sets

When a molecular compound exists in many conformers and/or it experiences distinct micro-environments, it is describable as a quantum mixed state:

$$\rho = \sum_{i=1}^{N} w_i |\psi_i >< \psi_i| \tag{4}$$

In Equation (4), w_i represents the weight of the i-th wavefunction and corresponds to its probability. Such a quantum mixed state ρ has been used to implement a Fuzzy set when the terms w_i (for $i = 1, ..., n$) have been interpreted as the degrees of membership of the different $|\psi_i >$ to ρ. Hence, w_i is also the fuzzy unit of information. The molecular system represented by ρ has a Fuzzy Entropy H [9] given by:

$$H = -\frac{1}{logN} \sum_{i=1}^{N} w_i log(w_i) \tag{5}$$

H assumes any real value included between 0 and 1. If the compound exists in just one state (i.e., one conformer experiencing just one micro-environment), $N = 1$, $w_i = 1$, and $H = 0$. On the other hand, if the compound exists in N distinct states (as it has N conformers or experiences N different micro-environments), which are equally probable, then $w_i = 1/N$ and $H = 1$. Of course, there is an infinite number of other possibilities which originate H values included between 0 and 1. The collection of N states, each described by its wavefunction $|\psi_i>$ (with $i = 1, \ldots, N$), constitutes a molecular fuzzy set. When the compound is perturbed by proper physical or chemical inputs, the w_i values change simultaneously. It is an unconventional form of quantum parallelism that allows to process fuzzy information at the molecular level, and hence what can be named ”quantum-fuzzy” information.

The determination of the $w_i$values appearing in Equations (4) and (5) can be accomplished by recording spectroscopic time-resolved signals and fitting them through the Maximum Entropy Method (MEM) [10]. For instance, MEM fits a fluorescent decay signal (I_{em}) by using a poly-exponential function with N terms:

$$I_{em} = \sum_{i=1}^{N} w_i e^{-t/\tau_i} \tag{6}$$

The relative weight w_i of the i-th lifetime (τ_i) represents the degree of membership of the wavefunction $|\psi_i>$ to the quantum mixed state ρ. The w_i values depend on the “chemical context”. Likewise, in Fuzzy logic, any set has a shape and position which are sensitive to the context. It is worthwhile noticing that the determination of H can also be accomplished through other techniques that allow collecting morphological, structural distributions, spectroscopic bands or chromatographic peaks. Any compound or material exhibits different H values depending on its physicochemical context. Any context-dependent quantum mixed state is potentially valuable for processing information in a parallel manner. An example is shown in Fig.2: it is a molecular ”transistor” that can be switched through electromagnetic radiation and thermal energy between two states, SpO and MC. The MC state exists as an abundant collection of conformers, whose composition is sensitive to the physicochemical context.

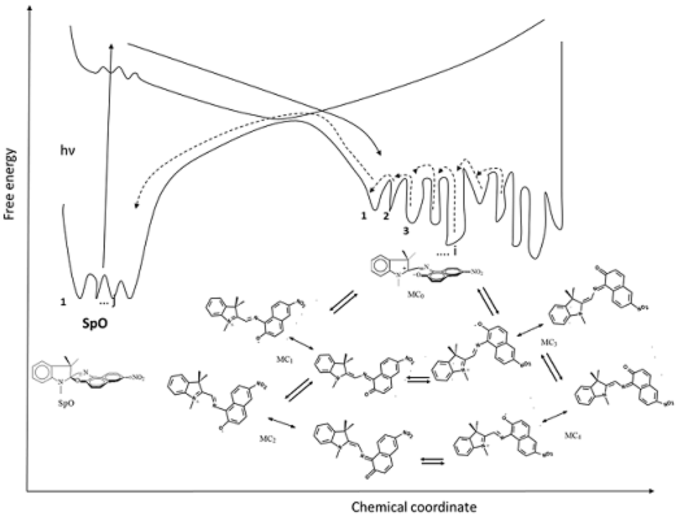


Fig. 2. A molecular transistor assuming two distinct states, SpO and MC. The MC state exists as a collection of conformers, which is sensitive to the physicochemical context. This compound can be exploited to implement the unconventional form of quantum parallelism based on micro-heterogeneity.

B. Granulation of Physicochemical Variables

Further growth of parallelism is achievable when two or more molecular compounds that exist as quantum mixed states are joined together. If the number of compounds is $j = 1, \ldots, M$, and for each of them the number of quantum states is $i = 1, \ldots, N_j$, the entire system is describable as the following quantum mixed state in the case they do not strongly affect each other:

$$\Pi = \sum_{j=1}^{M} \rho_j = \sum_{j=1}^{M} \sum_{i=1}^{N_j} w_{ji} |\psi_{ji} >< \psi_{ji}| \tag{7}$$

The sum $\sum_{i=1}^{N_j} w_{ji} = 1$ for each $j = 1, \ldots, M$. The total Fuzzy Entropy becomes:

$$H = \sum_{j=1}^{M} H_j = -\sum_{j=1}^{M} \frac{1}{logN_j} \sum_{i=1}^{N_j} w_{ji} log(w_{ji}) \tag{8}$$

Each compound is a Fuzzy set, and their mixture allows the granulation of the physicochemical variables. Systems of properly selected compounds constitute a strategy to build Fuzzy Logic Systems and mimic the parallelism of the sensory systems shown by living beings [11]. For instance, the three retinals humans have in the photosensitive cells, called cones, partition the visible spectral region into three fuzzy sets (see Fig.3). The three retinals work in a sort of quantum parallelism and confer humans the capability of distinguishing around 200 hues, 20 levels of saturation and 200 levels of brightness, i.e., up to almost 1 million of colours.

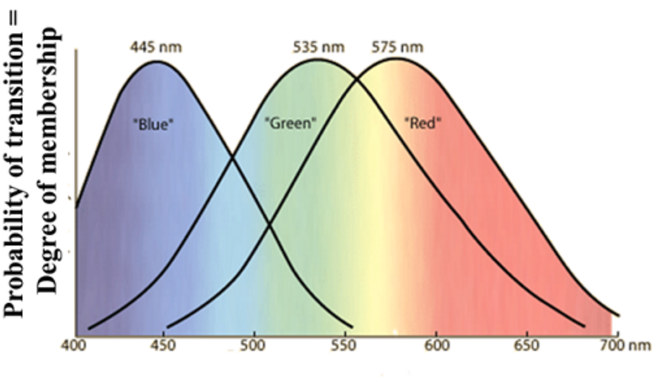


Fig. 3. Absorption spectra of the three cones on the retina, which granulate the visible spectral region

Such an approach has been mimicked to implement Biologically Inspired Photochromic Fuzzy Logic Systems that extend human vision from the visible to the UV spectral region [12] [13].

C. The Contribution of Neuromorphic Engineering

Another unconventional chemical strategy to contribute to the development of Quantum Artificial Intelligence involves the research field of neuromorphic engineering. In neuromorphic engineering, neural surrogates are implemented to design brain-like computing machines or for neuroprosthesis [14]. Traditionally, neural surrogates are implemented in hardware. However, some nonlinear chemical systems in wetware (i.e., in fluid solution) that mimic neural dynamics are also valuable. They can communicate through optical signals and give rise to spontaneous synchronization phenomena [15] [16]. When a surrogate of a pacemaker neuron sends an oscillatory optical signal to a photochromic compound, the latter becomes coloured, and the values of its chromaticity coordinates (x, y, and z) oscillate. The response of the photochromic compound can be depicted through a unitary vector that rotates within the space defined by the chromaticity coordinates [17]. If two or more distinct photochromic compounds receive the same oscillatory signals, they will be represented by distinct vectors that rotate synchronously like superposed quantum states. It is possible to manipulate these colour vectors by properly tuning the wavelength of the excitatory signal sent by the pacemaker surrogate [18].

D. Perspectives

This contribution highlights some unconventional strategies to develop Quantum Chemical Artificial Intelligence [7] [19]. These strategies rely on some forms of molecular parallelism, which do not require the delicate and fragile coherence of superposed quantum states. They can be technically accomplished in much easier and cheaper ways. Therefore, they should be known by the vast community of scientists who are contriving new algorithms to face intractable NP problems and recognise variable patterns more effectively.

Acknowledgment

P. L. G. thanks the European Union - NextGenerationEU under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041 - VITALITY. P. L. G. acknowledges Università degli Studi di Perugia and MUR for support within the project Vitality.

References

[1] B. Schumacher, “Quantum Coding,” Phys. Rev. A Vol. 51, pp. 2738-2747, 1995.

[2] M.A. Nielsen, I. L. Chuang, “Quantum Computation and Quantum Information.” Cambridge University Press, 2004.

[3] C.H. Bennett, D.P. DiVincenzo, ”Quantum information and computation.” Nature Vol. 404, pp. 247-255, 2000.

[4] M.B. Plenio, S. Virmani, ”An introduction to entanglement measures.” Quantum Inf. Comput. Vol. 7, pp. 1-51, 2007.

[5] W.H. Zurek, ”Decoherence and the transition from quantum to classical.” Physics Today. Vol. 44, pp. 36-44, 1991.

[6] P.L. Gentili, ”The Fuzziness of the Molecular World and Its Perspectives.” Molecules. Vol. 23, pp. 2074, 2018.

[7] P.L. Gentili, ”Establishing a New Link between Fuzzy Logic, Neuroscience, and Quantum Mechanics through Bayesian Probability: Perspectives in Artificial Intelligence and Unconventional Computing.” Molecules Vol. 26, pp. 5987, 2021.

[8] P.L. Gentili, P. Stano, ”Chemical neural networks inside synthetic cells? A proposal for their realization and modeling.” Front. Bioeng. Biotechnol. Vol. 10, pp. 927110, 2022.

[9] P.L. Gentili, J. Perez-Mercader, ”Quantitative estimation of chemical microheterogeneity through the determination of fuzzy entropy.” Front. Chem. Vol. 10, pp. 950769, 2022.

[10] E.T. Jaynes, ”Maximum Entropy and Bayesian Methods in Applied Statistics;” J.H. Justice, Ed.; Cambridge University Press: Cambridge, UK, 1986.

[11] P.L. Gentili, ”The human sensory system as a collection of specialized fuzzifiers: A conceptual framework to inspire new artificial intelligent systems computing with words.” J. Intell. Fuzzy Syst. vol. 27, pp. 2137-2151, 2014.

[12] P.L. Gentili, A.L. Rightler, B.M. Heron, C.D. Gabbutt, ”Extending human perception of electromagnetic radiation to the UV region through biologically inspired photochromic fuzzy logic (BIPFUL) systems.” Chem. Comm. Vol. 52, pp. 1474-1477, 2016.

[13] P.L. Gentili, A.L. Rightler, B.M. Heron, C.D. Gabbutt, ”Discriminating between the UV-A, UV-B and UV-C regions by novel Biologically Inspired Photochromic Fuzzy Logic (BIPFUL) systems: A detailed comparative study.” Dyes and Pigments. Vol. 135, pp. 169-176, 2016.

[14] J. Zhu, T. Zhang, Y. Yang, R. Huang, ”A comprehensive review on emerging artificial neuromorphic devices.” Appl. Phys. Rev. Vol. 7, pp. 011312, 2020.

[15] P.L. Gentili, M.S. Giubila, R. Germani, A. Romani, A. Nicoziani, A. Spalletti, B.M. Heron, ”Optical Communication among Oscillatory Reactions and Photo-Excitable Systems: UV and Visible Radiation Can Synchronize Artificial Neuron Models.” Angew. Chem. Int. Ed. Vol. 56, pp. 7535-7540, 2017.

[16] P.L. Gentili, M.S. Giubila, R. Germani, B.M. Heron, ”Photochromic and luminescent compounds as artificial neuron models.” Dyes and Pigments. Vol. 156, pp. 149-159, 2018.

[17] P.L. Gentili, ”The fuzziness of a chromogenic spirooxazine.” Dyes and Pigments. Vol. 110, pp. 235-248, 2014.

[18] B. Bartolomei, B.M. Heron, P.L. Gentili, ”A contribution to neuromorphic engineering: neuromodulation implemented through photochromic compounds maintained out of equilibrium by UV–visible radiation.” Rend. Fis. Acc. Lincei Vol. 31, pp. 39-52, 2020.

[19] P.L. Gentili, ”Photochromic and luminescent materials for the development of Chemical Artificial Intelligence.” Dyes and Pigments. Vol. 205, pp. 110547, 2022.

Session II: Quantum Optimization

Quantum Annealing for Constraint Satisfaction and Constrained Optimization Problems

Philippe Codognet
JFLI - Japanese-French Laboratory for Informatics
CNRS / Sorbonne University / University of Tokyo, Tokyo, Japan
codognet@is.s.u-tokyo.ac.jp

***Abstract*—In Quantum Annealing problems are encoded in quantum Hamiltonians (energy functions) and quantum dynamics is used to find solutions (ground states of minimal energy). Quantum computers such as the D-Wave systems are indeed implementing those ideas in hardware, as well as "quantum-inspired" devices based on classical electronics such as Fujitsu's Digital Annealing Unit and all those systems use the same modeling language: Quadratic Unconstrained Binary Optimization (QUBO). However, QUBO is a low-level language and for modeling classical AI problems such as constraint satisfaction and constrained optimization problems, we need to introduce higher-level abstraction in order to define complex constraints. We investigate different QUBO formulations for the encoding of integers into Booleans and for the encoding of constraints.**

***Index Terms*—quantum annealing, constraint satisfaction problems, constrained optimization problems, QUBO, combinatorial optimization, search algorithms**

I. Introduction

We present in this paper a general overview of previous works on Quantum Annealing and QUBO modeling which appeared in several recent publications [1]–[6].

In the domain of Artificial Intelligence, search algorithms and combinatorial problem solving are old but still very active research fields, cf. SAT (SATisfiability of Boolean formulas) and Constraint Programming. These topics can obviously benefit from the advances in Quantum Computing. Indeed, the use of quantum computers to solve concrete problems in the domain of combinatorial optimization and decision science, has started to raise tremendous interest in the last years, in both the *gate model* paradigm with the Quantum Approximate Optimization Algorithm (QAOA) and in the *quantum adiabatic computing* paradigm with Quantum Annealing (QA) [7], [8].

QA is conceptually derived from simulated annealing, but taking advantage of the *quantum tunneling* effect to overcome energy barriers and therefore escape local minima during the computation. A very interesting point is that, because QA is based on the formulation of problems as Ising models and the computation of Hamiltonians, it is related to the Quadratic Unconstrained Binary Optimization (QUBO), which has now become the standard input language for all Ising Machines [9]: quantum computers (D-Wave systems, NTT's Coherent Ising Machine, NEC quantum annealing processor), "quantum-inspired" dedicated hardware (Fujitsu's Digital Annealer Unit, Hitachi's CMOS Annealing Machine) and systems based on clusters of parallel machines (Toshiba's Simulated Bifurcation Machine, Fixstars Amplify Annealing Engine, NEC Vector Annealing).

However, QUBO is a low-level language and, for modeling classical AI problems such as constraint satisfaction and constrained optimization problems, we need to introduce higher-level abstractions in order to define complex constraints, similar to what was done in the domain of Constraint Programming [10]. Then, the issue of the different formulations and encodings of constraints in QUBO appears and provides interesting research issues.

We would like in this paper to make a global presentation of the domain of quantum annealing for solving combinatorial problems from the viewpoint of constraint programming and present general issues on the modeling aspects of the QUBO formalism and its use for quantum annealing. We advocate that, although QA is not yet mature enough to revolutionize this domain, it opens many interesting research perspectives and could become an effective challenger to classical methods in the years to come.

II. Quantum Annealing and QUBO

In the last decade, Quantum Annealing (QA) has been gaining success as a new approach for solving combinatorial problems, thanks to the development of quantum hardware such as D-Wave computers [11] and "quantum-inspired" hardware which makes it possible to experiment on a variety of abstract or real-life problems [12].

In the QA paradigm, combinatorial optimization problems can be described by Ising models and Ising Hamiltonians, the ground states of which correspond to the minimal solutions of the original problem, see for instance [13]. Interestingly, Ising models are equivalent to formulations in Quadratic Unconstrained Binary Optimization (QUBO). Therefore QUBO has become in the last years the standard input language for all quantum and quantum-inspired annealing hardware.

Consider n Boolean variables $x_1, ..., x_n$, a QUBO problem consists in minimizing an *objective function* defined by a quadratic expression over $x_1, ..., x_n$: $\sum_{i \leq j} q_{ij} x_i x_j$

It is therefore usual to represent a QUBO problem by a vector x of n binary decision variables and a square $n \times n$

matrix Q with coefficients q_{ij}, as the problem can be written:

$$\text{minimize } y = x^t Q x, \text{where } x^t \text{ is the transpose of } x$$

Observe that, as x_i are Boolean variables, $x_i^2 = x_i$, thus this quadratic formulation also includes a linear part, which corresponds to the coefficients on the diagonal of the Q matrix.

Although it is possible to express simple combinatorial problems directly in Ising or QUBO models, cf. [13], more complex problems are better modeled using the notion of *constraints*, as developed in the Constraint Programming paradigm [10]. This is exemplified by the recent development of Constrained Binary Models (CQM) by D-Wave Inc. [14] which integrate linear equations and inequations for QUBO models. However, general constraints are not limited to linear constraints but can also include symbolic constraints representing any logical relation between problem variables. Such constraints can be introduced in QUBO models by adding *penalties* (quadratic expressions with minimal value when the constraint is satisfied) to the objective function. Encoding constraints as penalties in QUBO can be done in different ways and an important issue is to relate and compare different QUBO formulations, in particular for performance evaluation.

III. From Integer to Booleans

Constraint Satisfaction Problems (CSPs) and Constrained Optimization Problems (COPs) are generally modeled in terms of integer variables and arithmetic or symbolic constraints on those variables. We thus need to encode these two aspects in QUBO.

There are mainly two schemes currently used for encoding integer variables in the QA community: the classical *one-hot* encoding and the *unary/domain-wall* encoding.

One-hot encoding encodes an integer variable $x \in \{1, \ldots, n\}$ by n Boolean variables x_i that have value 1 if x has value i and value 0 otherwise. To enforce that each variable as only one value we need the constraint $\sum_{i=1}^{n} x_i = 1$, which is usually called the *one-hot* constraint. Remarking that $\sum_{i=1}^{n} x_i = 1 \iff (\sum_{i=1}^{n} x_i - 1)^2 = 0$, we can develop this quadratic expression and remove the constant term to obtain the (quadratic) *penalty* expression for each original integer variable x, to be added to the QUBO objective function: $- \sum_{i=1}^{n} x_i + 2 \sum_{i<i'} x_i x_{i'}$

Domain-wall encoding, proposed in [15] in an Ising setting, encodes an integer variable $x \in \{1, \ldots, n\}$ by $n-1$ Boolean variables x'_i. Indeed, it is equivalent in a Boolean setting to the well-known *unary* encoding on a fixed number of bits: a number n is encoded by n bits set to 1, followed by zeros. This is also called *thermometer* encoding, and gives a unique unary encoding for each integer. For instance, 1110 is a valid unary/domain-wall encoding and represents the integer value 3, while 1001 and 0010 are not valid unary/domain-wall encodings. To be a valid encoding, we need to enforce the following constraint [2]: $\forall i \in \{0, \ldots, n-3\}, \quad x'_i \geq x'_{i+1}$. As a Boolean constraint $x \geq y$ can be represented in QUBO by the (quadratic) penalty $y - xy$, this corresponds to the following penalty in QUBO, for each original integer variable x: $\sum_{i=0}^{n-3} (x'_{i+1} - x'_i \; x'_{i+1})$

IV. Constraints as Penalties in QUBO

For more complex problems we need to define different types of constraints, as exemplified in the Constraint Programming paradigm [10], where problems are modeled by using a larger vocabulary of constraints, in particular the so-called ”global” or high-level constraints that involve symbolic and non-linear relations between problem variables.

Constraint expressions can be introduced in QUBO models as *penalties* in the objective function to minimize, that is, as quadratic expressions whose value is minimal when the constraint is satisfied. An easy way to formulate such a penalty is to create a quadratic expression which has value 0 if the constraint is satisfied and a positive value if the constraint is not satisfied, representing somehow the degree of violation of the constraint. [16] describes a set of simple penalty expressions for basic arithmetic constraints with two Boolean variables. It also proposes a general scheme to handle linear equations and also linear inequations by introducing slack variables. An interesting point is that penalties are *compositional*, that is, they can be added together to the QUBO objective function in order to represent the conjunction of constraints in the original problem. Nevertheless, when penalty expressions are added to the objective function, a penalty coefficient p_i has to be added for each penalty expression in order to make it compatible with the original objective function to optimize. This means that we have to be sure that the penalties coefficient corresponding to the constraint are large enough to make such constraint ”hard”, whereas the objective function is to be considered ”soft”.

Symbolic, non-linear constraints that are commonly used for modeling COPs and CSPs can also be considered. For instance, many classical problems such as the N-Queens, Magic Square, Quadratic Assignment Problem and Travelling Salesman Problem are subject to the constraint that each feasible solution forms a *permutation*. This can be formulated in QUBO with both one-hot and unary encoding of integers.

Consider n integer variables x_i with values in the domain $\{1, \ldots, n\}$, each x_i being one-hot encoded by n Boolean variables x_{ij}. A permutation constraint on $(x_1, \ldots, x_n)$ can be encoded in QUBO by $2 \times n$ pseudo-Boolean constraints representing one-hot constraints: one constraint for each of the n variables x_i stating that it can have only one value k (i.e., one-hot encoding) and one constraint for each of the n values k stating that it can be assigned to only one variable x_i. Each of these constraints is a one-hot constraint as described previously ($\sum x_{ij} = 1$, with different index sets), generating thus a penalty of the form described previously.

Adding all such penalties together and simplifying the quadratic expression gives the overall penalty for the permutation constraint:

$$\sum_{i=1}^{n} \sum_{j<j'} x_{ij} x_{ij'} \;+\; \sum_{j=1}^{n} \sum_{i<i'} x_{ij} x_{i'j} \;- \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}$$

Let us now consider n integer variables $x_0, \ldots, x_{n-1}$ with values in the domain $\{0, \ldots, n-1\}$, each x_i being unary/domain-wall encoded by $n-1$ Boolean variables x_{ij}. We observe that $(x_0, \ldots, x_{n-1})$ is a permutation of $(0, \ldots, n-1)$ if and only if: $\forall j \in \{0, n-2\} \;\; \sum_{i=0}^{n-1} x_{ij} = (n-1) - j$

Each of these $n-1$ pseudo-Boolean equations corresponds to a penalty: $(2(j-n)+3) \sum_{i=0}^{n-1} x_{ij} + 2 \sum_{i<i'} x_{ij} x_{i'j}$

We can sum up all these $n-1$ penalties and obtain the overall penalty corresponding to a permutation constraint in unary/domain-wall encoding:

$$\sum_{j=0}^{n-2} \; (\; (2(j-n)+3) \sum_{i=0}^{n-1} x_{ij} + 2 \sum_{i<i'} x_{ij} x_{i'j} \;)$$

We can see that the penalties for the *permutation* constraint are very different in one-hot and unary/domain-wall encodings and can have very different performance on some problems [4]

V. Learning Constraint formulations in QUBO

The definition of penalties corresponding to constraints can be simple (e.g., in the case of the permutation constraint in one-hot encoding) or sometime more complex (e.g., in the case of the permutation constraint in unary/domain-wall encoding).

An interesting approach, presented in [6] is to learn the penalty part of the QUBO matrices corresponding to complex integer constraints from examples, i.e. from tuples satisfying the given constraint. This learning process can be done by combining a limited number of sub-matrix patterns which are combined in order to create the appropriate QUBO matrix. Experiments show that this method has good scalability and is robust, as correct QUBO matrices can be learned over very scarce data (about 10 training elements only). Complex constraints such as permutation, all-different, linear sum, ordered or channel constraints can indeed be learned automatically.

VI. Experiments with Constraint Satisfaction and Constrained Optimization Problems

We have presented in [1], [3], [4] the QUBO formulation of well-known Constraint Satisfaction Problems such as N-queens, Magic Square, and an hard combinatorial problems such as the Costas Array Problem, as well as some performance results. The Costas Array Problem is interesting because its natural modeling in not quadratic but quartic (degree 4) and thus a "quadratization" phase is needed in order to produce a QUBO model. The Magic Square Problem is also interesting because it involves different types of constraints (many linear equations and a permutation constraint) which are interacting in the solving process. Moreover experiments have shown that in this case the unary/domain-wall encoding if more efficient than one-hot encoding, showing that choosing the best QUBO formulation is essential in order to have good performances with annealing solvers.

In [5] we compared the performance of a QUBO model of the Quadratic Assignment Problem (QAP) executed on quantum-inspired annealing machines w.r.t. metaheuristics approaches on classical hardware. Clearly, classical metaheuristics methods are still more efficient, but quantum annealing could be a possible challenger if quantum annealing systems can scale up and handle larger instances.

VII. Conclusion

We presented the use of quantum annealing for solving combinatorial problems by utilizing a high-level constraint language based on integer variables and a large variety of constraints, which are then translated in QUBO, the input language of quantum annealing systems. Although current quantum annealing systems are very limited in the size of the combinatorial problems that can be handled, they can provide proof-of-concept experiments. Moreover, quantum-inspired hardware can be used today to experiment with medium-size instances and to investigate best solutions for new encodings or transformations. We believe that with the progress to be expected for quantum hardware in the next years, this approach can become a competitive tool for solving combinatorial problems in the near future.

References

[1] P. Codognet, "Constraint solving by quantum annealing," in *ICPP Workshops 2021: 50th International Conference on Parallel Processing*, F. Silla and O. Marques, Eds. ACM, 2021, pp. 25:1–25:10.

[2] ——, "Domain-wall / unary encoding in QUBO for permutation problems," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022, pp. 167–173.

[3] ——, "Modeling the costas array problem in QUBO for quantum annealing," in *Evolutionary Computation in Combinatorial Optimization - 22nd European Conference, EvoCOP 2022*, ser. Lecture Notes in Computer Science, vol. 13222. Springer, 2022, pp. 143–158.

[4] ——, "Comparing QUBO models of the magic square problem for quantum annealing," in *Metaheuristics - 14th International Conference, MIC 2022*, ser. Lecture Notes in Computer Science, vol. 13838. Springer, 2022, pp. 470–477.

[5] P. Codognet, D. Diaz, and S. Abreu, "Quantum and digital annealing for the quadratic assignment problem," in *2022 IEEE International Conference on Quantum Software (QSW)*, 2022.

[6] F. Richoux, J.-F. Baffier, and P. Codognet, "Learning QUBO Models for Quantum Annealing: A Constraint-based Approach," in *ICCS23, 23rd International Conference on Computational Science*, Jul 2023.

[7] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E*, vol. 58, pp. 5355–5363, Nov 1998.

[8] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem," *Science*, vol. 292, no. 5516, pp. 472–475, 2001.

[9] N. Mohseni, P. McMahon, and T. Byrnes, "Ising machines as hardware solvers of combinatorial optimization problems," *Nature Reviews Physics*, vol. 4, pp. 363–379, 2022.

[10] F. Rossi, P. van Beek, and T. Walsh, Eds., *Handbook of Constraint Programming*, ser. Foundations of Artificial Intelligence. Elsevier, 2006.

[11] B. P. I., E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz, and J. Whittaker, "Architectural considerations in the design of a superconducting quantum annealing processor," *IEEE Transactions on Applied Superconductivity*, vol. 24, no. 4, pp. 1–10, 2014.

[12] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: introduction and review," *Reports on Progress in Physics*, vol. 85, no. 10, p. 104001, sep 2022.

[13] A. Lucas, "Ising formulations of many NP problems," *Frontiers in Physics*, vol. 2, 2014.

[14] Anonymous, "Measuring performance of the LEAP constrained quadratic model solver," D-Wave Inc., Tech. Rep., November 2022.

[15] N. Chancellor, "Domain wall encoding of discrete variables for quantum annealing and QAOA," *Quantum Science and Technology*, vol. 4, p. 045004, 2019.

[16] F. W. Glover, G. A. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," *4OR*, vol. 17, no. 4, pp. 335–371, 2019.

A preliminary study on Genome Assembly using Quantum Annealing

J.A. Álvarez-Ocete
University of Granada
Granada, Spain
joseantonioao32@gmail.com

E. Pomares-Porras
University of Granada
Granada, Spain
emilio.pomares.porras@gmail.com

A. Lasanta
Dept. of Algebra
University of Granada
Ceuta, Spain
alasanta@ugr.es

C. Cano
Dept. Computer Science and A.I.
University of Granada
Granada, Spain
carloscano@ugr.es

***Abstract*—In this text, we present a study on how to tackle the *de novo* genome assembly problem using quantum annealing. We formulate the problem as a Quadratic Unconstrained Binary Optimization (QUBO) problem and embed a QUBO model into a Dwave quantum annealer. A series of experiments using simulated annealing and D-Wave quantum systems are presented.**

***Index Terms*—quantum annealing, quantum optimization, QUBO, *de novo* genome assembly**

I. Introduction

Quantum annealers are a type of quantum computer focused on optimization. These quantum annealers have proven to be empirically useful for several real-world applications.

The *de novo* Genome Assembly problem is *one of the most important and difficult problems in Bioinformatics* [1] and deals with the reconstruction of a whole DNA genome from small redundant fragments of sequences (reads) generated at random positions of the original genome. This problem has been proven to be NP-complete [2] so current algorithms rely on heuristics to find suboptimal solutions in reasonable time. This problem has a strong impact in the DNA sequencing efforts worldwide and is key to develop Precision Medicine [3].

Some authors have already tacked the formulation of the Genome Assembly problem in a Quantum Annealer ([4] and [5]). Inspired by the work of these authors, we set out to first gain a deep understanding of quantum annealers and then tackle the *de novo* genome assembly problem using D-Wave's quantum annealers. This work is a summary of the results obtained in an undergraduate final thesis published at the University of Granada: https://github.com/Ocete/TFG [6]. In this summary, we briefly describe how to tackle the genome assembly problem with the use of quantum annealers by formulating the problem into a QUBO model and perform experiments on both simulators and a D-wave quantum annealer.

II. The Genome Assembly Problem

There are different formulations to address the *de novo* assembly problem which can be represented as combinatorial optimization problems on graphs: Overlap-Layout Consensus (OLC) methods, de Bruijn graph (DBG) methods, string graphs, greedy and hybrid methods (for a review see [3]). For our purposes, we will focus on OCL methods. In these methods, each node in the graph represents a different DNA read. Directed edges are associated a weight depending on how well the connected reads are stitched together. For the computation of the weights we will consider the length of the overlap between the corresponding reads without any mismatch, with a change of sign. For example, given the reads $r_1 = AATT$ and $r_2 = TTCC$, the perfect stitching will produce $AATTCC$, so the overlap between both reads is 2, giving a weight of -2.

A Hamiltonian path in our overlap graph will represent a series of reads in a certain order. By minimizing the total cost of our Hamiltonian path, we maximize the overlap between reads, resulting in the shortest possible final chain. This is exactly the same as solving the Travelling Salesman problem (TSP) associated with our overlap graph.

Let illustrate this process with a toy example from [4]. Suppose we are given the following four reads:

- $r_0 = ATGGCGTGCA$

This work is supported by Grant PID2021-128970OA-I00 funded by MCIN/AEI/10.13039/501100011033/FEDER.

- $r_1 = GCGTGCAATG$
- $r_2 = TGCAATGGCG$
- $r_3 = AATGGCGTGC$

If we compute the overlap between each pair of reads we obtain the overlap graph from figure 1.

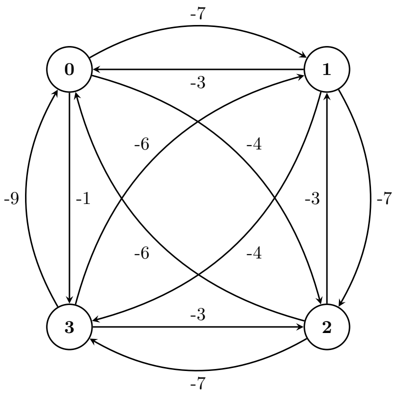
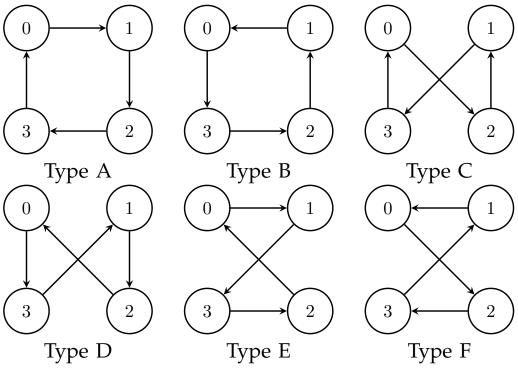


Fig. 1: OLC graph between the reads r_0, r_1, r_2, r_3 from the example. The node number corresponds to the read number.

A TSP would derive six different types of cycles in this graph (see Table I).



Type A

Type B

Type C

Type D

Type E

Type F

TABLE I: Types of cycles in a 4-node graph.

It is easy to check that the Hamiltonian cycle that minimizes the cost in this case is a cycle of type A. This cycle represents the solution of the TSP with the minimum cost (-30). We can easily compute the sequence resulting from this assembly by traversing the graph and weaving the reads together - if we start the cycle in r_0 this would result in the sequence: $ATGGCGTGCAATGGCGTGC$.

Therefore, given the DNA reads, we will : (a) compute the overlap graph using a distance between the reads, (b) formulate the corresponding TSP into a QUBO model (c) Obtain the Hamiltonian path of minimum cost on the graph by either using a simulated annealer or the D-Wave quantum annealer, (d) finally, we traverse the cycle and build the resulting genome sequence.

III. QUBO problem formulation

Let $B = \{0, 1\}$ and $f_Q : \mathbb{B}^n \longrightarrow \mathbb{R}$ be a quadratic polynomial over binary variables:

$$f_Q(x) = \sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij} x_i x_j$$

where $x_i \in \mathbb{B}$ for $i \in \{1, \cdots, n\}$ and coefficients: $q_{ij} \in \mathbb{R}$ for $1 \leq j \leq i \leq n$. A quadratic unconstrained binary optimization problem (QUBO) requires to find the binary vector x' minimizing:

$$f_Q : x' = \underset{x \in \mathbb{B}^n}{\arg\min}\ f_Q(x)$$

In matrix notation: $f_Q(x) = x^T Q x$, where Q is a symmetric matrix of size $n \times n$ containing coefficients q_{ii} in its diagonal and $q_{ij}/2$ in position (i, j) if $i \neq j$.

To formulate the genome assembly as a QUBO, we start from an OLC graph G containing nodes $v_1, \ldots, v_n$, and weights $w_{i,j}$ for every $i, j \in [1, \ldots, n]$, and our goal is to find a hamiltonian path of minimum total weight.

We define n^2 binary variables $x_{i,j}$. $x_{i,j}$ is set to 1 if and only if the path walks through node i at time point j. We also need to add more restrictions in order to obtain valid hamiltonian walks: we can only walk through one node at a specific time point j (position restriction) and we never walk through a node twice (repetition restriction). The final optimization function is the following:

$$\begin{aligned}\text{Minimize } & \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} w_{i,j} \sum_{p=0}^{n-1} x_{i,p} x_{j,p+1} \\ & + a \sum_{i=0}^{n-1} \sum_{p=0}^{n-1} x_{i,p} && \text{(self-bias)} \\ & + b \sum_{i=0}^{n-1} \Big(\sum_{p=0}^{n-1} x_{i,p} - 1 \Big)^2 && \text{(repetition)} \\ & + c \sum_{p=0}^{n-1} \Big(\sum_{i=0}^{n-1} x_{i,p} - 1 \Big)^2 && \text{(position)}\end{aligned} \tag{1}$$

Where a, b, c are penalty parameters for each restriction. In the experimentation, we set $a = -X$ and $b = c = X$ for different values of X.

IV. Experimentation and Preliminary Results

In order to configure the samplers and submit jobs to D-Wave quantum annealers we used the *D-Wave Ocean Software*, a suite of tools provided by D-Wave to use their quantum systems [7]. Using python and the provided packages we connected to *Leap*, a 'quantum' cloud service that provided access to Dwave's quantum computers [8].

To solve this problem we used the proposed QUBO formulation for the TSP and we first run a Simulated Annealer from *D-Wave Ocean Software*. We ran the same experiment 10.000 times with parameter $X = 1.6$ to get the distribution of solutions shown in Table II. Results showed that the best type of cycle (type A) agglomerates most of the samples, 37.22%. In the second place, the four types of cycles that have the exact same energy (C, D, E and F) also have similar number of samples. It is worth mentioning that there was not a single sample that encoded an invalid cycle. This means that the penalties values used for the experiment (1.6 for multi-location and repetition, and -1.6 for self-bias) are working to prevent invalid cycles.

Cycle type	Freq. (X=1.6)	Energy
Type A	3722	-7.9811
Type C	1474	-7.4541
Type D	1469	-7.4541
Type F	1458	-7.4541
Type E	1431	-7.4541
Type B	446	-6.927

TABLE II: Distribution of solutions for 10.000 runs on D-wave's Simulated Annealer.

We repeated the experimentation with the D-wave quantum annealer with 2000 qubits. For this purpose, we embedded the graph of solutions into the Chimera topology with D-Wave Ocean Software [7]. We tested the performance of the system with two different penalty values $X = 1.6$ (as in simulated annealing) and $X = 1.5$. Results are shown in table III.

Cycle type	Freq. (X=1.5)	Freq. (X=1.6)	Energy
Type A	131	50	-7.9811
Type C	124	46	-7.4541
Type D	40	91	-7.4541
Type F	55	117	-7.4541
Type E	112	118	-7.4541
Type B	99	64	-7.4541
Invalid	9221	9194	$>$ -5.6433

TABLE III: Distribution of solutions for 10.000 runs on D-Wave 2000Q.

Results obtained in the D-wave 2000Q quantum annealer showed a percentage of valid solutions below 10% of the solutions found for both penalty parameters (see Table III). Also, the distribution of valid solutions was not the expected for penalty parameter $X = 1.6$, since type A cycles (solutions with the lowest energy) were not the most frequent valid solutions found. Penalty parameter $X = 1.5$ provided a similar percentage of invalid solutions but significantly different distribution of valid solutions, as can be seen in the table III. Further research is being performed to identify potential causes for this behaviour to increase the number of valid solutions, explore new D-wave topologies and perform scalability studies for this problem.

Acknowledgment

This work is supported by Grant PID2021-128970OA-I00 funded by MCIN/AEI/10.13039/501100011033/FEDER.

References

[1] A.M. Phillippy. "New advances in sequence assembly." Genome research 27, 5. 2017.

[2] P. Medvedev. "Modeling biological problems in computer science: a case study in genome assembly." Briefings in bioinformatics 20, 4. p. 1376-1383. 2019.

[3] J. Sohn, and N. Jin-Wu. "The present and future of de novo whole-genome assembly." Briefings in bioinformatics 19, 1. p. 23-40. 2018.

[4] A. Sarkar, et al. "QuASeR: Quantum Accelerated de novo DNA sequence reconstruction." Plos one 16, 4. 2021.

[5] K. Nalecz-Charkiewicz and R. M. Nowak. "Algorithm for DNA sequence assembly by quantum annealing." BMC bioinformatics 23, 1. 2022.

[6] J. A. Álvarez-Ocete. Github repository https://github.com/Ocete/TFG.

[7] D-Wave Ocean Software Documentation. https://docs.ocean.dwavesys.com/en/stable/index.html.

[8] D-Wave Leap https://cloud.dwavesys.com/leap.

Quantum Algorithms for WMC, MPE and MAP

Fabrizio Riguzzi
Dipartimento di Matematica e Informatica, *University of Ferrara*, Ferrara, Italy, fabrizio.riguzzi@unife.it

***Abstract*—In Weighted Model Counting (WMC) we are given a propositional formula and a weight for each literal and we want to compute the sum of the weights of the models of the formula. In Most Probable Explanation (MPE) we seek the model with the highest weight while in Maximum A Posteriori (MAP) we look for the state of a subset of variables that maximizes the sum of the weights of the models that agree on that state. WMC, MPE and MAP find interesting applications in inference for graphical models.**

In this paper, we propose QWMC, QMPE and QMAP, quantum algorithms for performing WMC, MPE and MAP, respectively. They are all based on the quantum search/quantum model counting algorithms that are modified to take into account the weights.

In the black box model of computation, where we can only query an oracle for evaluating the Boolean function given an assignment, QWMC solves the problem approximately with a complexity of $\Theta(2^{\frac{n}{2}})$, where n is the number of Boolean variables, while classically the best complexity is $\Theta(2^n)$, thus achieving a quadratic speedup. QMPE and QMAP require $O(1/\sqrt{WMC})$ oracle calls, where WMC is the normalized between 0 and 1 weighted model count of the formula, while a classical algorithm has a complexity of $\Omega(1/WMC)$, again obtaining a quadratic speedup.

***Index Terms*—Quantum Search, Quantum Model Counting, Weighted Model Counting, Most Probable Explanation, Maximum A Posteriori**

I. Introduction

Weighted Model Counting (WMC) [1] generalizes model counting by assigning weights to literals: the aim is to compute the sum of the weights of the models of a propositional formula, where the weight of a model is given by the product of the weights of its literals. WMC was successfully applied to the problem of performing inference in graphical models [1], [2].

The Most Probable Explanation (MPE) [3] problem involves finding an assignment to all variables that satisfies a Boolean formula and has the maximum weight. The related Maximum A Posteriori (MAP) problem means finding an assignment of a subset of the variables such that the sum of the weights of the models of the formula that agree on the assignment is maximum.

In this paper we propose to use quantum computing for performing WMC, MPE and MAP. We call QWMC, QMPE and QMAP the quantum algorithms for performing WMC, MPE and MAP respectively. These quantum algorithms are based on the method of quantum model counting [4], [5], which we modify to take into account weights. Quantum model counting in turn is based on quantum search using Grover's algorithm [6]–[8] and on quantum phase estimation [9], the latter using the quantum Fourier transform [10]. In particular, the proposed algorithms modify the algorithms for unweighted counting and search by replacing the Hadamard gates with rotation gates, with the rotations depending on the weights. Moreover, QMAP differs from unweighted quantum search also because only a subset of the qubits is measured.

QWMC, QMPE and QMAP work under a black box computation model where we only have the possibility of querying an oracle giving the value of the formula for an assignment of the propositional variables. QWMC takes $\Theta(2^{\frac{n}{2}})$ oracle calls, while any classical algorithm takes $\Theta(2^n)$ oracle calls, thus achieving a quadratic speedup. Similarly, QMPE and QMAP take $O(1/\sqrt{WMC})$ oracle calls, where WMC is the weighted model count normalized between 0 and 1, while a classical algorithm has a complexity of $\Omega(1/WMC)$. QMPE and QMAP can also be seen as methods for sampling assignments with a probability distribution given by the weights.

This article is a summary of [11] (under submission) that in turn extends [12] by adding the QMPE and QMAP algorithms. All the proofs of the results in this paper can be found in [11].

Section II introduces the WMC, MPE and MAP problems. Section III discusses the QWMC algorithm, while Section IV presents QMPE and QMAP and Section V concludes the paper.

II. WMC, MPE and MAP Problems

Let X be a vector of n Boolean variables $[X_1, \dots, X_n]$ and let x be an assignment of values to X, i.e., a vector of n Boolean values $[x_1, \dots, x_n]$. We call x a *world.* Consider a propositional logic formula ϕ over X. If an assignment x of variables X makes formula ϕ evaluate to true, we write $x \models \phi$ and we say that x *satisfies* ϕ. We can also see ϕ as a function from $\{0,1\}^n$ to $\{0,1\}$ and express that x makes ϕ evaluate to true by $\phi(x) = 1$.

Given a formula ϕ in propositional logic over literals L (Boolean variables or their negation), and a weight function $w : L \to R^{\geq 0}$, the weighted model count (WMC) is defined as: $WMC(\phi, w) = \sum_{x:x\models\phi} W_x$, where $W_x = \prod_{l\in x} w(l)$ is the weight of model x according to weight function w.

The most probable state (MPE) of the variables is $MPE(\phi, w) = \operatorname{argmax}_{x:x\models\phi} W_x$.

Given a set of query variables Q, the most probable state of the query variables (MAP) is $MAP_Q(\phi, w) = \operatorname{argmax}_q \sum_{y:qy\models\phi} W_{qy}$ where qy is a world where variables in Q take value q and variables in $Y = X \setminus Q$ take value y.

This research was partly supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215 and by the 'National Group of Computing Science (GNCS-INDAM)".

S	R	W	ϕ	$weight$
0	0	0	1	$0.45 \cdot 0.7 \cdot 0.3 = 0.0945$
0	0	1	1	$0.45 \cdot 0.7 \cdot 0.7 = 0.2205$
0	1	0	0	$0.45 \cdot 0.3 \cdot 0.3 = 0.0405$
0	1	1	1	$0.45 \cdot 0.3 \cdot 0.7 = 0.0945$
1	0	0	0	$0.55 \cdot 0.7 \cdot 0.3 = 0.1155$
1	0	1	1	$0.55 \cdot 0.7 \cdot 0.7 = 0.2695$
1	1	0	0	$0.55 \cdot 0.3 \cdot 0.3 = 0.0495$
1	1	1	0	$0.55 \cdot 0.3 \cdot 0.7 = 0.1155$

TABLE I: Worlds for Example 1.

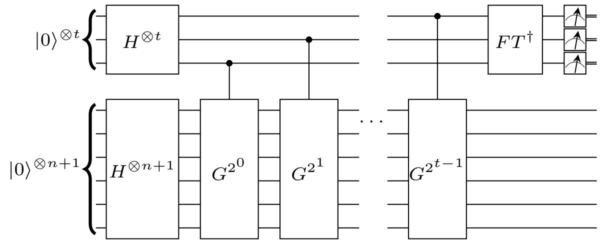


Fig. 1: Circuit for quantum counting.

Example 1. *Consider the Boolean formula $\phi = (\neg S \vee W) \wedge (\neg R \vee W) \wedge (\neg S \vee \neg R)$. Suppose the weights of literals are $w(S) = 0.55$, $w(\neg S) = 0.45$, $w(R) = 0.3$, $w(\neg R) = 0.7$, $w(W) = 0.7$ and $w(\neg W) = 0.3$. Table I shows the worlds together with the weight of each world. The WMC of ϕ is thus $WMC(\phi, w) = 0.0945 + 0.2205 + 0.0945 + 0.2695 = 0.679$.*

The MPE is $MPE(\phi, w) = [1, 0, 1]$ and the MAP of query variables S and W is $MAP_{SW}(\phi, w) = [0, 1]$.

III. Quantum Weighted Model Counting

The circuit for quantum counting [4], [5] is shown in Figure 1, where the second register has an extra bit X_{n+1} and formula ϕ is changed to $\phi \wedge X_{n+1}$. This is done in order to ensure that the model count is smaller than half the number of all assignments.

The circuit for performing quantum weighted model counting differs from the one in Figure 1 because the Hadamard operations applied to the lower register are replaced by rotations $R_y(\theta_i)$ where i is the qubit index except for the extra qubit for which the Hadamard operator is kept. Overall the gate $H^{\otimes n+1}$ is replaced by gate Rot shown in Figure 2. θ_i is computed as $\theta_i = 2 \arccos \sqrt{1 - w_i}$ where $w_i = w(X_i)$ and we suppose that the literal weights sum to 1, i.e., that $w(X_i) + w(\neg X_i) = 1$ for all bits X_i. The general case can be treated by using normalization.

The rotations prepare the state

$$|\varphi\rangle \quad = \quad \bigotimes_{i=1}^{n} (\sqrt{1 - w_i}\,|0\rangle + \sqrt{w_i}\,|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad (1)$$

If we define the normalized states

$$|\gamma\rangle \quad = \quad \frac{1}{\sqrt{\frac{1+\sum_{x:\phi(x)=0} W_x}{2}}} \left(\sum_{x} \sqrt{\frac{W_x}{2}}\,|x0\rangle + \sum_{x:\phi(x)=0} \sqrt{\frac{W_x}{2}}\,|x1\rangle \right)$$

$$|\delta\rangle \quad = \quad \frac{1}{\sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}}} \sum_{x:\phi(x)=1} \sqrt{\frac{W_x}{2}}\,|x1\rangle\,,$$

then $|\varphi\rangle$ can be expressed as

$$|\varphi\rangle = \cos\theta/2\,|\gamma\rangle + \sin\theta/2\,|\delta\rangle \qquad (2)$$

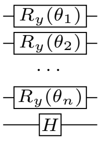


Fig. 2: Circuit for gate Rot.

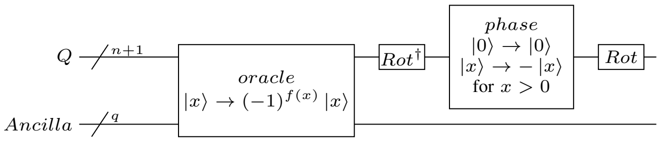

Q
$n+1$
$Ancilla$
q
$oracle$
$|x\rangle \to (-1)^{f(x)}\,|x\rangle$
$Rot^\dagger$
$phase$
$|0\rangle \to |0\rangle$
$|x\rangle \to -\,|x\rangle$
for $x > 0$
Rot

Fig. 3: Weighted Grover operator WG.

where

$$\cos\theta/2 \quad = \quad \sqrt{\frac{1 + \sum_{x:\phi(x)=0} W_x}{2}} \qquad (3)$$

$$\sin\theta/2 \quad = \quad \sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}} \qquad (4)$$

Gate Rot replaces $H^{\otimes n+1}$ also in the Grover operator G that becomes the weighted Grover operator WG shown in Figure 3[1]: The application of the weighted Grover operator rotates $|\varphi\rangle$ in the space spanned by $|\gamma\rangle$ and $|\delta\rangle$ by angle θ so $e^{i\theta}$ and $e^{i(2\pi-\theta)}$ are the eigenvalues of WG. θ can be found by quantum phase estimation.

We thus obtain

$$WMC(\phi, w) = \sum_{x:\phi(x)=1} W_x = 2\sin^2(\theta/2)$$

Theorem 1. *QWMC on n bits requires $\Theta(\sqrt{N})$ oracle calls to bound the error to $2^{-\frac{n+1}{2}}$ with probability $11/12$ using $t = \lceil n/2 \rceil + 5$ bits.*

Theorem 2. *The complexity of any classical algorithm for estimating $WMC(\phi, w)$ with a probability of at least $3/4$ within an accuracy of $2^{-\lceil \frac{n}{2} \rceil}$ is $\Omega(N)$ oracle calls.*

Therefore QWMC offers a quadratic speedup over classical computation.

IV. Quantum MPE and MAP

As for QWMC, suppose that the literal weights sum to 1. We perform quantum MAP by modifying the circuit for quantum search [6]–[8] as we modified the circuit for quantum counting in order to do QWMC. The circuit for performing quantum MAP (QMAP) is shown in Figure 4 and, similarly to the circuit for performing QWMC, the Grover operator is replaced by the Weighted Grover operator WG shown in Figure 3 and the initial $H^{\otimes n+1}$ gate is replaced by gate Rot of Figure 2. We suppose that the query bits come first and there are l of them, let us call Q the query bits, while the non query bits are called

[1] The previous paper [12] erroneously used G in place of WG

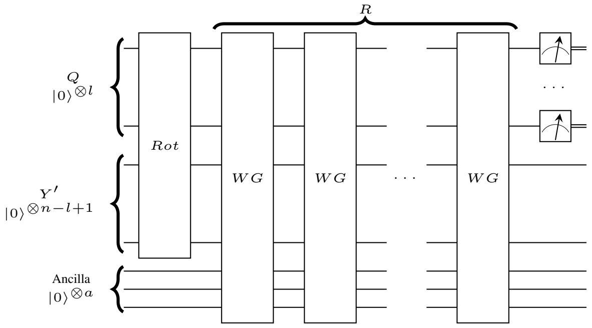


Fig. 4: The complete QMAP circuit.

Y and there are $n-l$ of them. Let Y' be Y with the extra qubit X_{n+1}, so overall Y' has $n-l+1$ bits.

The rotations prepare the state in Eq. 1 that can be expressed as in Eq. 2. The application of the weighted Grover operator rotates $|\varphi\rangle$ in the space spanned by $|\gamma\rangle$ and $|\delta\rangle$ by angle θ of Eq. 3.

If we perform R rotations where R is

$$R \quad = \quad \mathrm{CI}\left(\frac{\arccos\frac{\sum_{x:\phi(x)=1}W_x}{2}}{2\arcsin\frac{\sum_{x:\phi(x)=1}W_x}{2}}\right)$$

and CI indicates the closest integer, we maximize the probability of measuring a state of $|\delta\rangle$, i.e., a state x such that $\phi(x)=1$. R can be computed with QWMC.

Now we measure only the query qubits. Suppose that the results of the rotation brings $|\varphi\rangle$ exactly to $|\delta\rangle$. Let's apply the measurement $\{M_m = |q_m\rangle\langle q_m|\}$ to system Q where q_m is one of the computational basis state for Q. Then

$$P(m) \quad = \quad \frac{\sum_{y:\phi(q_m y)=1}W_{q_m y}}{0.5\cdot WMC(\phi,w)}$$

So this algorithm returns configurations of query bits q_m with a probability that is proportional to $\sum_{y:\phi(q_m y)=1}W_{q_m y}$.

Consider now the case that the final state is $a|\gamma\rangle + b|\delta\rangle$ instead of simply $|\delta\rangle$. As for Grover's algorithm, the error on the angle in the final state is at most $\theta/2$ so the amplitude a of $|\gamma\rangle$ in case of maximum error is $a = \cos(\pi/2 \pm \theta/2) = \mp\sin\theta/2 = \mp\sqrt{\frac{\sum_{x:\phi(x)=1}W_x}{2}}$ So the probability of measuring a value from γ is at most $\frac{\sum_{x:\phi(x)=1}W_x}{2}$ and the probability of measuring a value from δ (a solution) is at least $1-\frac{\sum_{x:\phi(x)=1}W_x}{2}$.

In the case in which $Y=[X_{n+1}]$, we are performing MPE inference and we call the algorithm QMPE.

We can use this algorithm as a probabilistic algorithm: we execute it for o iterations. At each iteration, we measure all Q bits obtaining q_m. After o iterations, we return the value that was found most frequently.

Theorem 3. *The number of applications of WG (and thus of oracle calls) required to maximize the probability of measuring one of the solutions in QMAP and QMPE is* $O(\frac{1}{\sqrt{\sum_{x:\phi(x)=1}W_x}})$.

Theorem 4. *Any classical probabilistic algorithm for solving MPE or MAP under the black box model of computation takes* $\Omega(\frac{1}{\sum_{x:\phi(x)=1}W_x})$ *oracle queries.*

Thus QMPE and QMAP achieve a quadratic speedup.

V. Conclusions

We have presented quantum algorithm QWMC, QMPE and QMAP for solving WMC, MPE and MAP respectively. We have shown that QWMC has a complexity of $\Theta(2^{\frac{n}{2}})$ evaluations of the Boolean formula, while QMPE and QMAP solve their respective problems with a complexity of $O(1/\sqrt{WMC})$. We have also shown that if we consider the Boolean formula as a black box that we can only query asking for the value of the function given the inputs, these algorithms provide quadratic speedups over classical algorithms with the same limitation. The black box setting may be of interest when the Boolean formula is given by a quantum physical system of which we don't know the internals. In that case the quantum algorithms can plug in the system directly, improving over classical algorithms.

Acknowledgment

The author would like to thank Mariia Mykhailova for interesting discussions on the topic of this paper.

This work has been partially supported by the Spoke 1 "FutureHPC & BigData" of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU), by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No. 952215, and by the "National Group of Computing Science (GNCS-INDAM)".

References

[1] T. Sang, P. Beame, and H. A. Kautz, "Performing bayesian inference by weighted model counting," in *20th National Conference on Artificial Intelligence (AAAI 2005)*, (Palo Alto, California USA), pp. 475–482, AAAI Press, 2005.

[2] M. Chavira and A. Darwiche, "On probabilistic inference by weighted model counting," *Artif. Intell.*, vol. 172, no. 6-7, pp. 772–799, 2008.

[3] T. Sang, P. Beame, and H. A. Kautz, "A dynamic approach for MPE and weighted MAX-SAT.," in *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 173–179, AAAI Press/IJCAI, 2007.

[4] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *Fortschritte der Physik: Progress of Physics*, vol. 46, no. 4-5, pp. 493–505, 1998.

[5] G. Brassard, P. Høyer, and A. Tapp, "Quantum counting," in *25th International Colloquium on Automata, Languages and Programming (ICALP 1998)* (K. G. Larsen, S. Skyum, and G. Winskel, eds.), vol. 1443 of *Lecture Notes in Computer Science*, pp. 820–831, Springer, 1998.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, (New York, NY, USA), pp. 212–219, ACM Press, 1996.

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search," 1996. arXiv preprint quant-ph/9605043.

[8] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical review letters*, vol. 79, no. 2, p. 325, 1997.

[9] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, "Quantum algorithms revisited," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 339–354, 1998.

[10] D. Coppersmith, "An approximate fourier transform useful in quantum factoring," 2002. arXiv preprint quant-ph/0201067.

[11] F. Riguzzi and M. Mykhailova, "Quantum algorithms for WMC, MPE and MAP." Available at SSRN https://ssrn.com/abstract=4169880.

[12] F. Riguzzi, "Quantum weighted model counting," in *24th European Conference on Artificial Intelligence (ECAI 2020)* (G. De Giacomo, A. Catala, B. Dilkina, M. Milano, S. Barro, A. Bugarín, and J. Lang, eds.), (Amsterdam, Berlin, Washington DC), pp. 2640–2647, IOS Press, 2020.

Encoding Extension-based Problems in Argumentation to QUBO

1st Marco Baioletti
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
marco.baioletti@unipg.it

2nd Francesco Santini
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
francesco.santini@unipg.it

***Abstract*—We propose an encoding of different NP-complete problems in extension-based Abstract Argumentation into *Quadratic Unconstrained Binary Optimisation* problems. The obtained formulation can be then solved by using quantum annealers, as already accomplished in preliminary tests.**

***Index Terms*—Abstract Argumentation, Quadratic Unconstrained Binary Optimisation.**

I. Introduction

Formal Argumentation can be credited to the pioneering works in logics of Pollock [1] and other authors. The premise is that (non-monotonic) reasoning can be done by creating and assessing arguments, which are made up of several justifications for a claim's validity. Arguments differ from proofs in that they are defeasible: whether a claim can be accepted depends not only on whether an argument supporting it exists, but also on whether potential opposing arguments exist, which can then be contested by attacking arguments, and so on.

The *Abstract Argumentation* theory of Dung [2] provides the foundation for a lot of current argumentation research. An argumentation framework, which is essentially a directed graph with the arguments represented as nodes and the attack relation represented by arrows, is the key idea in this study. An analysis of the question of which set(s) of arguments can be accepted, given such a network, leads to the definition of argumentation semantics.

The argumentation is said to be "abstract" because arguments have no internal structures and there is no specification of what an argument or an attack is. It is however enough to represent conflict among information, and it has connections with well-founded semantics of logic programs [2].

In regard to Abstract Argumentation, a number of proposals have been made in the literature; in Section II-A we summarise the background on some NP-complete problems that are related to *extensions*, which are sets of arguments that can survive the conflict together and thus represent collectively a reasonable position an autonomous reasoner might take.

A *Quadratic Unconstrained Binary Optimisation* problem [3] (*QUBO*), is a mathematical formulation that encompasses a wide range of critical *Combinatorial Optimisation* problems. QUBO problems are NP-complete, and a vast literature is dedicated to approximate solvers based on heuristics or meta-heuristics, such as *simulated annealing* approaches (*SA*), *tabu-serch*, *genetic algorithms* or *evolutionary computing* [4]. *Quantum* annealers and Fujitsu's *digital annealers*[1] can be used to find global minima by using quantum *fluctuations*. QUBO models are at the heart of experimentation with quantum computers built by D-Wave Systems.[2]

In this paper, we propose encodings of different Abstract Argumentation problems that are NP-complete problems as well. The results, here summarised, are new with respect to the pioneering work in [5]. As a general result, our goal is to deepen the research line opened there with the purpose of modelling and solving a wide range of these kinds of reasoning problems, with the help of quantum machines.

II. Background

A. Argumentation.

An *Abstract Argumentation Framework* (AF, for short) [2] is a tuple $\mathcal{F} = (\mathsf{A}, R)$ where A is a set of arguments and R is a relation $R \subseteq \mathsf{A} \times \mathsf{A}$. For two arguments $a, b \in \mathsf{A}$ the relation aRb means that argument a *attacks* argument b. An argument $a \in \mathsf{A}$ is *defended* by $S \subseteq \mathsf{A}$ (in $\mathcal{F}$) if for each $b \in \mathsf{A}$ such that bRa there is some $c \in S$ such that cRb. A set $E \subseteq \mathsf{A}$ is *conflict-free* ($\mathbf{cf}$ in $\mathcal{F}$) if and only if there are no $a, b \in E$ with aRb. E is *admissible* ($\mathbf{ad}$ in $\mathcal{F}$) if and only if it is conflict-free and each $a \in E$ is defended by E. Finally, the range of E in $\mathcal{F}$, i.e., $E^+_\mathcal{F}$, collects the same E and the set of arguments attacked by E: $E^+_\mathcal{F} = E \cup \{a \in \mathsf{A} \mid \exists b \in E : bRa\}$.

The *collective acceptability* of arguments depends on the definition of different *semantics* [2]. Semantics determine sets of jointly acceptable arguments, called *extensions*, by mapping each $\mathcal{F} = (\mathsf{A}, R)$ to a set $\sigma(\mathcal{F}) \subseteq 2^\mathsf{A}$, where 2^A is the power set of A, and σ parametrically stands for any of the considered semantics. The extensions under complete, preferred, stable, and semi-stable semantics are defined as follows. Given $\mathcal{F} = (\mathsf{A}, R)$ and a set $E \subseteq \mathsf{A}$, $E \in \mathbf{co}(\mathcal{F})$ iff E is admissible in $\mathcal{F}$ and if $a \in \mathsf{A}$ is defended by E in $\mathcal{F}$ then $a \in E$; $E \in \mathbf{pr}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and there is no $E' \in \mathbf{co}(\mathcal{F})$ s.t. $E' \supset E$; $E \in \mathbf{sst}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and there is no $E' \in \mathbf{co}(\mathcal{F})$ s.t. $E'^+_\mathcal{F} \supset E^+_\mathcal{F}$; $E \in \mathbf{st}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and $E^+_\mathcal{F} = \mathsf{A}$,

Figure 1 shows an AF with five arguments and five attacks. Given $\mathcal{F}$, the set of complete extensions is $\mathbf{co}(\mathcal{F}) =$

[1] Fujitsu's digital annealer: https://www.fujitsu.com/global/services/business-services/digital-annealer/.

[2] D-Wave webiste: https://www.dwavesys.com.

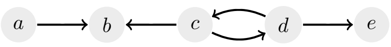


Fig. 1: An example of WAAF.

	Ver-σ	DC-σ	DS-σ	Ex-σ	NE-σ
Conflict-free	in L	in L	triv.	triv.	in L
Admissible	in L	**NP-c**	triv.	triv.	**NP-c**
Complete	in L	**NP-c**	P-c	triv.	**NP-c**
Preferred	coNP-c	**NP-c**	$\prod_2^P$-c	triv.	**NP-c**
Semi-stable	coNP-c	$\sum_2^P$-c	$\prod_2^P$-c	triv.	**NP-c**
Stable	in L	**NP-c**	coNP-c	**NP-c**	**NP-c**

TABLE I: The complexity of some problems in Abstract Argumentation.

$\{\{a\}, \{a, d\}, \{a, c, e\}\}$, while $\mathbf{st}(\mathcal{F}) = \{\{a, d\}, \{a, c, e\}\}$ is the set of stable extensions, for example.

We now report the definition of six well-known decision problems in Abstract Argumentation. *Credulous acceptance* $\mathbf{DC}$-σ: given $\mathcal{F} = (\mathsf{A}, R)$ and an argument $a \in \mathsf{A}$, is a contained in some $E \in \sigma(\mathcal{F})$? Sceptical acceptance $\mathbf{DS}$-σ: given $\mathcal{F} = (\mathsf{A}, R)$ and an argument $a \in \mathsf{A}$, is a contained in all $E \in \sigma(\mathcal{F})$? *Verification of an extension* $\mathbf{VER}$-σ: given $\mathcal{F} = (\mathsf{A}, R)$ and a set of arguments $E \subseteq \mathsf{A}$, is $E \in \sigma(\mathcal{F})$? *Existence of an extension* $\mathbf{EX}$-σ: given $\mathcal{F} = (\mathsf{A}, R)$, is $\sigma(\mathcal{F}) \neq \varnothing$? *Existence of non-empty extension* $\mathbf{NE}$-σ: given $\mathcal{F} = (\mathsf{A}, R)$, does there exist $E \neq \varnothing$ such that $E \in \sigma(\mathcal{F})$?

In addition, the work in [6] presents the task of *extension enforcement*: we consider the objective to change the attack relationship R of a framework $\mathcal{F} = (\mathsf{A}, R)$ such that a given set $T \subseteq \mathsf{A}$ becomes (a subset of) an extension under a given semantics σ. In this case, we say that the enforcement is *argument-fixed*, since only the attack relationship can be modified. *Strict enforcement* is satisfied if T is a σ-extension, while in *non-strict enforcement* T is only required to be a subset of a σ-extension. If we consider the Hamming distance of the changes, i.e., $|R\Delta R'| = |R \setminus R'| + |R' \setminus R|$, in [6] the authors impose a threshold $|R\Delta R'| \leq k$ as a further parameter of these problems. The complexity of some of these problems is reported in Tab. II.

In this paper, as proposed in [7], we look at the problem from an optimisation point of view:

Definition 1 ([7]). *Given $\mathcal{F} = (\mathsf{A}, R)$, $T \subseteq \mathsf{A}$, and semantics σ, strict extension enforcement is an optimisation problem where to goal is to find $\mathcal{F}^* = (\mathsf{A}, R^*)$ s.t.:*

$$R^* \in \operatorname*{argmin}_{R' \in enfst(\mathcal{F}, T, \sigma)} |R\Delta R'|$$

where $enfst(\mathcal{F}, T, \sigma) = \{R' | \mathcal{F}' = (\mathsf{A}, R'), T \in \sigma(\mathcal{F}')\}$. Similarly, we can define the same problem by considering non-strict enforcement (by defining enfnst).

B. QUBO

Quadratic Unconstrained Binary Optimisation (*QUBO*) [8] is a form of optimisation problems encompassing e.g. SAT/Constraint/(0,1)-ILP, which recently gained great popularity because of fast solvers and dedicated computing devices, such as quantum and digital annealers. A QUBO problem is defined in terms of n binary variables $x_1, \ldots, x_n$ and a $n \times n$ upper-diagonal matrix Q and consists in minimising the function $f(x) = \sum_{i=1}^{n} Q_{i,i} x_i + \sum_{i<j}^{n} Q_{i,j} x_i x_j$. The diagonal terms $Q_{i,i}$ are the linear coefficients and the non-zero off-diagonal terms $Q_{i,j}$ are the quadratic coefficients. This can be expressed more concisely as $\min_{x \in \{0,1\}^n} x^T Q x$, where x^T denotes the transpose of the vector x. The formulation of problem in QUBO consists in *i)* find a binary representation for the solutions, *ii)* define a penalisation function, which penalises unfeasible solutions (i.e., violating a constraint).

σ	strict	non-strict
Admissible	P	**NP-c**
Complete	**NP-c**	**NP-c**
Preferred	$\sum_2^P$-c	**NP-c**
Stable	P	**NP-c**
Grounded	**NP-c**	**NP-c**

TABLE II: The complexity of extension enforcement [7].

III. Encoding of problems

A. A Formulation in QUBO of Acceptance Tasks

In [5] we proposed for the first time an encoding of two well-known NP-complete problems in Abstract Argumentation as QUBO problems: $\mathbf{DC}$-σ and **Exists**-$\sigma^{\neg\varnothing}$, while the considered semantics was only **co**. Moreover, in [5] we solved this problem on some frameworks by directly implementing them by using the D-Wave Ocean SDK. We both used a simulated annealing algorithm and a real quantum annealer provided by the *LeapTM Quantum Cloud Service*.[3]

With respect to [5], by continuing on this research line, we have extended the encoding to all classical NP-complete problems highlighted in bold in Tab. I. Moreover, we have empirically validated all the encodings by comparing the obtained results with the simulated annealing algorithm against *ConArg* [9], an exact solver using *Constraint Programming*.

We present a QUBO encoding of some of the Abstract Argumentation problems in Sect. II. We assign to each argument an index, hence $\mathsf{A} = \{a_1, \ldots, a_n\}$, where n is the number of arguments. We use a set of n binary variables $x_1, \ldots, x_n$ to represent a set E of arguments: $a_i \in E$ if and only if $x_i = 1$. We denote by $\underline{x}$ the tuple $(x_1, \ldots, x_n)$ and by $\mathbf{x} \in \{0, 1\}^n$ a vector of possible values for $x_1, \ldots, x_n$. Each semantics σ will be associated with a quadratic penalisation function (or *Pfunction* for short) P_σ such that P_σ assumes its minimum value at $\mathbf{x}$ if and only if the corresponding set $E = \{a_i \in \mathsf{A} : x_i = 1\}$ is an extension valid for σ.

Most of the argumentation semantics require admissible sets. Hence, we define a Pfunction P_{adm} which enforces this property. P_{adm} is the sum of four terms and contains new additional variables. The first term forces the set E to be **conflict-free**: $P_{cf} = \sum_{iRj \text{ or } jRi} x_i x_j$. In fact, the value of P_{cf} corresponds to the number of self attacks in E and its value is 0 if and only if E is conflict-free.

The constraints to model the notion of **defence** are more complicated: we use a first set of additional variables $t_1, \ldots, t_n$, denoting which arguments are attacked by E:

[3] D-Wave Ocean SDK: https://github.com/dwavesystems/dwave-ocean-sdk.

$t_i = 1$ if and only if a_i is attacked by some argument of E. The variables $d_1, \ldots, d_n$ of the second set denote which arguments are defended by E: $d_i = 1$ if and only if a_i is defended (from all the possible attacks) by some arguments of E. For each argument a_i, the Pfunction P_t^i forces t_i to be 1 if and only if a_i is attacked by E, i.e., $t_i = \bigvee_{jRi} x_j$.

Let h_i be the number of attackers of a_i and let $i_1, \ldots, i_{h_i}$ be their indices. If $h_i = 0$, then t_i is simply 0, while if $h_i = 1$, then $t_i = x_{h_1}$: in these cases, we set $P_t^i = 0$. If $h_i = 2$, then $P_t^i = OR(t_i, x[i_1], x[i_2])$, where $OR(Z, X, Y) = W + X + Y + XY - 2Z(X + Y)$ is the way of expressing as a quadratic function the constraint that the binary variable Z is the disjunction of the binary variables X and Y, as shown in [10]. Finally, if $h_i > 2$, then $P_t^i = OR(t_i, x[i_1], \alpha_i^1) + OR(\alpha_i^1, x[i_2], \alpha_i^2) + \ldots + OR(\alpha_i^{h_i-3}, x[i_{h_i-2}], \alpha_i^{h_i-2}) + OR(\alpha_i^{h_i-2}, x[i_{h_i-1}], x[i_{h_i}]),$, where $\alpha_i^1, \ldots, \alpha_i^{h_i-2}$ are $h_i - 2$ auxiliary binary variables.

The other Pfunction P_d^i forces d_i to be 1 if and only if a_i is defended by E, i.e., $d_i = \bigwedge_{jRi} t_j$. If $h_i = 0$, then d_i is simply 1, while if $h_i = 1$, then $d_i = t_{h_1}$: in these cases, $P_d^i = 0$. If $h_i = 2$, then $P_d^i = AND(d_i, t[i_1], t[i_2])$, where $AND(Z, X, Y) = 3Z + XY - 2Z(X + Y)$ is the way of expressing the conjunction $Z = X$ and Y as a quadratic function [10]. Otherwise, if $h_i > 2$ then $P_d^i = AND(d_i, t[i_1], \delta_i^1) + AND(\delta_i^1, t[i_2], \delta_i^2) + \ldots + AND(\delta_i^{h_i-3}, t[i_{h_i-2}], \delta_i^{h_i-2}) + AND(\delta_i^{h_i-2}, t[i_{h_i-1}], t[i_{h_i}])$, where $\delta_i^1, \ldots, \delta_i^{h_i-2}$ are new $h_i - 2$ auxiliary binary variables.

The number of auxiliary variables needed for this encoding is hence $N = 2n + 2\sum_{i=1}^{n} \max(h_i - 2, 0)$, excluding the n variables $x_1, \ldots, x_n$. Note that, if $h = \max h_i$, then $N = O(nh)$. The final term $P_{def} = \sum_{i=1}^{n} x_i(1 - d_i)$ forces each argument in E to be defended by E. Summing up, the Pfunction for **admissible** sets is $P_{adm} = P_{cf} + \sum_{i=1}^{n} P_t^i + \sum_{i=1}^{n} P_d^i + P_{def}$. It is easy to prove that the minimum value of P_{adm} is 0 and the related values for $\underline{x}$ correspond to admissible sets. For the **complete** semantics, we simply need to add an additional term to P_{adm} which forces all the arguments defended by E to be elements of E: $P_{co} = P_{adm} + \sum_{i=1}^{n}(1 - x_i)d_i$.

B. Formulation in QUBO of extension enforcement

The task of extension enforcement can be formulated with similar techniques. Let us focus on the strict version of this problem. In order to simplify the notation, the arguments in the set T are the first k arguments $a_1, \ldots, a_k$ in A.

We use a first set of binary variables r_{ij}, for $i, j = 1, \ldots, n$. Each variable r_{ij} is 1 whether in the new attack relationship R', a_i attacks a_j. Moreover, we use the binary variables t_i, for $i = 1, \ldots, n$, and d_i, for $i = 1, \ldots, k$, as in the encoding of the acceptance.

We define a penalty function P_{co}^r which is zero if and only if T is complete set under the attack relationship described by r_{ij}. P_{co}^r is the sum of 5 terms.

The first term $P_{cf}^r = \sum_{i,j=1}^{k} r_{ij}$. enforces the set T to be conflict-free, in fact when $r_{ij} = 1$, with $i, j \leq k$, we have a self attack in T.

The second term is $P_t^r = \sum_{i=1}^{n} P_t^{r,i}$, where $P_t^{r,i}$, for each $i = 1, \ldots, n$, enforces the constraint $t_i = \bigvee_{j=1}^{k} r_{ji}$, which means that $t_i = 1$ if and only if the argument a_i is attacked by some argument $a_j \in T$. This term is encoded in QUBO using auxiliary binary variables, similar to what is done for P_t^i.

The third term is $P_d^r = \sum_{i=1}^{k} P_d^{r,i}$, where $P_d^{r,i}$, for each $i = 1, \ldots, k$, enforces the constraint $d_i = \bigwedge_{j=1}^{n}(r_{ji} \implies t_j)$, which means that $d_i = 1$ if and only if the argument $a_i \in T$ is defended against all its attacker by some elements of T. This term is encoded in QUBO using a new set of auxiliary variables to represent the implication $(r_{ji} \implies t_j)$, other than the same auxiliary variables used for P_d^i.

The fourth term is simply $\sum_{i=1}^{k}(1 - d_i)$, which requires that all arguments in T are defended, while the last term is $\sum_{i=k+1}^{n} d_i$, which add a penality for each argument defended by T, but not belonging to T.

The overall objective function to be minimized is $f = \sum_{a_i R a_j}(1 - r_{ij}) + \sum_{\neg a_i R a_j} r_{ij} + \lambda P_{co}^r$, where λ is a constant large, such that the minimum of f is obtained for $P_{co}^r = 0$.

IV. Conclusion

We introduced NP-complete problems in Abstract Argumentation that may benefit from QUBO encodings and their solution on quantum annealers. We summarised only some of them since others exist that consider, for example, weighted arguments and/or attacks. Moreover, further investigation is needed to better exploit the hardware and the connections among qubits, which are limited on D-Wave's architectures.

References

[1] J. L. Pollock, "How to reason defeasibly," *Artif. Intell.*, vol. 57, no. 1, pp. 1–42, 1992.

[2] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–358, 1995.

[3] P. Hammer and S. Rudeanu, "Boolean methods in operations research and related areas, ökonometrie und unternehmensforschung/econometrics and operations research," *Springer, Berlin*, vol. 1007, pp. 978–3, 1968.

[4] G. A. Kochenberger, J. Hao, F. W. Glover, M. W. Lewis, Z. Lü, H. Wang, and Y. Wang, "The unconstrained binary quadratic programming problem: a survey," *J. Comb. Optim.*, vol. 28, no. 1, pp. 58–81, 2014.

[5] M. Baioletti and F. Santini, "Abstract argumentation goes quantum: An encoding to QUBO problems," in *PRICAI 2022: 19th Pacific Rim International Conference on Artificial Intelligence*, ser. LNCS, vol. 13629. Springer, 2022, pp. 46–60.

[6] S. Coste-Marquis, S. Konieczny, J. Mailly, and P. Marquis, "Extension enforcement in abstract argumentation as an optimization problem," in *International Joint Conference on Artificial Intelligence, IJCAI.* AAAI Press, 2015, pp. 2876–2882.

[7] J. P. Wallner, A. Niskanen, and M. Järvisalo, "Complexity results and algorithms for extension enforcement in abstract argumentation," *J. Artif. Intell. Res.*, vol. 60, pp. 1–40, 2017.

[8] F. W. Glover, G. A. Kochenberger, and Y. Du, "Quantum bridge analytics I: a tutorial on formulating and using QUBO models," *4OR*, vol. 17, no. 4, pp. 335–371, 2019.

[9] S. Bistarelli, F. Rossi, and F. Santini, "Conarglib: an argumentation library with support to search strategies and parallel search," *J. Exp. Theor. Artif. Intell.*, vol. 33, no. 6, pp. 891–918, 2021.

[10] I. Rosenberg, "Reduction of bivalent maximization to the quadratic case," *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, vol. 17, p. 71–74, 1975.

A Quantum Evolutionary Strategy for Optimization Problems

1st Vincenzo Lipardi
Department of Advanced Computing Sciences
Maastricht University
Maastricht, Netherlands
vincenzo.lipardi@maastrichtuniversity.nl

2nd Roberto Schiattarella
Dipartimento di Fisica "E. Pancini"
University of Naples "Federico II"
Naples, Italy
roberto.schiattarella@unina.it

3rd Giovanni Acampora
Dipartimento di Fisica "E. Pancini"
University of Naples "Federico II"
Naples, Italy
giovanni.acampora@unina.it

***Abstract*—In this work we present an hybrid quantum-classical algorithm that significantly improves the performance of evolutionary algorithms on continuous optimization problems. It makes use of a classical evolutionary strategy to optimize shape and parameters of quantum circuits, which encode the solutions into the corresponding distribution probability over the computational basis states.**

I. Introduction

Evolutionary Algorithms (EAs) are a class of optimization algorithms widely applied on problems that cannot be solved through conventional techniques, because they avoid to compute high cost information, like gradient, by taking advantage of some bio-inspired stochastic operators.
There are three main reasons that let us think that quantum computing might a fundamental role in improving that kind of optimization methods:

1) Exponential advantage on space complexity:
2) Search can be moved on Hilbert spaces;
3) The stochasticity required is intrinsic in quantum phenomena.

The first point is justified by the use of an amplitude encoding technique which permits to represent an n-dimensional real vector on a quantum circuit equipped with $N = \lceil \log n \rceil$ qubits. In the second point we remark the possible benefit of moving the search in a different vector space with a higher dimension. The last point but not the least aims to get a good degree of randomness from the quantum phenomena involved in the algorithm itself.

II. A Quantum Evolutionary Strategy

The Quantum evolutionary strategy we are going to describe, the QES, deals with the search of the quantum circuit that better approximates the real vector solution of a continuous optimization problem. A classical optimization technique, an evolutionary strategy II-E, will learn the best shape and parameters of the circuit corresponding to the best solution through the map presented in II-B.

A. Problem Statement

Considering a function $f : A \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}$ in a closed bounded domain $\mathcal{D} \subseteq A$, the problem is finding an optimal solution $x^* \in A$ such that $f(x^*)$ is the optimal value, such as: the search for maximum (minimum) in maximization (minimization) problems. The search will be made in a generic n-dimensional box:

$$\mathcal{D} = [a_0, b_0] \times [a_1, b_1] \times \cdots \times [a_{n-1}, b_{n-1}]. \tag{1}$$

where $a_i, b_i \in \mathbb{R}$ and $i = 0, 1 \ldots, n-1$.
The problem so defined will be encoded in quantum circuits as described in the next section.

B. Encoding Technique

Given a N-qubit quantum circuit there is a quantum state $|\psi\rangle$ belonging to the Hilbert space $\mathcal{H}_N$. The vector $|\psi\rangle$ has 2^N complex coefficients, whose squared modulus form a probability distribution P over the computational basis states $|x\rangle \in \mathcal{H}_N$, which also correspond to the possible outcomes of a measure. Mathematically, we have a discrete variable X with values in all the possible combinations of the binary number x, which represent the corresponding $|x\rangle$ quantum states, and the set of probability values $p_i = P(X = x)$ have to fulfil the following properties:

$$\sum_{i=0}^{2^N-1} p_i = 1 \tag{2}$$

$$p_i \in [0,\ 1] \subset \mathbb{R} \tag{3}$$

where $i = 0, 1, \ldots, 2^N - 1$ and correspond to the decimal representation of the binary number x.
If we know the quantum circuit generating a quantum vector, we have access at these probabilities by executing multiple shots s; then each run is an event and the computational basis states are the possible outcomes on which the frequency probability distribution is built. Obviously, with the increase of the number of shots the sampled distribution P_s tends to the theoretical distribution P written in the squared modulus of $|\psi\rangle$. The running process to execute quantum circuits can be carried out on real quantum devices or by means of classical simulations.
In the proposed algorithm we respectively link the probability p_i to the component x_i of the vectors $x \in \mathcal{D}$. Then, we choose a number of qubits for our quantum circuits equal to

$$N = \lceil \log_2(n) \rceil + g, \tag{4}$$

where $g \in \mathbb{N}_0$ is an hyperparameter of the algorithm corresponding to the number of garbage qubits that we need to address generic continuous optimization problems.
Formally, N-qubit quantum circuits create probability distributions in the space of the probability distribution on all the $|x\rangle$ states, but we only use the first n states; in the following the space of the "effective" probability distributions is denoted as $\mathcal{P}$. As a result there is no guarantee about the normalization of the $P \in \mathcal{P}$ but a constraint is still remaining:

$$\sum_{i=0}^{n} p_i \leq 1. \tag{5}$$

At this stage, we can define the map

$$P \in \mathcal{P} \longrightarrow x \in \mathcal{D} \subset \mathbb{R}^n$$

requiring the surjectivity to explore the entire search space. The map $\mathcal{L}$ we have chosen works between the definition domain of probability and the subspaces $\mathcal{D}_i = [a_i, b_i]$, $\mathcal{L} : p_i \in [0, 1] \longrightarrow x_i \in \mathcal{D}_i$, acting as follows:

$$x_i = \mathcal{L}(p_i) = \begin{cases} p_i(b_i - a_i)n + a_i & \text{if } p_i = \left[0\,, \frac{1}{n}\right] \\ b_i & \text{otherwise} \end{cases} \tag{6}$$

The final map is a non-invertible function but anyway surjective. The resulting effect is that the genotypic space $\mathcal{P}$, is divided in an effective exploring zone, that is an n-dimensional cube of length $\frac{1}{n}$ where the algorithm explores the search space of the real problem, and its complement with respect to $\mathcal{D}$ named blind zone, where the algorithm is not affected by changing circuit parameters.

C. Quantum Circuit Initialization

The QES has a population composed of only one individual, a quantum circuit with N qubits as described previously. The first generation circuit is obtained through two sequential actions:

1) apply an Hadamard gate on each qubit of the circuit;
2) add an $R_Y(\theta)$ gate to the quantum circuit, on a random qubit with a random angle $\theta \in [0, 2\pi]$.

The first point changes the initial state of the quantum system from $|0 \ldots 0\rangle$ to $|+ \cdots +\rangle$, giving rise to an uniform probability distribution with $p_i = \frac{1}{2^N}$, which always fall inside the effective search space, since $0 < p_i < \frac{1}{n}$. We can drive the starting position of the p_i by adjusting the garbage qubits g; in particular, p_i dwindles with increasing g. A low value for g determines a small distance between the starting point in the genotypic search space and its bound. In principle, a good starting point could be center point $p_i = \frac{1}{2n}$, however it always depends on the problem. The second point has a twofold scope: it breaks the unjustified symmetry of the first candidate solution and it offers more possible actions to the evolution process.

D. Quantum Circuit Evaluation

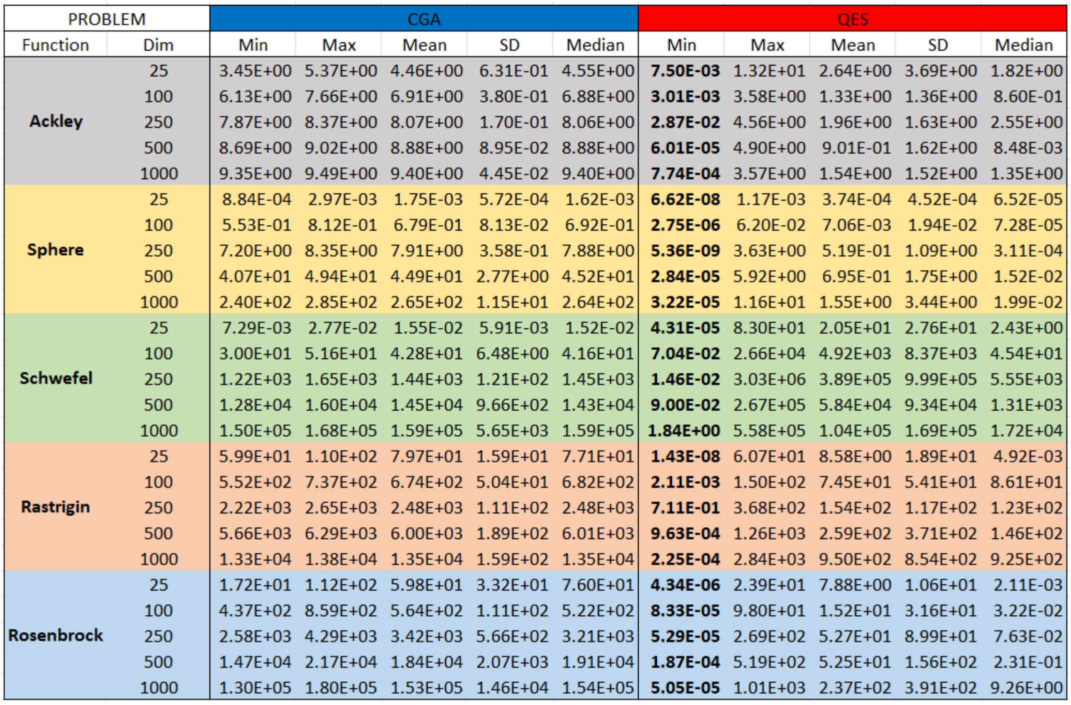
Give The fitness function of the evolutionary algorithm is the function f that defines the optimization problem, it is applied on the decoded solution $x = \mathcal{L}(p)$. Then, the fitness function for our quantum evolutionary algorithm is simply the function f or its opposite, depending on we respectively need to solve maximization or minimization problems.
The evaluation process has a significant dependence on the way the probability distribution is created. Qiskit provides us all the necessary for running quantum circuits on both classical simulators and real quantum processors trough the IBM Quantum Provider API. In case of *Statevector* simulations there are exact mathematical operations providing the accurate quantum state vector, hence the exact probability distribution P is obtained by the squared modulus of the complex coefficients. Otherwise, the *Qasm* simulator or real quantum circuit execution sample P by creating a frequency distribution P_s, where each event is represented by a shot. As the number of shots increases the sampled distribution P_s tends to P. However we need to increase the number of shots with the increasing of the dimension of the problem depending on the precision needed for the solutions.

E. Evolutionary Strategy

Evolutionary Strategies (ESs) are an example of EAs characterized by dealing with continuous candidate solutions. In QES we make use of a $(1 + \lambda)$-ES equipped with only a mutation strategy on quantum circuits playing the role of individuals of the EA [1].
The evolution is initialized has described in II-C, and after that λ quantum circuits are generated as copies of the single parent circuit. Then each of the λ circuit composing the offspring population is independently muted trough random choices between four actions. Now, the offspring is evaluated through the process described in II-D. If the best individual of the offspring has a better fitness than the parent, it will replace it; otherwise the parent survives. Then we have a mutation strategy that guides the search in $\mathcal{D}$ through the parent circuit evolution over the generations, approaching monotonically to the global optimum.
The mutation process is a two-level strategy. The first step consists in choosing one of the four possible actions to be applied to the quantum circuit:

1) ADD: Sample unitary $r \in \mathcal{G}$ and parameter θ uniformly and insert the corresponding gate at a random position.
2) DELETE: Delete gate at a random position from the circuit.
3) SWAP: Combination of DELETE and ADD at the same randomly chosen position.
4) MODIFY: Modify parameter θ of a randomly chosen gate:
$$\theta_{NEW} = \theta + \mathcal{N}(0,\, \Delta\theta)$$

where $\mathcal{N}(a,\, b)$ denotes a random number chosen with uniform probability in $[a, b]$, the angle $\Delta\theta$ permits to regulate the degree of the "modify" action and $p_{ACT} = [p_A, p_D, p_S, p_M]$



PROBLEM		CGA					QES				
Function	Dim	Min	Max	Mean	SD	Median	Min	Max	Mean	SD	Median
Ackley	25	3.45E+00	5.37E+00	4.46E+00	6.31E-01	4.55E+00	**7.50E-03**	1.32E+01	2.64E+00	3.69E+00	1.82E+00
	100	6.13E+00	7.66E+00	6.91E+00	3.80E-01	6.88E+00	**3.01E-03**	3.58E+00	1.33E+00	1.36E+00	8.60E-01
	250	7.87E+00	8.37E+00	8.07E+00	1.70E-01	8.06E+00	**2.87E-02**	4.56E+00	1.96E+00	1.63E+00	2.55E+00
	500	8.69E+00	9.02E+00	8.88E+00	8.95E-02	8.88E+00	**6.01E-05**	4.90E+00	9.01E-01	1.62E+00	8.48E-03
	1000	9.35E+00	9.49E+00	9.40E+00	4.45E-02	9.40E+00	**7.74E-04**	3.57E+00	1.54E+00	1.52E+00	1.35E+00
Sphere	25	8.84E-04	2.97E-03	1.75E-03	5.72E-04	1.62E-03	**6.62E-08**	1.17E-03	3.74E-04	4.52E-04	6.52E-05
	100	5.53E-01	8.12E-01	6.79E-01	8.13E-02	6.92E-01	**2.75E-06**	6.20E-02	7.06E-03	1.94E-02	7.28E-05
	250	7.20E+00	8.35E+00	7.91E+00	3.58E-01	7.88E+00	**5.36E-09**	3.63E+00	5.19E-01	1.09E+00	3.11E-04
	500	4.07E+01	4.94E+01	4.49E+01	2.77E+00	4.52E+01	**2.84E-05**	5.92E+00	6.95E-01	1.75E+00	1.52E-02
	1000	2.40E+02	2.85E+02	2.65E+02	1.15E+01	2.64E+02	**3.22E-05**	1.16E+01	1.55E+00	3.44E+00	1.99E-02
Schwefel	25	7.29E-03	2.77E-02	1.55E-02	5.91E-03	1.52E-02	**4.31E-05**	8.30E+01	2.05E+01	2.76E+01	2.43E+00
	100	3.00E+01	5.16E+01	4.28E+01	6.48E+00	4.16E+01	**7.04E-02**	2.66E+04	4.92E+03	8.37E+03	4.54E+01
	250	1.22E+03	1.65E+03	1.44E+03	1.21E+02	1.45E+03	**1.46E-02**	3.03E+06	3.89E+05	9.99E+05	5.55E+03
	500	1.28E+04	1.60E+04	1.45E+04	9.66E+02	1.43E+04	**9.00E-02**	2.67E+05	5.84E+04	9.34E+04	1.31E+03
	1000	1.50E+05	1.68E+05	1.59E+05	5.65E+03	1.59E+05	**1.84E+00**	5.58E+05	1.04E+05	1.69E+05	1.72E+04
Rastrigin	25	5.99E+01	1.10E+02	7.97E+01	1.59E+01	7.71E+01	**1.43E-08**	6.07E+01	8.58E+00	1.89E+01	4.92E-03
	100	5.52E+02	7.37E+02	6.74E+02	5.04E+01	6.82E+02	**2.11E-03**	1.50E+02	7.45E+01	5.41E+01	8.61E+01
	250	2.22E+03	2.65E+03	2.48E+03	1.11E+02	2.48E+03	**7.11E-01**	3.68E+02	1.54E+02	1.17E+02	1.23E+02
	500	5.66E+03	6.29E+03	6.00E+03	1.89E+02	6.01E+03	**9.63E-04**	1.26E+03	2.59E+02	3.71E+02	1.46E+02
	1000	1.33E+04	1.38E+04	1.35E+04	1.59E+02	1.35E+04	**2.25E-04**	2.84E+03	9.50E+02	8.54E+02	9.25E+02
Rosenbrock	25	1.72E+01	1.12E+02	5.98E+01	3.32E+01	7.60E+01	**4.34E-06**	2.39E+01	7.88E+00	1.06E+01	2.11E-03
	100	4.37E+02	8.59E+02	5.64E+02	1.11E+02	5.22E+02	**8.33E-05**	9.80E+01	1.52E+01	3.16E+01	3.22E-02
	250	2.58E+03	4.29E+03	3.42E+03	5.66E+02	3.21E+03	**5.29E-05**	2.69E+02	5.27E+01	8.99E+01	7.63E-02
	500	1.47E+04	2.17E+04	1.84E+04	2.07E+03	1.91E+04	**1.87E-04**	5.19E+02	5.25E+01	1.56E+02	2.31E-01
	1000	1.30E+05	1.80E+05	1.53E+05	1.46E+04	1.54E+05	**5.05E-05**	1.01E+03	2.37E+02	3.91E+02	9.26E+00

is a normalized list of probabilities for the corresponding action to occur. The second step of the mutation strategy consists in repeating, with a certain probability p_R, the entire mutation process at the end of each action. Therefore, we have introduced several hyperparameters to be tuned: $\Delta\theta$, p_{ACT} and p_R.

An important remark is that the chosen universal set $\mathcal{G}$ permits to explore the search space with both differentiable (angle parameters) and not-differentiable operations (adding, deleting and swapping discrete number of gates).

III. Results

The experimental evaluation of the QES aims to compare it to a classical genetic algorithm on a set of well-known real-encoded optimization problems. The baseline considered in the comparison is related to the genetic algorithms used in [2]. A set of 5 functions has been considered, as well as a set of 5 dimensionality of the problem for each of them. Namely, the size of the problems considered are 25, 100, 250, 500, 1000. The goal is to analyze the behavior of QES in optimizing both convex and non-convex functions characterized by an high dimensionality. Then, the QES has been tested on a heterogeneous sample of functions, on bowl-shaped sphere model and on other non-convex functions presenting peculiar features.

Note that the quantum results presented here come from the quantum circuit simulation executed on local computers through the statevector simulator on Qiskit.

IV. Conclusions

We presented an hybrid classical-quantum algorithm that in principle can solve any continuous optimization problem by learning shape and parameters of the quantum circuits in which the solutions are encoded. Note that it differs from the canonical Variational Quantum Circuits (VQCs) as the anstatz is learnt by the algorithm itself [3].

The preliminary results are promising and some interesting future directions have come up. An adaptation for NISQ devices is readily feasible, for example by cancelling the possibility to add more gate at the quantum circuits as soon as the maximum depth is reached.

An important future direction to carry out is the investigation on the quantum noise's influence over performances. About this point we expect two possible scenario: neither it will give the amount of stochasticity that our algorithm needs or the noise is too much that makes lost the information encoded in the circuits and our algorithm will become only a complicated random search. A deeper understanding on the ... is also required for a more theoretical justification of the results.

References

[1] Franken, L., Georgiev, B., Mucke, S., Wolter, M., Heese, R., Bauckhage, C., Piatkowski, N. (2022, July). Quantum Circuit Evolution on NISQ Devices. In 2022 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.

[2] Herrera, F., Lozano, M., Sánchez, A. M. (2003). A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. International Journal of Intelligent Systems, 18(3), 309-338.

[3] Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M., Gambetta, J. M. (2017). Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. nature, 549(7671), 242-246.

Application of Quantum Genetic Algorithms to Network Signal Setting Design

Giovanni Acampora
Department of Physics "Ettore Pancini"
University of Naples Federico II
Napoli, Italy
giovanni.acampora@unina.it

Angela Chiatto
Department of Physics "Ettore Pancini"
University of Naples Federico II
Napoli, Italy
angela.chiatto@unina.it

Stefano de Luca
Department of Civil Engineering
University of Salerno
Fisciano (SA), Italy
sdeluca@unisa.it

Roberta Di Pace
Department of Civil Engineering
University of Salerno
Fisciano (SA), Italy
rdipace@unisa.it

Alfredo Massa
Department of R&D
QuantumNet srl
Napoli, Italy
alfredo.massa@quantum-net.it

Roberto Schiattarella
Department of Physics "Ettore Pancini"
University of Naples Federico II
Napoli, Italy
roberto.schiattarella@unina.it

Autilia Vitiello
Department of Physics "Ettore Pancini"
University of Naples Federico II
Napoli, Italy
autilia.vitiello@unina.it

***Abstract*—The regulation of traffic lights in a signalised urban network requires optimizing objective functions that represent performance indicators of one or more intersections (such as delay or queue length). In this scenario, evolutionary algorithms are adopted to find suitable approximate solutions, in cases when no deterministic algorithm for finding the exact solution is known. This paper attempts to further improve the performance of evolutionary approaches by using a hybrid quantum-classical genetic algorithm to find the optimal configuration of the green signal timing regulating the traffic flow across two interacting junctions. The adopted algorithm, run on IBM quantum computer simulators, is shown to be suitable for the optimization problem at hand. Indeed, the experimental results highlight some of the strengths of the proposed technique with respect to the purely evolutionary approach, and encourage the application of this approach to more complex and close-to-real application scenarios.**

***Index Terms*—Genetic Algorithms, Optimisation, Quantum Computing, Signal Setting**

I. Introduction

The optimisation of a transportation system in terms of capacity and sustainability is a major and continuous engineering challenge. At urban level, most strategies involve the control of road junction networks. Traffic lights are one of the most common ways to achieve such control. The design of control variables can be formulated as an optimisation problem, often named *Network Signal Setting Design (NSSD)*. Such problem often involves a large number of decision variables, a series of constraints to be satisfied and, in many cases, more than one objective function to be optimised. It shows therefore a relevant complexity which prevents the search of the exact solution, especially in those cases where a given urban network needs to be regulated repeatedly for real time traffic control. Meta-heuristic algorithms are generally adopted since they allow for the approximation of the optimal solution of difficult problems. Genetic Algorithms (GAs) are one of the mainly used meta-heuristics thanks to their effectiveness and versatility. In a typical GA workflow a random population of solutions is generated; then, the population evolves through an iterative process involving the action of three *genetic operators*, named *selection*, *crossover* and *mutation*, whose purpose is to explore the solution space and gradually shift towards better solutions across the iterations. Different characterisations and implementations of such operators can heavily influence the performance of the algorithm, so they are usually tailored for the problem at hand. There has recently been a growing interest in combining techniques of computational intelligence, such as GAs, with quantum computing, a disruptive paradigm of computation that harnesses the quantum properties of matter (such as *superposition* and *entanglement*) to achieve a computational advantage with respect to some specific tasks. Besides computational power, from a GAs perspective the transition to a hybrid quantum-classical framework opens up innovative possibilities for the encoding of chromosome features and the implementation of genetic operators. In this regard, the quantum paradigm has already proven to be a valuable toolbox to take into consideration. Recently, a so-called *Quantum Mating Operator (QMO)* has been introduced and its use in a genetic optimization process has produced a significant improvement over algorithms equipped with conventional evolutionary operators [1]. In this work, a QMO-based GA has been used to minimize an objective function describing the total deterministic delay in a network, which is one of the most important performance indicators of a signalised junction with a direct impact on the waiting time of vehicles. The results are very promising and show that quantum-classical hybrid genetic optimization represents a high-potential approach for solving the NSSD problem and, in general, for optimization in transport systems, especially when quantum processors prove sufficiently mature for intensive use in real-world scenarios.

II. Network Signal Setting Design

A NSSD instance can be seen as an optimization problem aimed at finding the optimal configuration of a set of control variables to improve the performance of a road network with fixed layout. The control variables usually involve the *green timings*, which are the durations of the green signals for the different *approaches* of a given network (i.e., the number of incoming lanes receiving identical signals over time). More precisely, one usually assigns a duration to each *stage*: a stage is a time interval during which all the signals remain unchanged. In addition to the timing, green signals need to be scheduled. The scheduling regulating each single junction is commonly formalised by means of a *stage matrix* Δ. The number of rows of Δ is the number of approaches of the junction, while the number of columns corresponds to the number of stages. The generic entry $\delta_{ij} \in \Delta$ is equal to 1 if approach i receives a green signal during stage j, otherwise it is 0. Stages are repeated cyclically and the duration of a cycle is given by the sum of the durations of the stages. The cycle length is usually common to all the junctions in a network, however it is possible to introduce time offsets between cycles of different junctions. In such cases, offsets become additional decision variables to take into account. The optimisation can involve one or more objective functions, namely a number of performance indicators that depend on the control variables more or less directly.

III. Case Study

The network consists of an arterial joining two signalised junctions, namely an upstream junction J_l and a downstream junction J_h, as shown in Fig. 1. The distance d between the junctions is 300 m. J_l is an isolated junction, while J_h is a "T" junction characterized as in [2], except in our case two approaches out of five ($k = 1$ and $k = 2$) are interacting and the remaining three are isolated. This means that the arrival flows q_k (number of vehicles arriving at approach k per unit time) are fixed for $k \in \{3, 4, 5\}$, but vary in time for $k \in \{1, 2\}$ due to the interaction with the upstream node. Therefore, these time-dependent flows must be computed by performing a traffic simulation. The arrival flows q_k and the saturation flows s_k (maximum number of vehicles per unit time which coming from approach k and crossing the junction during a green signal in presence of a queue), are reported in Table I, which can be found in [13]. All of the approaches of both the junctions are signalised. The stage matrices describing the green scheduling can be found in [13].

The traffic flow simulation across the network of Fig. 1 reproduces the dynamic of the traffic flow for a total duration $T = 15$ min. The interval T is discretized in sub-intervals of duration $\Delta\tau = 1$ s. For every i-th sub-interval, different traffic profiles are computed according to a given traffic flow model. The model chosen for this study is based on Robertson's approach (1969). The first traffic profile is q_I; it enters the arterial from J_l and is set to the constant value of 1200 pcu/h. The constant stream speed is $v = 9$ m/s. The dispersion profile q_{Sk} reaching the stop line at the k-th approach of the downstream junction at time $i + \Delta i$ is given by the following recursive expression [7]:

$$q_{Sk}(i + \Delta i) = F q_{Ik}(i) + (1 - F) q_{Sk}(i + \Delta i - 1) \qquad (1)$$

where:

$$\Delta i = \lfloor \frac{0.5 + 0.8 t_{lh}}{\Delta\tau} \rfloor;$$

$$F = \frac{1}{1 + 0.4 t_{lh}}; \qquad (2)$$

$$t_{lh} = \frac{d}{v} = 33.3 \; s.$$

The profile q_{Uk} outgoing J_h from the k-th approach is given by: $q_{Uk} = 0$ in case of red signal at approach k; $q_{Uk} = s_k$ in case of green signal at approach k and $q_{Sk} > s_k$; $q_{Uk} = q_{Sk}$ otherwise (s_k is the saturation flow of approach k). Once the profiles q_S and q_U have been computed for each i-th sub-interval in T, it is possible to calculate the total deterministic delay D built up during T (at given J_h approach). Such delay represents the objective function to minimize and is given by:

$$D_k(T) = \sum_{i=1}^{T/\Delta\tau} \sum_{j=1}^{i} \left[q_{Sk}(j) - q_{Uk}(j) \right] \Delta\tau^2. \qquad (3)$$

The decision variables are the following: t_1, t_2 and t_3, i.e. the durations (in seconds) of the stages of J_l; t_4, t_5 and t_6, i.e. the durations (in seconds) of the stages of J_h; ϕ, i.e. the offset (in seconds) between the cycles of J_l and J_h. The duration of the cycle is common to both junctions and is given by $C = 64$ s, as prescribed in [2]. This implies that, for each junction, only two t_i out of three are independent. Therefore, the actual optimisation variables of our problem are reduced to t_1, t_2, t_4,

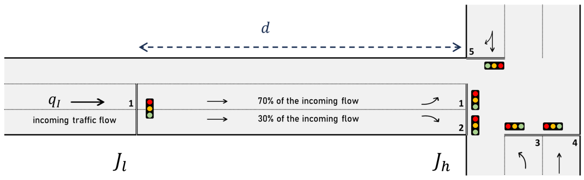


Fig. 1. Arterial layout.

t_5, and ϕ. Not every configuration of the optimisation variables can be accepted as a feasible solution for the problem, because of a number of constraints involving the *effective green* g_k and the *capacity factor* CF_k functions. Further information about such functions and the corresponding constraints can be found in [13].

IV. Experiments and Results

We choose a set of GAs for a performance comparison. In particular, our proposal based on a hybrid genetic algorithm is compared with two different classical GAs. In the following, the classical GAs will be referred to as GA_1 and GA_2, while the QMO-based GA will be denoted GA_3, where: GA_1 is a classical GA using a single point crossover operator, a bit flip mutation operator and a tournament selection operator; GA_2 is similar to GA_1, except a two-point crossover is adopted; GA_3 is a quantum-classical GA where crossover and mutation are implemented via QMO, and a tournament selection is used. The performance will be evaluated, upon hyperparameter tuning, by considering the average fitness values achieved by each GA_i through 30 independent runs. In each individual GA run, the initial population is made of 10 random chromosomes. One chromosome contains 5 genes, corresponding to the aforementioned optimisation variables, namely t_1, t_2, t_4, t_5, and ϕ. Each gene is an integer number n such that $0 < n < C - 1$, expressed in reflected binary representation. The delay function (3) is used as a fitness function to minimise and all the GAs use the same termination condition, i.e., the exceeding of a maximum number of fitness evaluations, set to 500. Every time it is necessary to evaluate the fitness function of a given chromosome throughout the evolutionary computation, the traffic simulation comes into play: since the fitness is given by the delay function (3), to obtain its value we need to compute the traffic profiles defined above, which ultimately depend on the genes t_1, t_2, t_4, t_5, and ϕ, and can only be obtained by running a traffic simulation.

The quantum part of the computation is performed through IBM's Qiskit development framework. In particular, GA_3 is executed twice: the first time we use Qiskit's *qasm_simulator*, a local simulator of an ideal quantum computer; the second time we use a different simulator reproducing the behaviour of the *ibm_nairobi* device, i.e. an actual 7-qubit quantum processor, whose behaviour is likely to be disturbed by environmental noise. The results in Fig. 2 show no remarkable difference among the performances of GA_1, GA_2 and GA_3 in the case of ideal quantum simulation. The case of GA_3 executed by a non-ideal simulator (last box plot of Fig. 2), instead, shows better performance both in terms of fitness values and dispersion of values. In conclusion, a comparison of QMO's performance with two other classical sets of genetic operators shows an advantage of the quantum operator in terms of stability and constraints satisfiability while exploring the space of possible solutions. Furthermore, the results suggest the suitability of this hybrid quantum-classical approach for the problem at hand and encourage its application to more challenging multi-criteria optimisation problems.

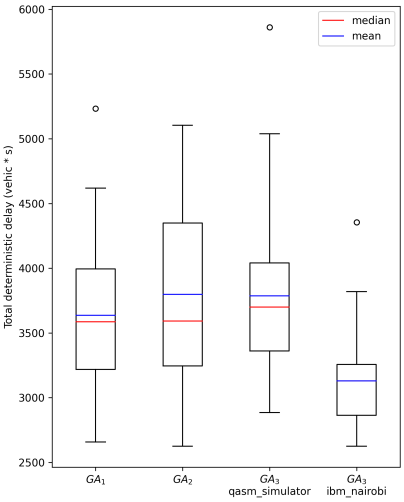


Fig. 2. Box plots showing mean and median values of the fitness function (i.e., the delay) over 30 executions.

References

[1] Acampora, G., Schiattarella, R., and Vitiello, A. (2022, July). Quantum Mating Operator: A New Approach to Evolve Chromosomes in Genetic Algorithms. In 2022 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.

[2] Cantarella, G. E., De Luca, S., Di Pace, R., and Memoli, S. (2013, September). The application of multicriteria genetic algorithms for signal setting design at a single junction. In 2013 8th EUROSIM Congress on Modelling and Simulation (pp. 472-477). IEEE.

[3] Nielsen, M. A., and Chuang, I. L. (2001). Quantum computation and quantum information. Phys. Today, 54(2), 60.

[4] Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., ... and Aspuru-Guzik, A. (2022). Noisy intermediate-scale quantum algorithms. Reviews of Modern Physics, 94(1), 015004.

[5] Robertson, D. I. (1969). TRANSYT: a traffic network study tool.

[6] Hunt, P. B., Robertson, D. I., Bretherton, R. D., and Winton, R. I. (1981). SCOOT-a traffic responsive method of coordinating signals (No. LR 1014 Monograph).

[7] Storani, F., Di Pace, R., Bruno, F., and Fiori, C. (2021). Analysis and comparison of traffic flow models: a new hybrid traffic flow model vs benchmark models. European transport research review, 13(1), 1-16.

[8] Di Pace, R., Fiori, C., Storani, F., de Luca, S., Liberto, C., and Valenti, G. (2022). Unified network tRaffic management frAmework for fully conNected and electric vehicles energy cOnsumption optimization (URANO). Transportation Research Part C: Emerging Technologies, 144, 103860.

[9] Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.

[10] Kumar, M., Husain, D., Upreti, N., and Gupta, D. (2010). Genetic algorithm: Review and application. Available at SSRN 3529843.

[11] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison. Reading.

[12] Fortin, F. A., De Rainville, F. M., Gardner, M. A. G., Parizeau, M., and Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. The Journal of Machine Learning Research, 13(1), 2171-2175.

[13] Acampora, G., Chiatto, A., de Luca, S., Di Pace, R., Massa, A., Schiattarella, R., and Vitiello, A. (2023, July). Application of Quantum Genetic Algorithms to Network Signal Setting Design. Accepted in 2023 IEEE Congress on Evolutionary Computation (CEC). IEEE.

Session III: Quantum Computing and Computational Intelligence

Continuous Variable Quantum Physics-Informed Neural Networks

Stefano Markidis
Computer Science Department, KTH Royal Institute of Technology, Stockholm, Sweden

***Abstract*—Physics-Informed Neural Networks (PINNs) have emerged as a powerful tool in scientific computing, encompassing tasks like solving Partial Differential Equations and data assimilation. This work investigates the design of PINNs using the Continuous Variable (CV) Quantum Processing Unit. We develop a straightforward Quantum PINN framework using a CV quantum computing paradigm to solve the one-dimensional Poisson problem. We study the impact of different optimizers, showing that the optimizer exploration of the training landscape in quantum PINNs is less effective than classical PINNs, with a basic Stochastic Gradient Descent (SGD) optimizer outperforming adaptive and high-order optimizers.**

***Index Terms*—Quantum Scientific Machine Learning**

I. Introduction

This paper discusses the concept of Quantum Physics-Informed Neural Networks (PINNs) and explores the potential of using Continuous Variable (CV) quantum computers for deploying PINNs. Classical PINNs are neural networks used to solve Partial Differential Equations (PDEs) in scientific computing [1]. PINNs consist of two interconnected neural networks: a surrogate network and a residual network. The surrogate network approximates the solution of the PDE at a given point, while the residual network calculates the error of the approximation [2]. The main strength is that PINNs leverage automatic differentiation to perform differential operator calculations without the need for discretization or meshes. This work investigates the potential of using Quantum Processing Units (QPUs) and associated software for deploying PINNs on CV quantum computers. Quantum PINNs employ variational quantum circuits and leverage CV quantum computing, which uses physical observables and is well-suited for approximating continuous functions [3]–[5].

II. Continuous Variable Quantum Computing

The CV quantum computing approach is founded on the concept of a *qumode*, which serves as the fundamental unit of information in CV quantum computing. We represent the qumode, denoted as $|\psi\rangle$, using a basis expansion of quantum states:$|\psi\rangle = \int \psi(x)\,|x\rangle\,dx$, where the states correspond to eigenstates of the $\hat{x}$ quadrature. These eigenstates are expressed as $\hat{x}\,|x\rangle = x\,|x\rangle$, with x representing a real-valued eigenvalue.

While qubit-based quantum computing employs discrete coefficients, the CV-based approach utilizes a continuous spectrum of coefficients. CV quantum operators include the position ($\hat{x}$) and momentum ($\hat{p}$) g. The position operator is defined as $\hat{x} = \int_{-\infty}^{\infty} x\,|x\rangle\,\langle x|\,dx$ where the vectors $|x\rangle$ are orthogonal.

Similar to the established qubit-based formulation, CV quantum computation can be expressed using low-level gates that can be realized, for example, through photonics devices [6]. A CV quantum program can be viewed as a sequence of gates acting on one or more qumodes. To develop CV quantum neural networks and CV quantum PINNs, four fundamental *Gaussian* gates operating on qumodes are required. These four linear gates are as follows:

- Displacement Gate - $D(\alpha)$: $\begin{bmatrix} x \\ p \end{bmatrix} \to \begin{bmatrix} x + \Re(\alpha) \\ p + \Im(\alpha) \end{bmatrix}$. This operator corresponds to a phase space shift by displacing a complex number $\alpha \in \mathbb{C}$.
- Rotation Gate - $R(\phi)$: $\begin{bmatrix} x \\ p \end{bmatrix} \to \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}$. This operator corresponds to a rotation of the phase space by an angle $\phi \in [0, 2\pi]$.
- Squeezing Gate - $S(r)$: $\begin{bmatrix} x \\ p \end{bmatrix} \to \begin{bmatrix} e^{-r} & 0 \\ 0 & e^{r} \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}$. This operation corresponds to a scaling operation in the phase space with a scaling factor $r \in \mathbb{C}$.
- Beam-splitter Gate - $BS(\theta)$: $\begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix} \to \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) & -\sin(\theta) \\ 0 & 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix}$. This operation is similar to a rotation between two qumodes by an angle $\theta \in [0, 2\pi]$.

Apart from these four fundamental Gaussian gates, there are also non-Gaussian gates, such as the *cubic* and *Kerr* gates. These non-Gaussian gates introduce non-linearity, similar to the activation functions in classical neural networks. More importantly, when combined with Gaussian gates in a sequence of quantum computing units, non-Gaussian gates provide universality to the CV quantum circuit, ensuring the ability to generate any CV state. In this study, we employ the Kerr gate that is typically represented as $K(\kappa)$, where $\kappa \in \mathbb{R}$ serves as the quantum gate parameter.

Lastly, in quantum computing, the outcome of an operation is determined through a measurement. In this study, the measured result is used to evaluate the expected value of the quadrature operator $\hat{x}$, given by:$\langle\psi_x|\,\hat{x}\,|\psi_x\rangle$. In a simple regression task, where the goal is to approximate a function $f(x)$, a quantum neural network is trained to estimate $\langle\psi_x|\,\hat{x}\,|\psi_x\rangle = f(x)$ for all values of x. This is achieved by finding optimal gate parameters (α, ϕ, etc.) that satisfy the equation through an optimization process.

III. A Continuous-Variable Quantum PINN

Quantum Physics-Informed Neural Networks (Quantum PINNs) build upon the foundation of CV quantum neural networks, as demonstrated in previous studies [5], [7]. Fig. 1 provides an overview of the workflow and resource utilization involved in employing a quantum PINN for solving a 1D Poisson equation with associated boundary conditions ($\nabla^2\tilde{\Phi}(x) = b(x)$). The Poisson equation serves as a fundamental governing equation in scientific computing, describing various phenomena such as electrostatic and gravitational forces.

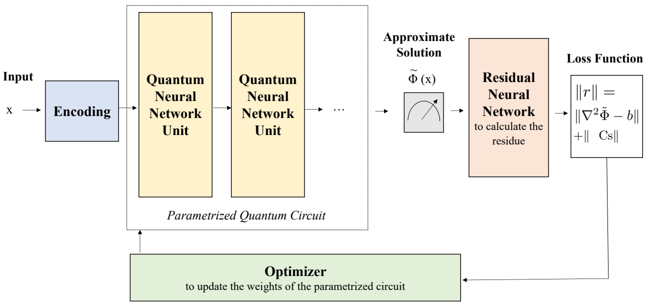

Input
x
Encoding
Quantum Neural Network Unit
Quantum Neural Network Unit
...
Parametrized Quantum Circuit
Approximate Solution
$\tilde{\Phi}$ (x)
Residual Neural Network
to calculate the residue
Loss Function
$\|r\| =$
$\|\nabla^2\tilde{\Phi} - b\|$
$+\|$ Cs$\|$
Optimizer
to update the weights of the parametrized circuit

Fig. 1: A diagram representing the workflow in a quantum PINN.

In the initial step of the quantum PINN process, the collocation point is encoded as a displacement in the vacuum state. This encoding entails a real-valued displacement without any displacement or squeezing in the phase space. Other encoding strategies are also possible. A notable advantage of employing CV quantum neural networks is the convenience of input encoding into the quantum neural network without requiring normalization. Following the initial displacement encoding, the qumode proceeds into the quantum PINN, which consists of two interconnected neural networks:

- Quantum Surrogate Neural Network: The quantum neural network surrogate is a CV quantum neural network described in Ref. [8]. It constitutes a parameterized quantum circuit that takes the collocation point coordinate (x) as input, encoded as a displacement or a rotation in the vacuum state. The output of this network is the approximate solution ($\tilde{\Phi}(x)$), represented by the expected value of the quadrature operator.
- Residual Neural Network: Quantum PINNs operate in a matrix-free manner, eliminating the need to store and encode the problem matrix and source vector. Instead, the residual network encodes the governing equations. The residual network is not trained, meaning its weights are not updated. Its primary function is to provide the quantum surrogate neural network with a loss function, specifically the residual function for inner points (ip) within the domain $|r|_{ip} = |\nabla^2\tilde{\Phi}(x) - b(x)|$. Furthermore, in addition to satisfying the governing equation for inner points, the collocation points on the boundaries must fulfill the problem's boundary conditions.

The fundamental component of a quantum neural network is the quantum neural network unit, also known as a quantum network layer in the literature, which shares similarities with the classical neural network unit. Fig. 2 illustrates an example of the quantum neural unit elements. The first three components of the quantum neural unit consist of an initial interferometer, a squeezing gate, and a subsequent interferometer. It has been demonstrated in Ref. [8] that the combined effect of these operations is analogous to multiplying the phase space vector by the neural network weights W, which are the parameters of the interferometers and squeezing gates. Similar to classical neural networks, a displacement gate emulates the addition of a bias b. Lastly, a Kerr gate (or cubic gate) introduces a non-linearity akin to the activation function σ in classical neural networks.

By stacking multiple quantum neural units in a sequence, we can construct a quantum neural network. It is important to note that, for a single qumode, each gate can be controlled by a total of seven gate parameters (α, ϕ, r, θ, and κ), which can either be real-valued (ϕ, θ, and κ) or complex numbers (α, r). The complex-valued parameters can be expressed using two real numbers in Cartesian form (with real and imaginary parts) or polar form (with amplitude and phase). The training of quantum neural networks aims to determine the parameterized quantum circuit values (α, ϕ, r, θ, and κ) for different qumodes and quantum neural units, with the objective of minimizing the PINN cost function, which corresponds to the PDE residual.

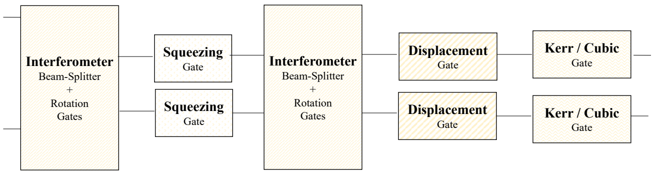


Fig. 2: An example of CV quantum neural unit.

IV. Results

In this work, we employ the Python `Strawberryfields` CV quantum simulator [9], [10] along with a set of Python modules to facilitate efficient vector calculations on the CPU and utilize additional optimizers not present in the TensorFlow/Keras framework [11]. Our implementation relies on Python version 3.10.4, as well as the NumPy (1.22.4) and SciPy (1.8.1) modules. The experiments are conducted using the quantum computer simulator provided by the `Strawberryfields` framework, specifically version 0.22.0. Throughout all simulations, a single quantum mode is utilized, and a cutoff dimension of 125 is employed for the Fock basis. In this study, we evaluate the effectiveness of several optimizers including Stochastic Gradient Descent, Adam, Simultaneous Perturbation Stochastic Approximation (SPSA), and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS-B) optimizers. An implementation of a CV Quantum PINN is available on GitHub [1].

To simplify the evaluation of the quantum PINN, we focus on a one-dimensional Poisson problem with fixed Dirichlet boundary conditions, given by $d^2\Phi/dx^2 = b(x)$. For testing purposes, we select two types of sources (represented by the b term in the Poisson equation) and two different domain sizes: *(i)* Quadratic: $b(x) = x(x-1)$, [0, 1], $\Phi(0) = 0$, $\Phi(1) = 0$. In this case, the solution is a parabola with the first derivative equal to zero at the domain's center, $x = 0.5$. *(ii)* Sinusoidal: $b(x) = \sin(2x)$, [0, 2π], $\Phi(0) = 0$, $\Phi(2\pi) = 0$. This test case is more challenging as the solution has four points where the first derivative is zero. Fig. 3 shows the cost function value together with the final error for the different problems using different optimizers and a quantum PINN with four units. By analyzing the cost function value and final error, it is clear that the SGD optimizer outperforms the adaptive and SPSA optimizers. In general, we find that adaptive optimizers, such as Adam, tend to converge to local minima in the training landscape without exiting. A noisier optimizer, such as SGD, can escape the local minima and better explore the optimization landscape.

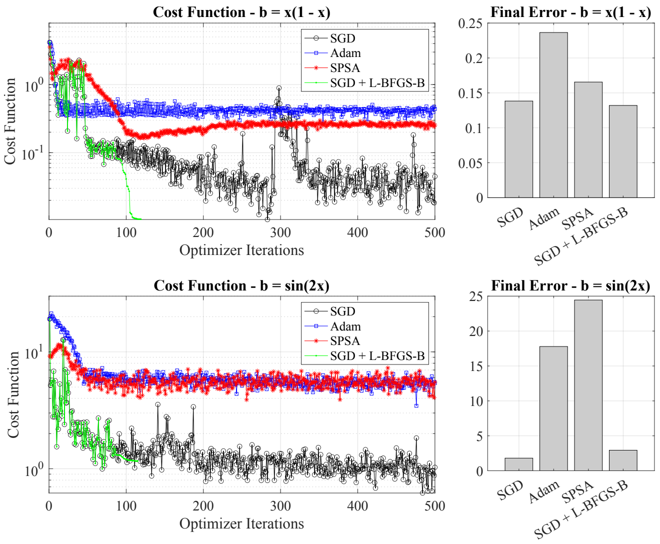


Fig. 3: Cost function value evolution and final errors for different stochastic optimizers.

V. Conclusion

This work focused on the development of Physics-Informed Neural Networks (PINNs) for solving differential equations with a CV quantum computing approach. Quantum PINN solvers are variational quantum circuit solvers, augmented with a classical residual network that serves as a cost function for the optimization process. We demonstrated that a CV quantum neural network can serve as a surrogate for approximating the solution to a PDE to a certain extent, as evidenced by our tests on the one-dimensional Poisson equation with quadratic and sinusoidal source terms. The choice of optimizer had the most significant impact on the accuracy of the PINN solver, with SGD outperforming adaptive optimizers in terms of accuracy and stability. A key area for further development of quantum PINNs is to address their current limitations in terms of accuracy. In our experiments, it was challenging to reduce the error below a certain threshold or achieve faster convergence within a finite number of iterations. This difficulty is likely related to the *barren plateau* problem, which affects hybrid quantum-classical algorithms involving an optimization step [12]. In fact, the optimizer is entrapped on a barren plateau within the training landscape, with an exponentially small probability of escaping it. The *barren plateau* problem is rooted in the geometry of parameterized quantum circuits (in our case, the quantum surrogate neural network) and training landscapes associated with hybrid classical-quantum algorithms. Potential strategies to mitigate this problem include using different quantum network architectures, employing skip connections, applying dropout techniques, using a structured initial guess similar to quantum simulations, and pre-training.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[2] S. Markidis, "The old and the new: Can physics-informed deep-learning replace traditional linear solvers?" *Frontiers in big Data*, p. 92, 2021.

[3] A. E. Paine, V. E. Elfving, and O. Kyriienko, "Quantum kernel methods for solving differential equations," *arXiv preprint arXiv:2203.08884*, 2022.

[4] S. Markidis, "On physics-informed neural networks for quantum computers," *Frontiers in Applied Mathematics and Statistics*, vol. 8, 2022.

[5] M. Knudsen and C. B. Mendl, "Solving differential equations via continuous-variable quantum computers," *arXiv preprint arXiv:2012.12220*, 2020.

[6] K. Fukui and S. Takeda, "Building a large-scale quantum computer with continuous-variable optical technologies," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 55, no. 1, p. 012001, 2022.

[7] O. Kyriienko, A. E. Paine, and V. E. Elfving, "Solving nonlinear differential equations with differentiable quantum circuits," *Physical Review A*, vol. 103, no. 5, p. 052416, 2021.

[8] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Physical Review Research*, vol. 1, no. 3, p. 033063, 2019.

[9] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, "Strawberry fields: A software platform for photonic quantum computing," *Quantum*, vol. 3, p. 129, 2019.

[10] T. R. Bromley, J. M. Arrazola, S. Jahangiri, J. Izaac, N. Quesada, A. D. Gran, M. Schuld, J. Swinarton, Z. Zabaneh, and N. Killoran, "Applications of near-term photonic quantum computers: software and algorithms," *Quantum Science and Technology*, vol. 5, no. 3, p. 034010, 2020.

[11] S. Markidis, "Programming quantum neural networks on nisq systems: An overview of technologies and methodologies," *Entropy*, vol. 25, no. 4, p. 694, 2023.

[12] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[1] https://github.com/smarkidis/Quantum-Physics-Informed-Neural-Network

Quantum optimization of Binary Neural Networks

1st Pietro Torta
International School for Advanced Studies (SISSA)
Trieste, Italy
ptorta@sissa.it

2nd Guglielmo Lami
International School for Advanced Studies (SISSA)
Trieste, Italy
glami@sissa.it

3rd Juan Carrasquilla
Vector Institute, MaRS Centre
Toronto, Canada
carrasqu@vectorinstitute.ai

4th Mario Collura
International School for Advanced Studies (SISSA)
Trieste, Italy
mcollura@sissa.it

5th Estelle Inack
Perimeter Institute for Theoretical Physics
Waterloo, Canada
einack@perimeterinstitute.ca

***Abstract*—We design a method to optimize classical binary neural networks (BNNs) with a hybrid classical-quantum optimization scheme. This approach allows to optimize binary weights, biases and model hyperparameters in the same loop, effectively training different classical BNNs in quantum superposition. We perform large-scale simulations via sampling a quantum circuit described as a Matrix Product State. By repeatedly measuring the final quantum state, one can obtain many *different* optimal BNNs.**

***Index Terms*—Binary neural networks, Variational Quantum Algorithms, Tensor Networks, Matrix Product States, Quantum circuits**

I. Introduction

Binary neural networks (BNNs) are considered a powerful paradigm in the field of artificial intelligence and machine learning [1], [2]. They have gained significant attention because of their potential ability to address some key challenges. Instead of traditional neural networks, BNNs represent and process data using only binary variables ±1, thus greatly reducing memory and computational cost. This improvement can be crucial for resource-constrained devices, such as embedded systems. Other benefits arise from energy efficiency and faster inference. However, the discrete nature of the neural weights and biases makes it difficult to apply traditional gradient-based optimization methods.

A possible way to overcome such limitations relies on the use of Hybrid Classical-Quantum Optimization Techniques [3]–[5]. Such methods are considered one of the main practical applications of quantum devices in the current Noisy Intermediate-Scale Quantum (NISQ) era. They rely on an appropriate parameterized quantum circuit, whose parameters are optimized in order to minimize a suitable loss function (corresponding to a the expectation value of a given hamiltonian). The optimal parameters are learned in a loop with a classical computer, which evaluate the parameter updates at each step.

The binary nature of BNNs parameters enables a straightforward embedding of the system into any qubits based quantum platform [3]–[5]. Interestingly, in such a framework one can also optimize some (binary) *hyper*parameters controlling the architecture itself or the activation functions [5]. This fact can represent a huge breakthrough compared with standard classical techniques. Indeed, as a matter of fact, the quantum superposition principle allows the optimization of many neural networks at once.

II. Model and methods

We consider a neural network having binary weights and biases $w_i, b_i \in \pm 1$. Our model includes also additional binary variables representing a set of relevant hyperparameters, as for instance an architectural choice or the selection of the activation function (see Figure 1). The vector of all binary parameters is dubbed $\boldsymbol{\sigma}$. In a supervised learning framework, the learning of a task is obtained by optimizing a suitable loss function $\mathcal{L}_{\mathrm{cl}}$ over the parameters $\boldsymbol{\sigma}$. Given a data-set $\mathcal{D} = \{(\boldsymbol{x}^{(\mu)}, \boldsymbol{y}^{(\mu)})\}_{\mu=1}^{\mathcal{N}}$, consisting of a series of inputs $\boldsymbol{x}$ and corresponding outputs $\boldsymbol{y}$, the loss is expressed as

$$\mathcal{L}_{\mathrm{cl}}(\boldsymbol{\sigma}) = \frac{1}{\mathcal{N}} \sum_{\mu=1}^{\mathcal{N}} \ell\big(\,\mathrm{BNN}(\boldsymbol{x}^{(\mu)};\boldsymbol{\sigma}),\, \boldsymbol{y}^{(\mu)}\big)\,. \tag{1}$$

Here, $\mathrm{BNN}(\boldsymbol{x};\boldsymbol{\sigma})$ is the function returning the output of the binary neural network for an input $\boldsymbol{x}$ and a fixed set of (hyper)parameters $\boldsymbol{\sigma}$. Notice that $\boldsymbol{x}^{(\mu)}, \boldsymbol{y}^{(\mu)}$ are not constrained to be binary numbers. The energy landscape defined by $\mathcal{L}_{\mathrm{cl}}(\boldsymbol{\sigma})$ is highly non-convex.

To encode the problem in a quantum system, we maps the classical binary variables $\sigma_i \in \pm 1$ into Pauli operators

$\hat{\sigma}_i^z$ acting on a set of qubits (or spin-1/2 variables). Hence, we regard the initial cost function as a quantum Hamiltonian being diagonal in the standard computational basis of quantum computation

$$\mathcal{L}_{\text{cl}}(\sigma_1, \dots, \sigma_N) \to \hat{H}(\hat{\sigma}_1^z, \dots, \hat{\sigma}_N^z) \, , \tag{2}$$

where N is the total number of (hyper)parameters and correspondingly the total number of qubits. At this point, to learn means finding the ground-state of $\hat{H}$, namely minimizing the expectation value $\langle \psi | \hat{H} | \psi \rangle$ over the normalized wave functions $|\psi\rangle$. Notice that, since the $|\psi\rangle$ can be an arbitrary complex superposition of classical binary strings, the optimizer can explore at the same time many possible parameter configurations, or even different neural architectures and activation functions. To perform the quantum optimization, we consider a subset of states, which are obtained from the trivial initial state[1] $|\mathbf{0}\rangle = |0\rangle^{\otimes N}$ by acting with a parameterized quantum circuit, i.e. $|\psi_{\boldsymbol{\theta}}\rangle = \hat{U}(\boldsymbol{\theta}) \, |\mathbf{0}\rangle$, $\boldsymbol{\theta}$ being the set of parameters of the circuit and $\hat{U}(\boldsymbol{\theta})$ the corresponding unitary operator. Our quantum loss function is therefore

$$\mathcal{L}(\boldsymbol{\theta}) = \langle \psi_{\boldsymbol{\theta}} | \hat{H} | \psi_{\boldsymbol{\theta}} \rangle = \sum_{\boldsymbol{\sigma}} \mathcal{L}_{\text{cl}}(\boldsymbol{\sigma}) \, p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \, , \tag{3}$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = | \langle \boldsymbol{\sigma} | \psi_{\boldsymbol{\theta}} \rangle |^2$ is the probability to observe the system in the classical string $\boldsymbol{\sigma}$ after measuring $\hat{\sigma}_i^z$. We restrict ourselves to the simple hardware efficient quantum circuit, which consists of layers of single qubit rotations alternated by entangling layers of CNOTs (see Figure 1). The depth of the circuit is dubbed P. In an experimental setup, $\mathcal{L}(\boldsymbol{\theta})$ can be approximated by taking N_s samples (snapshots) from the wave function $|\psi_{\boldsymbol{\theta}}\rangle$ and using the unbiased estimator

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}) = \frac{1}{N_s} \sum_{s=1}^{N_s} \mathcal{L}_{\text{cl}}(\boldsymbol{\sigma}_s) \, . \tag{4}$$

To investigate the performance of this approach, we consider the MNIST data set, so that from now on the vector $\boldsymbol{x}$ will represent the input image and $\boldsymbol{y}$ the corresponding (one-hot encoded) label. In some cases, we considered a scaled-down version of the data set in which the original pictures (of size 28 × 28) were compressed to $L \times L$ pixels. We choose the standard cross-entropy loss function. To optimize the parameters $\boldsymbol{\theta}$ we use the Quantum Natural Simultaneous Perturbation Stochastic Approximation (QN-SPSA) optimizer [6]. This approach consists in a Natural Gradient Descent, where both the loss function gradient and the metric tensor are replaced with stochastic estimators obtained by sampling random directions in the parameter space. Interestingly, the computational cost of QN-SPSA is independent of the number of parameters, thus being extremely advantageous in the case of a circuit with a large number of parameters.

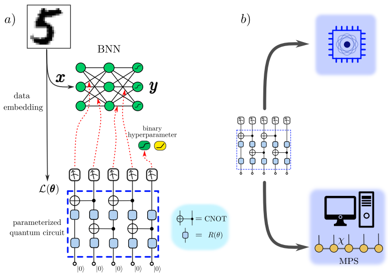


Fig. 1. *a)* The problem of optimizing a BNN based on a given dataset $\{(\boldsymbol{x}^{(\mu)}, \boldsymbol{y}^{(\mu)})\}_{\mu=1}^{\mathcal{N}}$ is considered. A quantum circuit depending on a set of parameters $\boldsymbol{\theta}$ is trained to minimize the loss function $\mathcal{L}(\boldsymbol{\theta})$. After the training, the readouts of the qubit measurements are interpreted as parameters or hyperparameters of the BNN. *b)* We consider two possible scenarios. In the first, the circuit is implemented on a quantum device. In the second, the circuit is simulated on a classical computer with MPS techniques. The maximum MPS bond-dimension χ is regarded as an effective regularization parameters for the optimization.

A. Matrix Product States implementation

Matrix Product States (MPS) provide a compact and flexible representation of quantum states, enabling efficient simulations [7], [8]. Recently, they have been shown to be very useful also in the context of Machine Learning [9]–[11] and hybrid quantum-classical algorithms [12]. For the present work, we use MPS to implement an efficient classical simulator of the outlined quantum circuit $\hat{U}(\boldsymbol{\theta})$.

A crucial parameter of an MPS is its *bond-dimension* (usually dubbed as χ), namely the size of the matrices involved in the wave function representation [7], [8]. The bond-dimension determines the accuracy and computational efficiency of MPS calculations. From a physical point of view, setting a maximum bond-dimension corresponds to having a cut-off on the maximum amount of *entanglement* between the quantum constituents of the system which can be encoded by the ansatz [7]. In general, to simulate a quantum circuit of depth P with full accuracy, one has to scale χ as $\sim \exp(P)$. However, in the cases in which the entanglement growth is bounded one can use a reasonable value of χ still obtaining a good fidelity. For our purpose, here we investigate also the case in which the maximum bond-dimension χ is set to a small value, thus obtaining a very rough approximation of the "true" variational quantum state $|\psi_{\boldsymbol{\theta}}\rangle = \hat{U}(\boldsymbol{\theta}) \, |\mathbf{0}\rangle$. This approach is meaningful in the perspective of creating a (quantum-inspired) *classical optimization algorithm*, i.e. when one is not interested in having a perfect emulator of the quantum circuit but rather in obtaining an effective optimization tool. Similar approaches have been recently explored in several works [3], [9], [13], [14]. In a sense, we interpret the bond-dimension χ as a sort of "regularization" parameter of the simulation, which

[1] We use the standard convention in quantum computation, identifying the two eigenstates of $\hat{\sigma}^z$, up $|\uparrow\rangle$ and down $|\downarrow\rangle$, with the computational basis states $|0\rangle$ and $|1\rangle$ respectively.

interpolates the true quantum algorithm, for $\chi \sim \exp(P)$, with a sort of fictitious classical optimization machine, for $\chi \ll \exp(P)$ (see Figure 1 *b*)).

Finally, let us mention that one can achieve an unbiased sampling of the MPS wave function with a computation cost of $\mathcal{O}(\chi^2 N_s N)$, where N_s is the number of samples (snapshots) and N the number of qubits [15], [16]. This sampling is particularly useful for our model since, contrary to the case of a single layer network [3], for deep architectures one is never able to recast the loss function as a Matrix Product Operator (MPO) and consequently to compute the expectation value $\langle \psi_{\boldsymbol{\theta}} | \hat{H} | \psi_{\boldsymbol{\theta}} \rangle$ with the efficiency guaranteed by the Tensor Network formalism [3]. Indeed, here we use the sampling to approximate $\mathcal{L}(\boldsymbol{\theta})$ through the estimator in Eq. 4 (regardless of the complexity of the loss function). Notice that, unlike standard neural networks methods, here the optimization is performed over the entire data set.

III. Results

We report some preliminary findings of our simulations. New results will be obtained soon and discussed in the new version of the draft.

Figure 2 represents the convergence of the optimization loop for the reduced MNIST dataset with $L = 8$ and 0/1 images only. We use a circuit of depth $P = 2$ trained to optimize a single-layer logistic classifier. The total number of network parameters and correspondingly of qubits is $N = 65$. At this stage, we fix a large values of the MPS bond-dimension, in order to obtain a full accuracy in the quantum circuit simulator. Afterwards, we run the QN-SPSA algorithm for 1500 iteration steps.

At each step of the optimization, we estimate the cross entropy loss with a set of $N_s = 1500$ samples, and we store the *best* 1, 5, 10% of these classical configurations (corresponding to the classical BNNs with lowest cross entropy loss $\mathcal{L}_{\text{cl}}(\boldsymbol{\sigma}_s)$). Indeed, obtaining a sizable fraction of the shots with a small value of the loss function is already a successful result for the optimization. Indeed, these BNNs are expected to yield high test-set classification accuracy, and this may even lead to an early-stop of the optimization loop. This is confirmed in Fig. 2: here, we plot the test-set accuracy during the training, computed at each iteration as an average over the whole sample, or only over these subsets of "best shots". After few iterations, i.e. much before the cross entropy reaches convergence (inset), the best sampled configurations already yield a nearly-optimal test accuracy.

We find $\tilde{\mathcal{L}}(\boldsymbol{\theta}^*) \simeq 0.14 \pm 0.08$ for the optimal circuit parameters $\boldsymbol{\theta}^*$. By extracting $N_s = 10^4$ shots from the final state $|\psi(\boldsymbol{\theta}^*)\rangle$ we observe that $\approx 62\%$ of them correspond to a test accuracy larger than 0.9 (90% of correctly classified images) and $\approx 2.4\%$ of them correspond to a test accuracy larger than 0.95 (95% of correctly classified images). Interestingly, the circuit does not just learn a single optimal setting of the network parameters, but rather a batch of solutions, reminiscent of Bayesian approaches. Indeed, even by measuring the final optimal state $N_s = 10^4$ times, all sampled BNNs are different. More quantitatively, we evaluate the mutual Hamming distance between optimal configurations. We find an average Hamming distance of $\simeq 22$ for the samples with test accuracy larger than 0.9 and $\simeq 18$ for the samples with test accuracy larger than 0.95. In conclusion, the optimal quantum state does not collapse on one or few classical BNNs, and one may interpret its square modulus as a probability distribution that concentrates on optimal BNNs architectures. Remarkably, for sufficiently large quantum circuits, this distribution would be classically intractable.

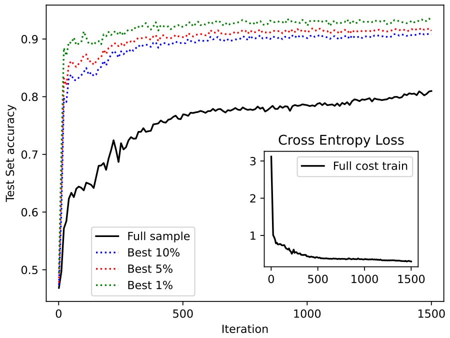


Fig. 2. The training of a circuit op depth $P = 2$ learning a classification network for the 8×8 MNIST dataset. We plot the test set accuracy of the 1, 5, 10% best shots and the loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta})$ at each step.

References

[1] Chunyu Yuan and Sos S. Agaian. A comprehensive review of binary neural network. *Artificial Intelligence Review*, mar 2023.

[2] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1, 2016.

[3] Guglielmo Lami, Pietro Torta, Giuseppe E. Santoro, and Mario Collura. Quantum annealing for neural network optimization problems: A new approach via tensor network simulations. *SciPost Physics*, 14(5), may 2023.

[4] Pietro Torta, Glen B. Mbeng, Carlo Baldassi, Riccardo Zecchina, and Giuseppe E. Santoro. Quantum Approximate Optimization Algorithm applied to the binary perceptron, 2021.

[5] Juan Carrasquilla, Mohamed Hibat-Allah, Estelle Inack, Alireza Makhzani, Kirill Neklyudov, Graham W. Taylor, and Giacomo Torlai. Quantum HyperNetworks: Training Binary Neural Networks in Quantum Superposition, 2023.

[6] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner. Simultaneous Perturbation Stochastic Approximation of the Quantum Fisher Information. *Quantum*, 5:567, oct 2021.

[7] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, jan 2011.

[8] Pietro Silvi, Ferdinand Tschirsich, Matthias Gerster, Johannes Jünemann, Daniel Jaschke, Matteo Rizzi, and Simone Montangero. The tensor networks anthology: Simulation techniques for many-body quantum lattice systems. *SciPost Physics Lecture Notes*, mar 2019.

[9] James Dborin, Fergus Barratt, Vinul Wimalaweera, Lewis Wright, and Andrew G Green. Matrix product state pre-training for quantum machine learning. *Quantum Science and Technology*, 7(3):035014, may 2022.

[10] E. Miles Stoudenmire and David J. Schwab. Supervised learning with quantum-inspired tensor networks. 2016.

[11] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3), jul 2018.

[12] Rishi Sreedhar, Pontus Vikstål, Marika Svensson, Andreas Ask, Göran Johansson, and Laura García-Álvarez. The quantum approximate optimization algorithm performance with low entanglement and high circuit depth, 2022.

[13] Rishi Sreedhar, Pontus Vikstål, Marika Svensson, Andreas Ask, Göran Johansson, and Laura García-Álvarez. The Quantum Approximate Optimization Algorithm performance with low entanglement and high circuit depth, 2022.

[14] E. Miles Stoudenmire and David J. Schwab. Supervised Learning with Quantum-Inspired Tensor Networks, 2017.

[15] E M Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, may 2010.

[16] Andrew J. Ferris and Guifre Vidal. Perfect sampling with unitary tensor networks. *Phys. Rev. B*, 85:165146, Apr 2012.

Compiling Quantum Circuits for the Graph Coloring Problem*

Angelo Oddi, Riccardo Rasconi
ISTC - Instituto di Scienze e Tecnologie della Cognizione
CNR - Consiglio Nazionale delle Ricerche
Roma, Italy
{angelo.oddi, riccardo.rasconi}@istc.cnr.it

Marco Baioletti
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Perugia, Italy
marco.baioletti@unipg.it

***Abstract*—Current quantum computing technologies limit the qubit interaction distance allowing the execution of gates between adjacent qubits only. This has opened the way to the exploration of possible techniques aimed at guaranteeing nearest-neighbor (NN) compliance in any quantum circuit through the addition of a number of so-called *swap* gates between adjacent qubits. In addition, technological limitations (*decoherence* effect) impose that the overall duration (i.e., *depth*) of the quantum circuit realization be minimized. In this work we investigate the application of an upgraded version of the greedy randomized search (GRS) technique originally introduced in the literature that synthesises NN-compliant quantum circuits realizations, starting from a set of benchmark instances of different size belonging to the *Quantum Approximate Optimization Algorithm* (QAOA) class tailored for the Graph Coloring problem. We propose a comparison between the presented method and the SABRE compiler, one of the best-performing compilation procedures present in Qiskit, an open-source SDK for quantum development, both from the CPU efficiency and from the solution quality standpoint.**

***Index Terms*—randomized search, quantum circuit compilation, planning, scheduling, optimization**

I. Introduction

Quantum algorithms process information represented as qubits, the basic unit of quantum information, and quantum operations (called gates) are the building blocks of quantum algorithms. In order to be run on real quantum computing hardware, quantum algorithms must be compiled into a set of elementary machine instructions (or *gates*).

Since the currently available Noisy Intermediate-Scale Quantum (NISQ) devices [2] suffer a number of technological problems such as noise and *decoherence*, it is important that the process that carries out the quantum computation be somehow adapted to the physical limitations of the quantum hardware of interest, by means of a proper compilation. Usually, NISQ algorithms require error mitigation techniques to recover useful data, which however make use of precious qubits to be implemented. Thus, the creation of a computer with tens of thousands of qubits and sufficient error correction capabilities would eventually end the NISQ era. These "beyond-NISQ" devices would be able, for example, to implement Shor's algorithm, for very large numbers, and break RSA encryption. Until that point however, the need to produce circuits runnable in the current (or near-future) quantum architectures in a reasonably reliable manner (i.e., counting on noise minimization techniques rather than on error-correcting techniques) will stand. Hence, the need to provide quantum circuit compilation procedures that minimize the effects of decoherence by minimizing the circuit's depth.

In this work, we investigate the performance of an upgraded version of the greedy randomized search (GRS) technique [3]–[5] originally introduced in [6] applied to the problem of compiling quantum circuits to emerging quantum hardware. In particular, we experiment on a set of benchmark instances belonging to the Quantum Alternate Operator Ansatz (QAOA) class [7] tailored for the Graph Coloring problem, and devised to be executed on top of a hardware architecture inspired by Rigetti Computing Inc. [8] (see Figure 1). We compare our algorithm's performance against the SABRE compiler [9], one of the best compilers present in the Qiskit framework, and demonstrate the superiority of our approach.

II. The QCC problem

The problem tackled in this work consists in compiling a given quantum circuit on a specific quantum hardware architecture. To this aim, we focus on the Quantum Alternating Operator Ansatz (QAOA) framework [7] a generalization of the *Quantum Approximate Optimization Algorithm* (QAOA) circuits [10], [11], a class of hybrid quantum algorithms often used in the literature to solve problems like the Max-Cut, for the resolution of the Graph Coloring (GC), a problem that has so far received much less attention. The quantum circuits that solve the benchmark problems considered in this work are characterized by a high number of commuting quantum gates (i.e., gates among which no particular order is superimposed) that allow for great flexibility and parallelism in the solution, which makes the corresponding optimization problem very interesting and allows for an a significant depth minimization potential to limit circuit's decoherence [12].

Given a graph $G(V, E)$ with $n = |V|$ nodes and $m = |E|$ edges, the objective of the GC is to maximize the number

*This work is a summary of the published paper [1], which reports part of the work developed during the joint ARIADNA study *Meta-Heuristic Algorithms for the Quantum Circuit Compilation Problem* with the Advanced Concepts Team (ACT) of the European Space Agency (ESA), ESA Contract No. 4000134995/21/NL/GLC/my.

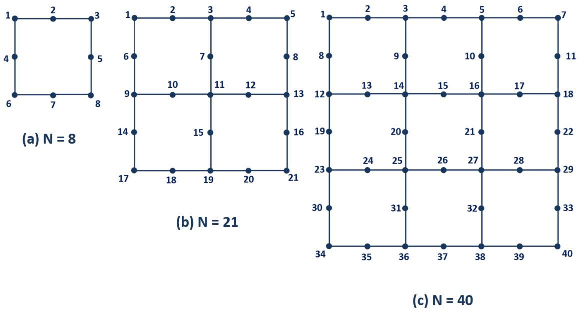


Fig. 1: Three quantum chip designs characterized by an increasing number of qubits ($N = 8, 21, 40$) inspired by Rigetti Computing Inc. Every qubit is located at a different location (node), and the integers at each node represent the qubit's identifier.

of edges in E that have end points with different colours, using for each node one among k available colours ($k > 2$). Similarly to the Max-Cut problem case, the quantum state preparation circuit within the QAOA solving framework relative to the Graph Coloring problem is divided in the following ordered phases: (i) *initial state preparation* (INIT block), (ii) *phase-shift* (P-S block), and (iii) *mixing* (MIX block). The detailed description of the QAOA phases cannot be provided for reasons of space; the interested reader may refer to [1].

In order to be executed, a quantum circuit must be mapped on a quantum chip which determines the circuit's hardware architecture specification [13]. The chip can be seen as an undirected multigraph whose nodes represent the qubits (quantum physical memory locations) and whose edges represent the types of gates that can be physically implemented between adjacent qubits of the physical hardware (see Figure 1 as an example of three chip topologies of increasing size). Since a 2-qubit gate requiring two specific qstates can only be executed on a pair of adjacent qubits, the required qstates must be made nearest-neighbors (NN) prior to gate execution. NN-compliance can be obtained by adding a number of SWAP gates so that every pair of qstates involved in the quantum gates can be eventually made adjacent, allowing all gates to be correctly executed.

III. A Greedy Randomized Search Algorithm

In this section we provide a short description of the *Greedy Randomized Search (GRS)* procedure used to compile the circuit introduced in previous Section II. For more details, the reader is referred to [1]. Algorithm 1 depicts the complete randomized search algorithm for generating a near-optimal solutions, which is designed to invoke the CompileCircuit() procedure until a stop criterion is satisfied. It essentially realizes an optimization cycle in which a new solution S is computed at each iteration through the CompileCircuit() algorithm. The optimization process continues until a stopping condition (generally a max time limit) is met, where the *GRS* procedure returns the best solution ever found. As can be readily observed, the efficacy of the *GRS* mainly depends on the efficacy of the CompileCircuit() procedure (described in the following section), which has the task of synthesizing increasingly better solutions.

Algorithm 1 Greedy Randomized Search

```
Require: A problem $P$, stop criterion
  $S_{best} \leftarrow$ CompileCircuit($P$)
  while (stopping criterion not satisfied) do
    $S \leftarrow$ CompileCircuit($P$)
    if (depth(S) $<$ depth($S_{best}$)) then
      $S_{best} \leftarrow S$
    end if
  end while
  return ($S_{best}$)
```

A. Compile Circuit Algorithm

The CompileCircuit() procedure is a heuristically-based iterative algorithm that builds a solution from scratch using a randomized ranking heuristic. This heuristic assigns a score to the gates taking into account the "neighbouring cost" of all the gates that have yet to be inserted in the solution. At each iteration, a subset of gates that guarantee the fastest realization of the neighbouring conditions of all the remaining gates is generated and one gate is selected at random from this subset, for insertion in the current partial solution.

The CompileCircuit() procedure takes as input a *QCCP* problem, and proceeds by selecting and inserting in the *partial solution S* one gate operation at a time until all the gates in the input circuit are in S. The core of the CompileCircuit() procedure is the function SelectExecutableGate() (see [14]), which returns at each iteration either one of the gates of the initial circuit (not yet inserted) or a SWAP gate necessary to guarantee NN-compliance as described in the previous Section II. This function has been explicitly designed for minimizing the solution depth, in particular its implementation is inspired to [15], such that the selection of a gate is based on two criteria: (i) the earliest start time gate selection (a value correlated to depth minimization); (ii) a metric to minimize the number of swaps.

IV. Experimental Evaluation

We have implemented and tested the proposed ideas leveraging the Qiskit open-source quantum-related framework [16], the proposed procedure was implemented in Python in order to allow its integration within Qiskit. The benchmark set for the graph colouring circuits is obtained as an extension of part of the $N8$ benchmark set for the Max-Cut problem [12]. Following the approach in [12], the graph G for which the optimal coloring assignment needs to be found are randomly generated as Erdös-Rényi graphs. In particular, 100 graphs are generated for the $N = 8$ qubit case. Half (50 problems) are generated by choosing N of $N(N-1)/2$ edges over 7 qstates randomly located on the circuit of size 8 qubits (referred as 'Utilization' $u = 90\%$). The other 50 problems are generated

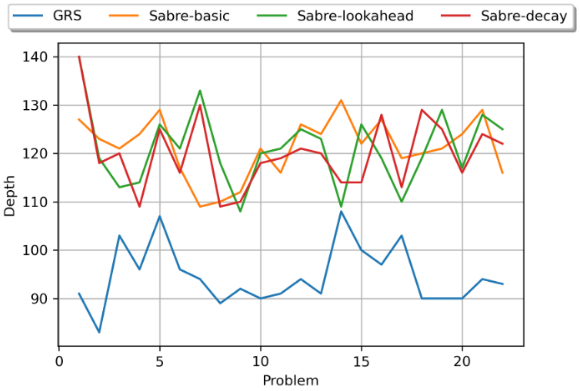


Fig. 2: Comparison between GRS and SABRE

by choosing N edges over 8 qstates - referred as utilization $u = 100\%$). For the graph colouring benchmark, we only consider the $N8$ problems with utilization $u = 100\%$, and such that the connected graph contains exactly 7 nodes, assigning three colours ($k = 3$) to each node of the graph, for a total of 22 graph instance problems. Hence, quantum processors with at least 21 qubits (7 nodes times 3 colours) are necessary for the execution of such instances. More specifically, we consider a Rigetti-inspired 21 qubit processor and set $p = 2$ (two PS-mixing passes). The Python version of the proposed greedy randomized search (*GRS*) algorithm compiles a *QAOA* circuit with the following choices: (i) a *one-hot* encoding to represent the graph-coloring problems [17], and (ii) a decomposition procedure for the *QAOA* blocks based on the identification of odd and even MIX_{XY} gates [7], [18]. Figure 2 compares the proposed GRS algorithm with the SABRE compiler available in Qiskit (*SabreSwap*), launched according to its three different heuristics (*basic*, *lookahead*, and *decay*). The algorithms are compared with respect to the depth of the compiled circuits (the circuit's depth represents the longest path in the compiled circuit graph). For each algorithm, a CPU time limit of 10 seconds is imposed on each run. From the results in Figure 2 it is clear that GRS outperforms SABRE in all the latter's execution modes. One possible explanation for the superiority of GRS is its capability to better exploit the commutativity rules of the gates in the QAOA-based Graph Coloring quantum circuit instances. Indeed, unlike SABRE, GRS imposes no particular order in the selection of the W_N, *P-S*, and MIX_{XY} macro-gates as the solution is built, beyond the precedence constraints originally present in the input quantum circuit.

V. Conclusions

In this paper we have considered the compilation techniques for Noisy Intermediate-Scale Quantum (NISQ) devices [2]. In particular, we have explored the Quantum Alternating Operator Ansatz (QAOA) framework [7] for solving optimization problems and studied the quantum circuits for the Graph Coloring reference problem. We have proposed a greedy randomized search (GRS) algorithm targeted at optimizing the compilation of quantum circuits and defined an original benchmark set for testing compilation algorithms. On the basis of our empirical validation the proposed GRS algorithm outperforms other compilation algorithms available in the Qiskit framework.

References

[1] A. Oddi, R. Rasconi, M. Baioletti, V. G. Santucci, and H. Beck, "Quantum circuit compilation for the graph coloring problem," in *AIxIA 2022 - Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, vol. 13796. Springer, 2022, pp. 374–386. [Online]. Available: https://doi.org/10.1007/978-3-031-27181-6_26

[2] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-08-06-79

[3] J. Hart and A. Shogan, "Semi-greedy heuristics: An empirical study," *Operations Research Letters*, vol. 6, pp. 107–114, 1987.

[4] M. G. Resende and R. F. Werneck, "A hybrid heuristic for the p-median problem," *Journal of Heuristics*, vol. 10, no. 1, pp. 59–88, Jan 2004. [Online]. Available: https://doi.org/10.1023/B:HEUR.0000019986.96257.50

[5] A. Oddi and S. Smith, "Stochastic Procedures for Generating Feasible Schedules," in *Proceedings 14th National Conference on AI (AAAI-97)*, 1997, pp. 308–314.

[6] A. Oddi and R. Rasconi, "Greedy randomized search for scalable compilation of quantum circuits," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, W.-J. van Hoeve, Ed. Cham: Springer International Publishing, 2018, pp. 446–461.

[7] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, Feb 2019. [Online]. Available: http://dx.doi.org/10.3390/a12020034

[8] E. A. Sete, W. J. Zeng, and C. T. Rigetti, "A functional architecture for scalable quantum computing," in *2016 IEEE International Conference on Rebooting Computing (ICRC)*, Oct 2016, pp. 1–6.

[9] G. Li, Y. Ding, and Y. Xie, "Tackling the qubit mapping problem for nisq-era quantum devices," *CoRR*, vol. abs/1809.02573, 2018. [Online]. Available: http://arxiv.org/abs/1809.02573

[10] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, November 2014.

[11] G. G. Guerreschi and J. Park, "Gate scheduling for quantum algorithms," *arXiv preprint arXiv:1708.00023*, July 2017.

[12] D. Venturelli, M. Do, E. Rieffel, and J. Frank, "Temporal planning for compilation of quantum approximate optimization circuits," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4440–4446. [Online]. Available: https://doi.org/10.24963/ijcai.2017/620

[13] D. Maslov, S. M. Falconer, and M. Mosca, "Quantum circuit placement: Optimizing qubit-to-qubit interactions through mapping quantum circuits into a physical experiment," in *Proceedings of the 44th Annual Design Automation Conference*, ser. DAC '07. New York, NY, USA: ACM, 2007, pp. 962–965. [Online]. Available: http://doi.acm.org/10.1145/1278480.1278717

[14] M. Baioletti, R. Rasconi, and A. Oddi, "A Novel Ant Colony Optimization Strategy for the Quantum Circuit Compilation Problem," in *Evolutionary Computation in Combinatorial Optimization - EvoCOP 2021*, ser. LNCS, vol. 12692. Springer, 2021, pp. 1–16. [Online]. Available: https://doi.org/10.1007/978-3-030-72904-2_1

[15] S. Chand, H. K. Singh, T. Ray, and M. Ryan, "Rollout based heuristics for the quantum circuit compilation problem," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 974–981.

[16] "Qiskit: An open-source framework for quantum computing," 2021.

[17] F. G. Fuchs, H. Ø. Kolden, N. H. Aase, and G. Sartor, "Efficient encoding of the weighted max $$k$$-cut on a quantum computer using qaoa," *SN Computer Science*, vol. 2, no. 2, p. 89, 2021. [Online]. Available: https://doi.org/10.1007/s42979-020-00437-z

[18] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, "xy mixers: Analytical and numerical results for the quantum alternating operator ansatz," *Phys. Rev. A*, vol. 101, p. 012320, Jan 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.101.012320

Multi-class classification based on quantum state discrimination

1st Roberto Giuntini
University of Cagliari
Cagliari, Italy
giuntini@unica.it

2nd Giuseppe Sergioli
University of Cagliari
Cagliari, Italy
giuseppe.sergioli@gmail.com

***Abstract*—We present a general framework for the problem of multi-class classification using classification functions that can be interpreted as fuzzy sets. We specialize these functions in the domain of Quantum-inspired classifiers, which are based on quantum state discrimination techniques. In particular, we use unsharp observables (Positive Operator-Valued Measures) that are determined by the training set of a given dataset to construct these classification functions.**

***Index Terms*—Quantum state discrimination, Classification, Pretty good measurement.**

I. Introduction

In previous works, different quantum-inspired algorithms for the classification of classical data have been developed [4]–[7], [9]. These algorithms are based on the idea of finding a representation of classical datasets in terms of quantum representatives. More specifically, each feature-vector of a dataset is encoded in a pure quantum state. This process is called *quantum encoding*. Then, using classical hardware, quantum state discrimination techniques (taken from quantum information theory) are applied to the quantum representatives to find a solution to the classification problem at issue. This is the case, for example, of the *Helstrom quantum classifier* (HQC), which turns out to be particularly beneficial in terms of accuracy for binary classification [8]. The problem, however, is that currently no direct method to extend HQC to multi-class classification is known.

To overcome this difficulty, in this work we take inspiration from the so-called *Pretty Good Measurement* [12] to define a quantum-inspired multi-class classifier. The proposed algorithm first assigns a (not necessarily projective) positive operator-valued measure (POVM) (the Pretty Good Measurement) to a given training dataset. This POVM is then used to assign, via Born's rule, a probability-value to each element of the test dataset. Such a procedure allows one to define a function that *directly* classifies any object of the test dataset, without resorting to the (time consuming) strategies such as *One-vs-One* or *One-vs-All*) that are used for classifiers that do not natively support classification tasks with more than two classes.[1] We refer to this algorithm as the *PGM classifier*. Just like the Helstrom classifier, the PGM classifier, being quantum-inspired, brings a concrete computational advantage when running on fully classical hardware. The main goal of this work is to provide a precise mathematical description of the PGM classifier algorithm.

II. General setting for Quantum-inspired classification

By following standard supervised classification, each object x is associated with a vector $\vec{x}$ (called *object-vector* or *feature-vector*) of a d-dimensional Hilbert space $\mathcal{H}^d$. [2] A *pattern* is defined as a pair $(\vec{x_j}, \lambda_j)$ where $\vec{x_j}$ is a feature-vector and λ_j is the class label which denotes the class which the object is supposed to belong to. For simplicity, we identify the set L of all class-labels with a finite sequence $(1, \ldots, \ell)$ of natural numbers that are in one-to-one correspondence with the ℓ classes which the objects belong to. Thus, a *training dataset* can be represented as a set $\mathcal{S}_{\mathrm{tr}} := \{(\vec{x}_1, \lambda_1), \ldots (\vec{x}_m, \lambda_m)\}$, where $\forall j \in \{1, \ldots, m\}$: $\lambda_j \in L$. Given any class label $i \in L$, we can define the set $\mathcal{S}^i_{\mathrm{tr}}$ of all object-vectors whose associated class label is i:

$$\mathcal{S}^i_{\mathrm{tr}} := \{\vec{x}_j \in \mathcal{S}_{\mathrm{tr}} : \lambda_j = i\}. \tag{1}$$

The cardinality of $\mathcal{S}^i_{\mathrm{tr}}$ is denoted by $|\mathcal{S}^i_{\mathrm{tr}}|$. Clearly, $\sum_{i=1}^{\ell} |\mathcal{S}^i_{\mathrm{tr}}| = m$. The task of supervised classification is to infer a classifier-function (simply, a *classifier*) from the training dataset that assign, as accurately as possible, a class-label to any object-vector $\vec{x}$. Formally, a classifier can be defined as a map $Cl : \mathbb{C}^d \to L$.

Let $\mathcal{S}_{\mathrm{tr}} = \{(\vec{x}_1, \lambda_1), \cdots, (\vec{x}_m, \lambda_m)\}$ be a training dataset. In order to define a classifier (based on $\mathcal{S}_{\mathrm{tr}}$) one defines a map f (called "learning function") that associates to any feature-vector $\vec{x}$ a sequence of ℓ-numbers belonging to the unit real-interval $[0, 1]$: $f : \mathbb{C}^d \to [0, 1]^\ell$. The i^{th}-component of $f(\vec{x})$ will be denoted by $f(\vec{x})_i$.

The interpretation of the $i-$th component of $f(\vec{x})$ depends on the intended meaning of the function f itself. For example, if one adopts a "fuzzy perspective", the value $f(\vec{x})_i$ may

This work is partially supported by the following projects: 1) MIUR, project PRIN 2017: Theory and applications of resource sensitive logics (code: 20173WKCM5). 2) MIUR project PRIN 2017: Logic and Cognition: theory, experiments, applications (code: 20173YP4N3). 3) FdS project: Ubiquitous Quantum Reality (UQR): understanding the natural processes under the light of Quantum-like structures.

[1] A numerical comparison between these approaches is analyzed in a separate work [2].

[2] Unlike standard representations in machine learning, we do not exclude features that can be represented as complex numbers.

represent the *degree of membership* of the object x (whose object-vector is $\vec{x}$) to the class labeled by λ_i. Thus, any label i determines a *fuzzy set* $f_i : \mathbb{R}^d \to [0,1]$ such that for any $\vec{x} \in \mathbb{R}^d$, $f_i(\vec{x}) = f(\vec{x})_i$. In a probabilistic framework, instead, $f(\vec{x})$ is assumed to be a *probability-vector* (i.e. $\sum_{i=1}^{\ell} f(\vec{x})_i = 1$) and the value $f(\vec{x})_i$ can be interpreted as the probability that the object x (with associated feature-vector $\vec{x}$) belongs to the class labeled by i.

The *classifier determined by* f (or simply, the f*-classifier*) is the map $Cl_f : \mathbb{C}^d \to L$ that assigns to any feature-vector $\vec{x} \in \mathbb{C}^d$ the class-label that is associated to the greatest value of $f(\vec{x})_i$, with $1 \leq i \leq \ell$.

Since it may happen that f returns more than one class-label when there are matching $f(\vec{x})_i$ values, we pose by convention

$$Cl_f(\vec{x}) := \min\{i \in L \,:\, f(\vec{x})_i = \max\{f(\vec{x})_k \,:\, 1 \leq k \leq \ell\}\}\,. \tag{2}$$

A classifier Cl_f is called *probabilistic* iff for any $\vec{x} \in \mathbb{C}^d$ we have: $\sum_{i=1}^{\ell} f(\vec{x})_i = 1$. In other words, a classifier is probabilistic iff for any $\vec{x}$ the sequence $(f(\vec{x})_1, \ldots, f(\vec{x})_\ell)$ is a probability-vector.

The task of supervised classification is to infer a classifier-function (simply, a *classifier*) from the training dataset that assign, as accurately as possible, a class-label to any object-vector $\vec{x}$.

Formally, a classifier can be defined as a map

$$Cl : \mathbb{C}^d \to L.$$

Given a training dataset, the construction of a quantum classifier is based on three basic steps: i) applying a *quantum feature map* (or *quantum encoding*) to encode the object-vectors of the training dataset into quantum objects [7]; ii) finding a suitable learning function f that determines the quantum classifier; iii) applying the quantum classifier to a quantum-encoded *object-vectors* to obtain the labels of the classes which the objects belong to.

Let us consider a training dataset $\mathcal{S}_{\mathrm{tr}} = \{(\vec{x}_1, \lambda_1), \cdots, (\vec{x}_m, \lambda_m)\}$. A quantum encoding is a map that associates with any object-vector $\vec{x}$ of $\mathbb{C}^d$ a pure quantum state (called *object quantum-state*) $\rho_{\vec{x}}$ of a Hilbert space $\mathbb{C}^n$, whose dimension n depends on the number of the d-features. Given a quantum encoding $\vec{x} \mapsto \rho_{\vec{x}}$, a *quantum pattern* is any pair $(\rho_{\vec{x}_j}, \lambda_j)$. A *quantum training dataset* is defined as the set of all quantum patterns:

$$\mathcal{S}_{\mathrm{Qtr}} := \{(\rho_{\vec{x}_1}, \lambda_1), \ldots, (\rho_{\vec{x}_m}, \lambda_m)\}\,.$$

Given any class label $i \in L$, we can also define the set $\mathcal{S}^i_{\mathrm{Qtr}}$ as the set of all object quantum-states $\rho_{\vec{x}_j}$ that are associated to the set $\mathcal{S}^i_{\mathrm{tr}}$ of all i-object-vectors:

$$\mathcal{S}^i_{\mathrm{Qtr}} := \{\rho_{\vec{x}_j} \,:\, \vec{x}_j \in \mathcal{S}^i_{\mathrm{tr}}\}. \tag{3}$$

Similarly to the case of the Nearest-Mean Classifier where one associates to any class of object-vectors its centroid, in our quantum-inspired approach, one can define the crucial notion of *quantum centroid*. Intuitively, the quantum centroid associated to the class-label i is the density operator (in an appropriate Hilbert space) that represents the uniform sampling of all i-object quantum states.

Definition 2.1: Let $i \in L$ be a class-label. The quantum centroid associated to i (denoted by $\rho_{(i)}$) is:

$$\rho_{(i)} := \frac{1}{|\mathcal{S}^i_{\mathrm{Qtr}}|} \sum_{\vec{x}_j \in \mathcal{S}^i_{\mathrm{tr}}} \rho_{\vec{x}_j},$$

where $|\mathcal{S}^i_{\mathrm{Qtr}}|$ is the cardinality of $\mathcal{S}^i_{\mathrm{Qtr}}$. Clearly, $|\mathcal{S}^i_{\mathrm{Qtr}}| = |\mathcal{S}^i_{\mathrm{tr}}|$.

As one can easily realize, the ℓ class-labels are in one-to-one correspondence with the set $\{\rho_{(1)}, \ldots, \rho_{(\ell)}\}$ of all quantum centroids.

Let $\mathcal{S}_{\mathrm{tr}} = \{(\vec{x}_1, \lambda_1), \ldots (\vec{x}_m, \lambda_m)\}$ be a quantum training dataset. How to define a possible "quantum learning function" f in terms of $\mathcal{S}_{\mathrm{Qtr}}$? As happens in the classical case, different answers are possible. In our approach we take the move to interpret the set of all class-labels as possible outcomes of a measurement. Let us briefly recall the notion of (quantum) measurement.

Let $\mathcal{B}(\mathcal{H})^+$ be the set of all positive semidefinite operators acting on a finite dimensional Hilbert space $\mathcal{H}$. A *measurement* is defined as a map $\mathcal{M}$ from a finite non-empty set $\mathcal{O}$ (representing a set of possible outcomes of a physical quantity) into $\mathcal{B}(\mathcal{H})^+$ such that $\sum_{i \in \mathcal{M}} \mathcal{O}(i) = \mathbb{I}$. A measurement is said to be a *von Neumann measurement* iff every $\mathcal{M}(i)$ is a projection, i.e., $\mathcal{M}(i)^* = \mathcal{M}(i)\mathcal{M}(i) = \mathcal{M}(i)$, where $\mathcal{M}(i)^*$ is the adjoint of $\mathcal{M}(i)$.

A *quantum classifier* is a classifier Cl_f, where the function f is determined by a measurement $\mathcal{M} : L \to \mathcal{B}(\mathcal{H})^+$ (see [12], Definition 2.34). More precisely:

Definition 2.2: A *quantum classifier* is a classifier Cl_f (see Equation(2)) such that the learning function $f : \mathbb{C}^d \to [0,1]^\ell$ satisfies the following condition: there exists a measurement $\mathcal{M} : L \to \mathcal{B}(\mathcal{H})^+$ such that

$$\forall \vec{x} \in \mathbb{C}^d : \, f(\vec{x})_i = \mathrm{tr}(\mathcal{M}(i)\rho_{\vec{x}}),$$

where tr is the trace of a matrix.

Intuitively, the i-th component $f(\vec{x})_i$ of the learning function f represents the probability that the object encoded by $\rho_{\vec{x}}$ belongs to the class i.

An interesting question is whether classification accuracy can be improved by increasing the dimension of the state space where the object quantum-states live.

Although computation in a larger feature space generally increases runtime, the expected improvement in prediction accuracy is crucial in certain machine learning applications, such as those specialized in medical diagnosis.

In our case, the dimensional increasing of the feature space is obtained by encoding any object-vector $\vec{x}$ as a tensor product $\rho_{\vec{x}} \underbrace{\otimes \ldots \otimes}_{\text{n-times}} \rho_{\vec{x}}$ of the object quantum-states $\rho_{\vec{x}}$. Accordingly, the set of all object quantum-states that are associated to the

n-copies of all i-objects is defined by tensorizing the object quantum-states in $\mathcal{S}^i_{\mathrm{Qtr}}$:

$$\mathcal{S}^{i^{(n)}}_{\mathrm{Qtr}} := \{\rho_{\vec{x}_j} \underbrace{\otimes \ldots \otimes}_{n\text{-times}} \rho_{\vec{x}_j} \,:\, \vec{x}_j \in \mathcal{S}^i_{\mathrm{tr}}\}.$$

Accordingly, the n-copy quantum centroid of $\mathcal{S}^{i^{(n)}}_{\mathrm{Qtr}}$ can be defined as follows:

$$\rho^{(n)}_{(i)} := \frac{1}{|\mathcal{S}^i_{\mathrm{Qtr}}|} \sum_{\vec{x}_j \in \mathcal{S}^i_{\mathrm{tr}}} \rho_{\vec{x}_j} \underbrace{\otimes \ldots \otimes}_{n\text{-times}} \rho_{\vec{x}_j}. \tag{4}$$

It should be noticed that, in general, $\rho^{(n)}_{(i)} \neq \underbrace{\rho_{(i)} \otimes \ldots \otimes \rho_{(i)}}_{\text{n-times}}$.

The $\otimes^n$-generalization of the quantum classifier introduced in Definition 2.2 can now be naturally defined as a function $f : \mathbb{C}^{d^n} \to [0,1]^\ell$ determined by a measurement $\mathcal{M} : L \to \mathcal{B}(\otimes^n \mathbb{C}^d)^+$. As we will show in the sequel, this procedure turns out to be advantageous in improving classification accuracy.

III. Pretty Good Measurement and multi-class classification

Given an ensemble of possible states (generally more than two) with their respective a priori probabilities

$$R = \{(\mathrm{p}_1, \rho_1), \cdots, (\mathrm{p}_\ell, \rho_\ell)\}, \tag{5}$$

it may be difficult to find an analytical description for the exact optimal measurement $Opt(R)$ associated to R. One possible solution is to search for a sub-optimal measurement that can be expressed in an analytical form. This is known as the so-called *Pretty Good Measurement* [12], which we will introduce in the following. Let us define the average state of R as

$$\sigma = \sum_{i=1}^{\ell} \mathrm{p}_i \rho_i.$$

For each $i : 1 \leq i \leq \ell$, let us define the following operator

$$E_i = (\sigma^\urcorner)^{1/2} \mathrm{p}_i \rho_i (\sigma^\urcorner)^{1/2},$$

where $\sigma^\urcorner$ is the pseudoinverse (or Moore-Penrose inverse) of σ. It can be seen that $\sum_{i=1}^{\ell} E_i = P_{im(\sigma)}$, where $P_{im(\sigma)}$ is the projection onto the subspace spanned by the image of σ. The set $\{E_i\}_{i=1}^{\ell}$ does not determine a measurement since in general $\sum_{i=1}^{\ell} E_i < \mathbb{I}$. For each i with $1 \leq i \leq \ell$, let us define the following operators: $F_i := E_i + \frac{1}{\ell} P_{ker(\sigma)}$, where $P_{ker(\sigma)}$ is the projection onto the subspace spanned by the kernel of σ. It turns out that the map $\mathcal{F} : \{1, 2, \cdots, \ell\} \to \mathcal{B}(\mathbb{C}^n)^+$ such that for any $i \in \{1, \ldots, \ell\}$ $\mathcal{F}(i) = F_i$ is a measurement, called Pretty Good Measurement or, shortly, PGM. It turns out the PGM $\mathcal{F}$ is *sub-optimal* since $\mathrm{p}^{\mathcal{F}}_{succ}(R) \geq Opt(R)^2$. Let us now turn to the general problem of multi-class classification. After the quantum encoding procedure, we consider the quantum training datasets S^i_{Qtr} as defined in Equation (3) and their respective quantum centroids $\rho_{(i)}$ as defined in Definition 2.1. Hence, it is possible to consider the ensemble R as defined in Equation (5) where, as in the Helstrom case, the a priori probability-values are assumed to be equal, i.e., for any $i \in \{1, \ldots, \ell\} : p_i = \frac{1}{\ell}$. Thus, we can associate to R a Pretty Good Measurement. In this case, the learning function f is defined as follows:

$$\forall \vec{x} \in \mathbb{C}^d, \forall i \in \{1, \ldots, \ell\} : \, f(\vec{x})_i := \mathrm{tr}(F_i \rho_{\vec{x}}).$$

According to Definition 2.2, the multi-class quantum classifier determined by f, called *Pretty Good Classifier* (*PGM classifier*), is defined as follows:

$$\begin{aligned} Cl_f(\vec{x}) := \min\{& i \in \{1, \cdots, \ell\} : \\ & \mathrm{tr}(F_i \rho_{\vec{x}}) = \max\{\mathrm{tr}(F_k \rho_{\vec{x}}) \,:\, 1 \leq k \leq \ell\}\}. \end{aligned}$$

Notice that if $P_{ker(\sigma)} = \mathbb{O}$, we can replace F_i by E_i in the above equation.

We can also generalize this framework by taking the tensor product of n-copies of states. Thus, the definition of the multi-class quantum classifier can be naturally extended as:

$$\begin{aligned} Cl_f(\vec{x}) := \min\{& i \in \{1, \cdots, m\} : \\ & \mathrm{tr}(F^{(n)}_i \rho^{(n)}_{\vec{x}}) = \max\{\mathrm{tr}(F^{(n)}_k \rho^{(n)}_{\vec{x}}) : 1 \leq k \leq \ell\}\}. \end{aligned}$$

As a final remark, let us notice that the PGM not only performs very well in terms of accuracy, but also avoids to decompose an n-ary classification into a combinatorial number of binary classifications, as required by the standard “One versus One” or “One versus Rest” procedures. A detailed experiment comparing the performance of PGM with other standard classifiers is presented in [2].”

References

[1] C.W. Helstrom, Quantum detection and estimation theory, Journal of Statistical Physics 1 (1969) 23152.

[2] R. Giuntini, F. Holik, D.K. Park, H. Freytes, C. Blank, G. Sergioli, A quantum-inspired algorithm for direct multi-class classification, Applied Soft Computing 134 (2023) 109956.

[3] O.H. Montiel Ross, A Review of Quantum-Inspired Metaheuristics: Going From Classical Computers to Real Quantum Computers, IEEE Access 8 (2020) 814-838.

[4] E. Santucci, G. Sergioli, Classification Problem in a Quantum Framework, 2017.

[5] G. Sergioli, G.M. Bosyk, E. Santucci, R. Giuntini, A Quantum-inspired version of the classification problem, International Journal of Theoretical Physics 56 (2017) 3880-3888.

[6] G. Sergioli, E. Santucci, L. Didaci, J.A. Miskczak, R. Giuntini, A Quantum-inspired version of the Nearest Mean Classifier, Soft Computing 22 (2018) 691-705.

[7] G. Sergioli, G. Russo, E. Santucci, A. Stefano, S. E. Torrisi, S. Palmucci, C. Vancheri, R. Giuntini, Quantum-inspired minimum distance classification in a biomedical contexts, International Journal of Quantum Information 16 (2018) 1840011.

[8] G. Sergioli, R. Giuntini, H. Freytes, A new quantum approach to binary classification, PLoS ONE 14 (2019) e0216224.

[9] G. Sergioli, C. Militello, L. Rundo, L. Minafra, F. Torrisi, G. Russo, K. Loon Chow, R. Giuntini, A quantum-inspired classifier for clonogenic assay evaluations, Scientific Reports 11 (2021) 2830.

[10] W. Van Leekwijck, E.E. Kerre, Defuzzification: criteria and classification, Fuzzy Sets and Systems 108, (1999) 159-178.

[11] V. Vargas Caldern, F.A. Gonzlez, H. Vinck Posada, Optimisation-free density estimation and classification with quantum circuits, Quantum Mach. Intell. 4 (2022) 16.

[12] J. Watrous, The Theory of Quantum Information, Cambridge University Press, 2018.

Quantum Fuzzy Inference Engine for Particle Accelerators Control

Giovanni Acampora, *Senior Member, IEEE,* Michele Grossi, Michael Schenk,
Roberto Schiattarella, *Student Member, IEEE*

***Abstract*—Recently, there has been a growing interest in investigating the role of emergent topics, such as quantum computing, in the design of efficient fuzzy inference engines to overcome some issues such as the rule explosion problem. In this scenario, the Quantum Fuzzy Inference Engine (QFIE) could play a pivotal role due to its ability to generate an exponential computational advantage over conventional fuzzy inference engines. However, there are no practical demonstrations that the current generation of quantum computers can reliably run QFIE to efficiently manage complex systems. This paper bridge this gap by using, for the very first time, QFIE to control critical systems such as those related to particle accelerator facilities at the European Organization for Nuclear Research (CERN). As demonstrated by a series of experiments performed at the Advanced Proton Driven Plasma Wakefield Acceleration Experiment (AWAKE), QFIE is able to efficiently control such an environment, paving the way for the use of fuzzy-enabled quantum computers in real-world applications.**

***Index Terms*—Quantum computing, fuzzy control systems, Particle Accelerators.**

I. Introduction

Fuzzy sets and logic theory introduced by Lofti Zadeh [1], [2] has the capability of model classes of objects that do not have precisely defined criteria of membership, in a way to mimic human thinking on computers. Starting from Zadeh's theory of fuzzy logic, Fuzzy Rule-Based Systems (FRBSs) have been developed and they have found a widespread set of applications in the field of automatic control and decision-making [3], [4]. The reason for this success can be explained by the fact that expert knowledge is easily introduced into these systems using *fuzzy rules*. Despite their success, FRBSs suffer from the so-called *fuzzy rule explosion problem*: The number of rules in a FRBS grows exponentially with the number of variables that make up the system. This problem severely limits the ability of FRBS to handle systems with a large number of variables.. Recently, the emergent quantum computing paradigm has been explored to implement a new generation of efficient fuzzy inference engines to potentially overcome the critical limitation of FRBSs. The first quantum fuzzy inference engine, known as QFIE, has been developed in [5]: it uses massive parallelism provided by quantum phenomena, such as superposition and entanglement, to reach an exponential advantage in computing fuzzy rules with respect to the classical counterpart. However, this quantum engine has only been tested on control applications characterized by simple dynamics, such as inverse pendulum, and, as a result, there is no concrete evidence of its operational usability. This paper fills this gap by exploring the potential of QFIE in an important area such as the automatic control of particle accelerators at the European Organization for Nuclear Research (CERN), achieving a twofold result. On the one hand, this research proves that QFIE can be used reliably in complex application scenarios, and on the other hand, it demonstrates that fuzzy logic and FRBS are suitable methods for supporting particle physics experiments at CERN. In fact, QFIE has been shown to reliably support automatic control of particle accelerators that are currently manually tuned due to lack of models or beam instrumentation. The advantages of using QFIE in the automatic control of particle accelerators have been demonstrated in an experimental setting related to the CERN facility known as the Advanced Proton Driven Plasma Wakefield Acceleration Experiment (AWAKE). QFIE was used to implement a complex 10-dimensional control system by using a quantum simulation, due to the limitations that characterize the current generation of quantum computers, known as *Noisy-Intermediate-Scale Quantum (NISQ)* devices. The suitability of QFIE in dealing with the above CERN facility has been assessed in terms of sample efficiency, i.e. the number of actions performed by the controller to achieve the desired behavior of the particle beam. Indeed, optimizing this kind of efficiency is essential in the context of accelerator operation to minimize the impact on the beam time available for physics experiments.

II. QFIE for Particle Accelerators Control

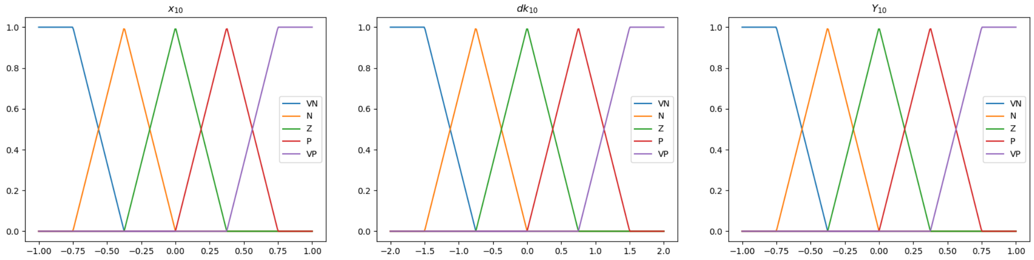
The Advanced Wakefield Experiment (AWAKE) at CERN uses high-intensity 400 GeV proton bunches from the Super Proton Synchrotron (SPS) as a plasma wakefield driver. Electron bunches are simultaneously steered into the plasma cell to be accelerated by the proton-induced wakefields. Electron energies up to 2 GeV have been demonstrated over a plasma cell of 10 m length corresponding to an electric field gradient of 200 MV/m [6]. The ultimate goal for AWAKE is to reach a field gradient of 1 GV/m. These numbers are to be compared to conventional accelerating structures using radio-frequency

G. Acampora and R. Schiattarella are with the Department of Physics "Ettore Pancini'; University of Naples Federico II, Naples, 80126 Italy. E-mail: {giovanni.acampora, roberto.schiattarella }@unina.it.

G. Acampora, R. Schiattarella are with Istituto Nazionale di Fisica Nucleare, Sezione di Napoli, 80126, Italy.

M. Schenk is with the Beams Department at the European Organization for Nuclear Research (CERN), 1211 Meyrin, Switzerland. E-mail: michael.schenk@cern.ch.

M. Grossi is with the IT Department at the European Organization for Nuclear Research (CERN), 1211 Meyrin, Switzerland. E-mail: michele.grossi@cern.ch.



(a) (b) (c)

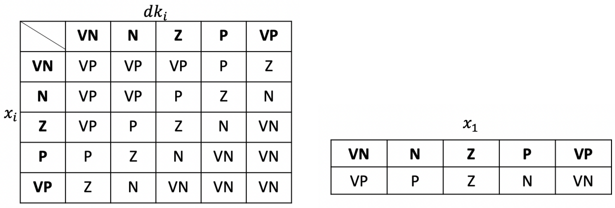
Fig. 1. Fuzzy Partitions for $QFIE_{10}$ Input 1(a), 1(b) and Output 1(c) variables.

(rf) cavities in the X-band regime, which are currently limited to accelerating field gradients of about 150 MV/m [7]. The AWAKE electron source and beam line are particularly interesting for algorithm preparation and testing due to the high repetition rate and insignificant damage potential in case of losing the beam at accelerator components. The AWAKE electrons are generated in a 5 MV photocathode rf gun, accelerated to 18 MeV and then transported through a beam line of 12 m to the AWAKE plasma cell. The trajectory is controlled with 10 horizontal and 10 vertical steering dipoles according to the measurements of 10 BPMs (per plane). The BPM electronic read-out is at 10 Hz and acquisition through the CERN middleware is at 1 Hz. The goal was to develop a QFIE-based controller able to correct the horizontal trajectory with similar accuracy as the response matrix-based singular value decomposition (SVD) algorithm that has been traditionally used [8]. The input state of the controller is formalized as a 10-dimensional vector of horizontal beam position measured with respect to the reference trajectory. Accordingly, the controller action is a 10-dimensional vector of corrector dipole magnet kick angles within a range of $\pm 300\,\mu$rad. To evaluate the performance of the controller, a reward function is used that consists of the negative root-mean-squared (rms) of the measured beam trajectory with respect to the reference at all the BPMs. Developing a single QFIE controlling simultaneously all the corrector dipole magnets along the AWAKE trajectory would reflect in a quantum circuit too big for being classically simulated or executed on a current NISQ device. Therefore to solve the control problem an approach based on the D-NISQ reference model proposed in [9] has been exploited: the original 10-dimensional problem was divided into ten 1-dimensional control problems, where each corrector dipole magnet K_i with $i \in [1, 10]$ is controlled by a QFIE, $QFIE_i$ with $i \in [1, 10]$. Each $QFIE_i \quad \forall i \in [2, 10]$ acts considering two input variables x_i and dk_i, where the former refers to the distance from the ideal position of the beam registered by the corresponding BPM_i, while the latter refers to the sum of the deviation carried out by the magnets that are placed previously to the i-th magnet on the AWAKE beam line. Formally, denoting with $\hat{y}_i$ the corrector dipole magnet kick angles computed by $QFIE_i$, the dk_m input variable for $QFIE_m$ is defined as follows:

$$dk_m = \sum_{i=1}^{m-1} \hat{y}_i. \tag{1}$$

The action of $QFIE_1$ depends just on the position of the particle beam at the first beam position monitor along the trajectory. In detail, dk_i is defined in an interval [-2,2] $\forall i \in [2, 10]$; x_i is defined in an interval [-1,1] $\forall i \in [1, 10]$; the output corrector dipole magnet kick angles y_i are defined in the normalized interval [-1,1] $\forall i \in [1, 10]$.

The fuzzy partitions used for the variables of each $QFIE_i$ are the same. In particular, Fig. 1 shows them for $QFIE_{10}$. Moreover, Fig. 2(a) shows the fuzzy rule base for $QFIE_i \quad \forall i \in [2, 10]$, while Fig. 2(b) shows the fuzzy rule base for $QFIE_1$.



x_i \ dk_i	VN	N	Z	P	VP
VN	VP	VP	VP	P	Z
N	VP	VP	P	Z	N
Z	VP	P	Z	N	VN
P	P	Z	N	VN	VN
VP	Z	N	VN	VN	VN

x_1				
VN	N	Z	P	VP
VP	P	Z	N	VN

(a) (b)

Fig. 2. Rule set for $QFIE_i \; \forall i \in [2, 10]$ (a) and $QFIE_1$ (b). The conjunction of the elements of the first row and column represents the antecedent part of a fuzzy rule having as consequent the corresponding matrix element. For instance, the first rule in $QFIE_i$ corresponds to the sentence *If dk_i is Very Negative and x_i is Very Negative then the correction angle is Very Positive.*

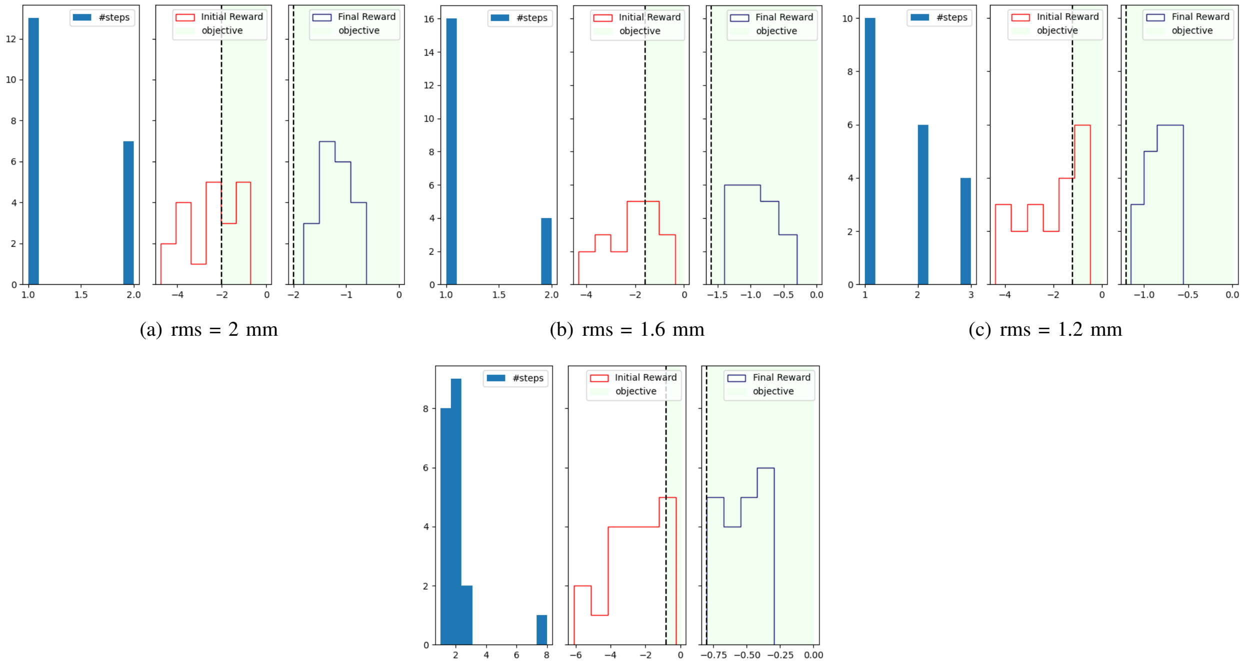
To minimize the number of interactions of the whole controller with the environment a bias factor b has been multiplied by the ten $QFIE$s output. Formally, denoting with $\hat{y}_i$ the output computed by $QFIE_i$, the final corrector dipole magnet kick angle y_i used to modify the environment state is obtained as follows:

$$y_i = \begin{cases} \hat{y}_i \cdot b & \text{if } \hat{y}_i \cdot b \in [-1, 1] \\ 1 & \text{if } \hat{y}_i \cdot b > 1 \\ -1 & \text{if } \hat{y}_i \cdot b < -1 \end{cases} \tag{2}$$

In our experiments, b has been set to 10.

III. Online Tests on Real AWAKE Environment

The QFIE-based FRBS was evaluated on the real AWAKE environment to test sim-to-real transfer. Figure 3 shows the histograms reporting the obtained results: QFIE has been tested by considering four different levels of reward objective threshold value, -2 mm (Fig. 3(a)), -1.6 mm (Fig. 3(b)), -1.2



(a) rms = 2 mm

(b) rms = 1.6 mm

(c) rms = 1.2 mm

(d) rms = 0.8 mm

Fig. 3. Online experiments on the real AWAKE beam line. Each plot refers to a different value of rms threshold.

mm (Fig. 3(c)) and -0.8 mm (Fig. 3(d)). A lower absolute reward value reflects in more precise control of the particle beam. For each threshold value, 20 independent episodes were collected. Histograms in Fig. 3 report respectively the distribution of the number of interactions environment-control required to reach the desired rms value, the distribution of the 20 initial state reward values and the distribution of the 20 final state reward values. As highlighted by the plots, the QFIE-based controller is able to solve the control problem also in the real AWAKE environment. Indeed the objective reward value is reached in 100% of the episodes considered. The number of steps required to achieve such an impressive result is in line with the simulated environment, except for an outlier in the scenario with an rms threshold equal to 0.8 mm.

IV. Conclusion

In this work, a QFIE-based FRBS [5] has been experimentally tested for the very first time to control real-world environments, such as those related to particle physics accelerators at CERN facilities. The main result obtained from this research is twofold: on the one hand, it has been shown that QFIE is able to control these complex environments; on the other hand, it has been proved that FRBSs could be a valid tool for real-time control of particle accelerators for the physics experiments at CERN. In detail, the research was carried out on the AWAKE use case, where the 10-dimensional environment is much more complex and current NISQ devices are not ready to handle the resulting QFIE circuits. In this case, the simulated quantum circuits were tested on real data as an online controller of the beam line. This result proves for the very first time the capability of a FRBS to control a real particle accelerator. In the future QFIE based FRBS will be developed and tested for more complex experiments and environments, where no analytical solutions are available to control the systems. Moreover, further tests on real quantum hardware execution of quantum circuits implementing QFIE will be carried out.

References

[1] L. A. Zadeh, "Fuzzy sets,"information and control", 1965, vol. 8, 338–353," *Fuzzy Algorithms,"Information and Control*, vol. 12, pp. 94–102, 1968.

[2] ——, "Fuzzy sets," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 394–432.

[3] M. Sugeno, "An introductory survey of fuzzy control," *Information sciences*, vol. 36, no. 1-2, pp. 59–83, 1985.

[4] R.-E. Precup and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in industry*, vol. 62, no. 3, pp. 213–226, 2011.

[5] G. Acampora, R. Schiattarella, and A. Vitiello, "On the implementation of fuzzy inference engines on quantum computers," *IEEE Transactions on Fuzzy Systems*, pp. 1–15, 2022.

[6] E. Adli, A. Ahuja, O. Apsimon, R. Apsimon, A.-M. Bachmann, D. Barrientos, F. Batsch, J. Bauche, V. Berglyd Olsen, M. Bernardini *et al.*, "Acceleration of electrons in the plasma wakefield of a proton bunch," *Nature*, vol. 561, no. 7723, pp. 363–367, 2018.

[7] R. Agustsson, P. Carriere, O. Chimalpopoca, V. Dolgashev, M. Gusarova, S. Kutsaev, and A. Y. Smirnov, "Experimental studies of a high-gradient x-band welded hard-copper split accelerating structure," *Journal of Physics D: Applied Physics*, vol. 55, no. 14, p. 145001, 2022.

[8] Y. Chung, G. Decker, and K. Evans, "Closed orbit correction using singular value decomposition of the response matrix," in *Proceedings of International Conference on Particle Accelerators*. IEEE, 1993, pp. 2263–2265.

[9] G. Acampora, F. Di Martino, A. Massa, R. Schiattarella, and A. Vitiello, "D-nisq: a reference model for distributed noisy intermediate-scale quantum computers," *Information Fusion*, vol. 89, pp. 16–28, 2023.

Genetic Algorithms as Classical Optimizer for the Quantum Approximate Optimization Algorithm

Giovanni Acampora, Angela Chiatto, Autilia Vitiello
Department of Physics "Ettore Pancini"
University of Naples Federico II
Naples, Italy
{giovanni.acampora, angela.chiatto, autilia.vitiello}@unina.it

***Abstract*—Optimization is one of the research areas where quantum computing could bring significant benefits. In this scenario, a hybrid quantum-classical variational algorithm, the Quantum Approximate Optimization Algorithm (QAOA), is receiving much attention for its potential to efficiently solve combinatorial optimization problems. This approach works by using a classical optimizer to identify appropriate parameters of a problem-dependent quantum circuit, which ultimately performs the optimization process. Unfortunately, learning the most appropriate QAOA circuit parameters is a complex task that is affected by several issues, such as search landscapes characterized by many local optima. Moreover, gradient-based optimizers, which have been pioneered in this context, tend to waste quantum computing resources. Therefore, gradient-free approaches are emerging as promising methods to address this parameter-setting task. Following this trend, our published work [1] proposes, for the first time, the use of genetic algorithms as gradient-free methods for optimizing the QAOA circuit. The proposed evolutionary approach has been evaluated in solving the MaxCut problem for graphs with 5 to 9 nodes on a noisy quantum device. As the results show, the proposed genetic algorithm statistically outperforms the state-of-the-art gradient-free optimizers by achieving solutions with a better approximation ratio.**

***Index Terms*—Genetic Algorithms, Quantum Approximate Optimization Algorithm, Quantum Computing, Quantum Optimization algorithms.**

I. Introduction

In the field of quantum optimization research, the Quantum Approximate Optimization Algorithm (QAOA) is receiving much attention for its potential to efficiently solve combinatorial optimization problems. Indeed, since its introduction, QAOA has been applied to several different problems such as the tail assignment problem [2] and max k-coloring [3]. In detail, QAOA belongs to the category of variational quantum algorithms since it follows a hybrid classical-quantum scheme. Precisely, a quantum computer is used to run a quantum circuit characterized by free real parameters that computes the solution to the optimization problem, whereas, a classical computer is used to find the optimal values for these parameters. Typically, to achieve this goal, a classical optimizer is used. In general, this uses an iterative approach that starts with a guess of the optimal parameters. Then, in each iteration, the quantum circuit is run with the current values for the parameters to obtain the result of the optimization process and this result is evaluated by a cost function to update the quantum circuit parameters accordingly until a termination criterion, such as the number of iterations, is reached.

Unfortunately, the task of optimizing the parameters of the QAOA quantum circuit could be difficult due to several issues. In particular, recent studies have shown that the search landscape in QAOA is non-convex and contains many local minima [4]. Moreover, as aforementioned, the optimization process of the QAOA quantum circuit exploits the quantum computer which is a resource that should not be wasted. Therefore, a classical optimizer is good when it performs few calls to the quantum device. The first classical optimizers used for QAOA were the gradient-based approaches due to their success in the classical learning approaches. However, gradient-based approaches tend to fall into local minima and make multiple calls to quantum devices to compute the derivatives of the cost function. Hence, the idea of applying gradient-free approaches which consist of optimization methods that do not rely on the derivatives of the cost function are emerging. The first attempts to apply gradient-free optimizers are promising as reported in the work of Fernández-Pendás et al. [5], but finding a good classical optimizer that guarantees a good performance for QAOA is still an open issue.

Starting from these considerations, our published work [1] proposes, for the first time, to apply an evolutionary approach, and in particular, genetic algorithms to the task of optimizing the QAOA parametrized quantum circuit. The proposed genetic algorithm follows a standard structure equipped with elitism. The solution of the optimization problem addressed by the proposed genetic algorithm represents the set of real parameters of the QAOA circuit. The suitability of the proposed approach is shown in an experimental session involving the application of the proposed evolutionary approach to optimize the QAOA quantum circuits devoted to solve the MaxCut problem by considering graph instances with 5 to 9 nodes. The experiments are carried out on a noisy quantum processor in order to show the performance of the proposed approach in a real not ideal scenario. As shown by the results, the proposed genetic algorithm statistically outperforms the state-of-the-art gradient-free approaches by achieving better solutions in terms of a well-known metric in the optimization domain, namely the approximation ratio.

II. A Genetic Algorithm as classical optimizer for QAOA

The goal of our published work [1] is to investigate a Genetic Algorithm (GA) as a classical optimizer to find the set of the optimal circuit parameters in QAOA. Firstly, this proposal is motivated by the drawbacks suffered by the gradient-based approaches. Furthermore, QAOA performance increases when the number of circuit parameters increases. In this scenario where the solution of the optimization task related to QAOA circuit parameters is characterized by high dimensionality, population-based metaheuristics could be more efficient to manage and examine the candidate solutions than the other basic gradient-free optimizers, typically based on approximation or direct search approaches beginning from only a single point. In addition, starting from an initial random population of possible solutions to the problem at hand, GAs search the sub-optimal solution by applying proper stochastic operators, and not deterministic rules.

Briefly, the optimization process of GAs is inspired by the principle of the Darwinian evolution, such as the natural selection: the initial population evolves stochastically toward better solutions thanks to the survival of the fittest solutions over the generations. Each candidate solution to the problem at hand, denoted as *chromosome*, is represented by a string of numbers, denoted as *genes* [6]. To evaluate the quality of each chromosome, it is necessary to define a proper *fitness function*, represented by the cost function in the case of problems to be minimized. This means that the fittest solutions are the chromosomes with the smallest cost values. After generating randomly the initial population of chromosomes, the evolution process takes place in successive iterations, denoted as generations. Each generation involves the application of the so-called *genetic operators*, such as selection, crossover and mutation. Among the different selection mechanisms, we employ the tournament selection due to its proved efficiency [7], [8]. As for the crossover and the mutation operators, the Uniform crossover and the Gaussian mutation are selected, respectively, because they represent an adequate choice for real-coded chromosomes. Typically, during the evolution process, the best chromosome of the current generation is inserted into the next one in order to prevent its possible disappearance. Therefore, the proposed genetic algorithm applies this elitism strategy to maintain the fittest chromosome found throughout the generations. Generations are repeated until a termination criterion is reached.

III. Experiments and results

This section reports the experimental set up used in [1] to evaluate the performance of the proposed GA for solving the MaxCut problem by means of QAOA, in comparison with the traditional gradient-free optimizers. Then, the main results obtained by the compared gradient-free optimizers are discussed in terms of the measure, named quantum approximation ratio, and by performing the Wilcoxon signed rank test [9].

A. Experimental setup

In our work, experiments involve graphs with nodes n from 5 to 9. In particular, we have collected a subset of graph instances described in [10] and accessible at a public repository[1]. The MaxCut problem is solved for each graph with n nodes by means of QAOA. For each problem instance the QAOA circuit is made up of p variational layers, each of which consists of a so-called cost layer depending on the graph structure, and a so-called mixer layer. In order to investigate the performance of the classical optimizer by considering different sizes of the QAOA circuit, in our experimentation, three values for the factor p were used, i.e. $p = 3, 5, 7$. Indeed, increasing the p value leads to increase the number of layers, and as a consequence, the number of gate parameters. The QAOA circuits are implemented using Qiskit™, an open-source framework written in Python programming language and provided by IBM. To run QAOA circuits, the used quantum device is the noisy simulator, namely *Fake Montreal*, made available by Qiskit™ and equipped with 27 qubits.

The gradient-free optimizers taken into account for the comparison with the proposed GA are the ones already applied to optimize QAOA: namely, Constrained Optimization BY Linear Approximation (COBYLA) method, Nelder–Mead method, Modified Powell's method, Simultaneous Perturbation Stochastic Approximation (SPSA) method. The implementation of these methods is offered by Qiskit and Scipy Python modules. The hyper-parameter configurations used for these methods are those set by default by the corresponding Python libraries as already done in the work [5]. As for the configuration of the proposed GA, the values for hyper-parameters have been set to typical ones because of the difficulty of performing a tuning procedure due to the long time of computation. In detail, the configuration of the proposed GA used in our experiments is: $pop\_size = 10$, $k = 3$, $p_c = 0.7$, $p_g = 0.5$, $p_m = 0.25$, $\mu = 0$ and $\sigma = 0.1$. In order to perform a fair comparison and to investigate the advantages of the proposed GA in comparison to the other gradient-free techniques by considering the same effort and employment of the quantum resources, all the involved classical optimizers stop when the maximum number of cost function evaluations, set to 500, is reached. This value corresponds to the number of queries to the quantum device. Indeed, every time the optimizer explores a possible solution, a quantum query to the quantum device is needed to compute the corresponding cost value. Because the optimization process strongly depends on the initialization procedure, 50 independent runs ($n\_runs = 50$) for each classical optimizer and for each combination of the values of p and n are executed. In order to carry out a fair comparison, in each run, the random seed used to generate the initial values of the gate parameters is fixed, so that all the optimizers have been initialized in the same way. Furthermore, in the case of GA, the initial random population contains the chromosome with the same initial values of the gate parameters fed into the

[1] https://code.ornl.gov/qci/qaoa-dataset-version1/-/tree/master/Graphs

other optimizers. At the end of each run, the best cost value reached is stored.

In order to evaluate the performance of the compared classical optimizers, the measure r named quantum approximation ratio is used. In detail, this measure is defined as follows:

$$r = \frac{\langle C \rangle}{C_{max}} \qquad (1)$$

where $\langle C \rangle$ is the optimal expectation value of the cost function reached by the optimizer for a given graph, and C_{max} is the global optimal value for that graph. As a consequence, the higher the value of r, the better is the performance of the classical optimizer. The value 1 represents the optimal result. Moreover, in the paper, we denote the *average quantum approximation ratio* r_a as:

$$r_a = \frac{1}{n\_runs} \sum_{t=1}^{n\_runs} \frac{\langle C_t \rangle}{C_{max}} \qquad (2)$$

where $\langle C_t \rangle$ is the optimal expectation value of the cost function reached by the optimizer for a given graph in the t-th run, C_{max} is the global optimal value for that graph, and n_runs is the number of runs set to 50 in our experimentation as aforementioned.

Moreover, in order to investigate the significance of the impact of the proposed GA on the other classical gradient-free optimizers, a statistical test is conducted by applying the non-parametric statistical method known as Wilcoxon signed rank test.

B. Results and discussion

This section summarizes the obtained results for solving the MaxCut problem for all the graph instances with different number of nodes n at different values of p in terms of quantum approximation ratio. As it is possible to see in [1], GA achieves high values of average quantum approximation ratio values for all the considered test graphs. Indeed, the mean of the average quantum approximation ratio is always higher than 0.8. Moreover, GA outperforms all the other classical optimizers by considering all metrics (i.e., minimum, maximum, mean, and median) related to the average quantum approximation ratio values, except for $n = 8$ and $p = 3$ where SPSA is characterized by a maximum average quantum approximation ratio higher than GA.

In order to give significance to these results, the Wilcoxon's test has been applied between our GA and each one of the compared classical optimizers for each combination of values of p and n. Each sample related to a classical optimizer is composed of 20 elements where each element is related to a graph and represents the *average quantum approximation ratio* for this graph. The statistical significance of the the Wilcoxon's test is expressed by p-values for the pairwise comparisons involving GA and another approach for all combinations of values of p and n. In detail, the one-sided version of the test is considered where the null hypothesis is the equivalence of the two compared algorithms and the alternative one states, instead, that GA is better than the compared approach. The null hypothesis is rejected when the reported p-value is less than the typical significance level $\alpha = 0.05$. Therefore, rejecting the null hypothesis in the pairwise comparisons leads to state that GA statistically outperforms the compared approach at 95% confidence level. The null hypothesis is always rejected except for the pairwise comparison involving GA and SPSA in the case $n = 8$ and $p = 3$. Hence, GA statistically outperforms at 95% confidence level all compared classical optimizers for all combinations of the values n and p, except for the case $n = 8$ and $p = 3$ against SPSA. However, it is important to note that GA statistically outperforms SPSA in all other cases, hence, also when the problem instances for the case $n = 8$ become more complex, i.e., with $p = 5$ and $p = 7$.

IV. Conclusion

Our published work [1] proposes, for the first time, the use of evolutionary algorithms, and in particular genetic algorithms, to optimize QAOA circuit parameters. The benefits provided by the proposed approach have been proved by means of a comparative study with the state-of-the-art gradient-free optimizers (COBYLA, Nelder-Mead, Powell's modified method and SPSA) in solving the MaxCut problem for a specified set of graphs with 5 to 9 nodes at fixed queries of the noisy quantum device. The experiments involved a different number of QAOA parameters to be optimized with increasing values of the parameter p, in order to test the proposed algorithm when QAOA provides reasonable performance in solving optimization problems. As the results of the experimental session show, GA achieves high values of the approximation ratio for all considered test graphs. In addition, GA statistically outperforms state-of-the-art gradient-free optimizers at the 95% confidence level.

References

[1] G. Acampora, A. Chiatto, and A. Vitiello, "Genetic algorithms as classical optimizer for the quantum approximate optimization algorithm," *Applied Soft Computing*, vol. 142, p. 110296, 2023.

[2] P. Vikstål, M. Grönkvist, M. Svensson, M. Andersson, G. Johansson, and G. Ferrini, "Applying the quantum approximate optimization algorithm to the tail-assignment problem," *Physical Review Applied*, vol. 14, no. 3, sep 2020.

[3] E. Bourreau, G. Fleury, and P. Lacomme, "Mixer hamiltonian with qaoa for max k-coloring : numerical evaluations," 2022.

[4] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1–8.

[5] M. Fernández-Pendás, E. F. Combarro, S. Vallecorsa, J. Ranilla, and I. F. Rúa, "A study of the performance of classical minimizers in the quantum approximate optimization algorithm," *Journal of Computational and Applied Mathematics*, vol. 404, p. 113388, 2022.

[6] D. E. Goldberg, "Genetic algorithms in search," *Optimization, and MachineLearning*, 1989.

[7] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of genetic algorithms*. Elsevier, 1991, vol. 1, pp. 69–93.

[8] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," *Evolutionary Computation*, vol. 4, no. 4, pp. 361–394, 1996.

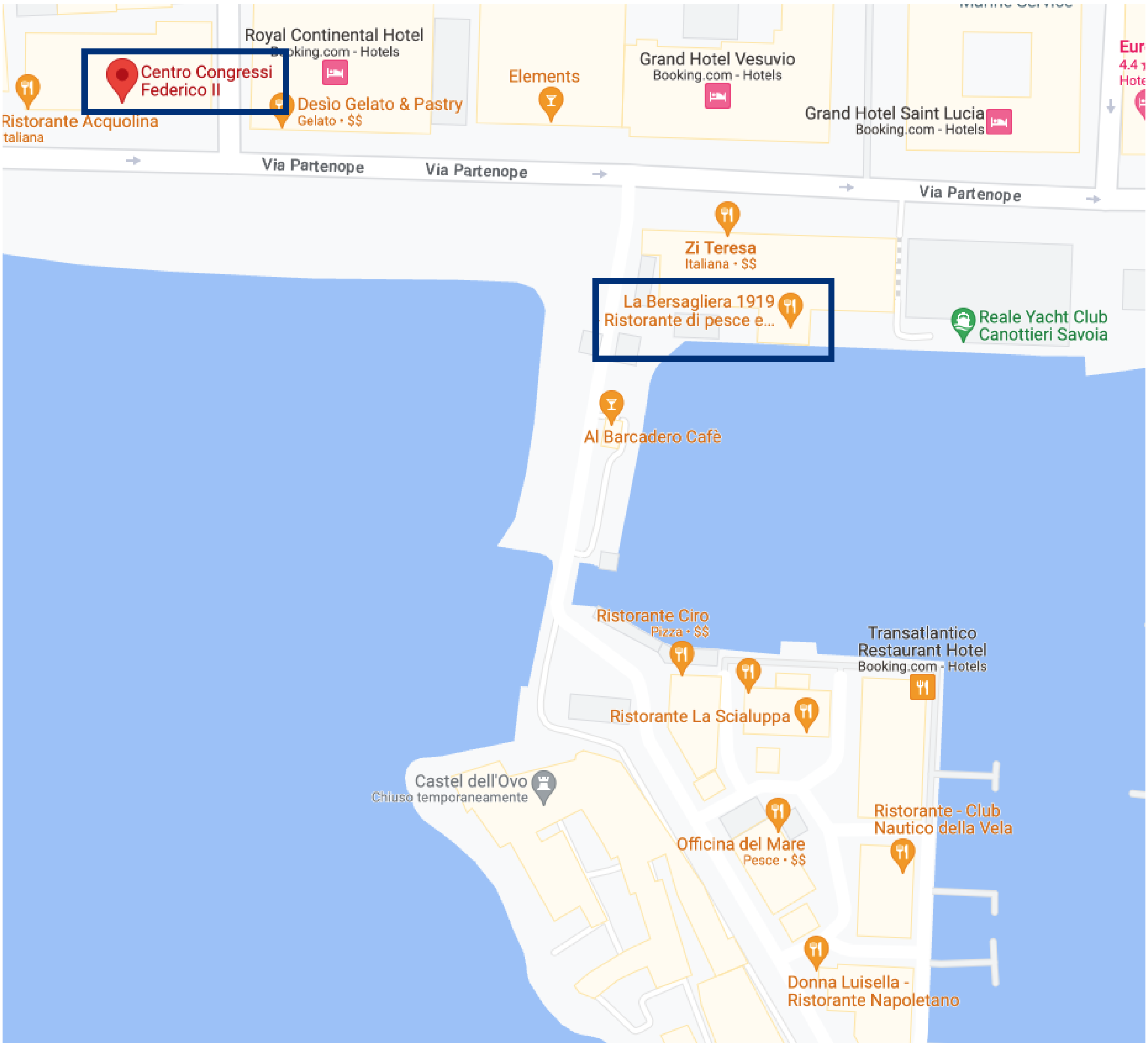
[9] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[10] P. C. Lotshaw, T. S. Humble, R. Herrman, J. Ostrowski, and G. Siopsis, "Empirical performance bounds for quantum approximate optimization," *Quantum Information Processing*, vol. 20, no. 12, pp. 1–32, 2021.

Registration: Hall, Centro Congressi Federido II, via Partenope 36, 80121 Naples, Italy

Talks: Aula Magna, Centro Congressi Federido II, via Partenope 36, 80121 Naples, Italy

Social Dinner: La Bersagliera Restaurant, Borgo Marinari, 10/11, 80121 Naples, Italy


Centro Congressi Federico II
Royal Continental Hotel
Booking.com - Hotels
Ristorante Acquolina
Italiana
Desìo Gelato & Pastry
Gelato · $$
Elements
Grand Hotel Vesuvio
Booking.com - Hotels
Grand Hotel Saint Lucia
Booking.com - Hotels
Via Partenope
Via Partenope
Via Partenope
Zi Teresa
Italiana · $$
La Bersagliera 1919
Ristorante di pesce e...
Reale Yacht Club Canottieri Savoia
Al Barcadero Cafè
Ristorante Ciro
Pizza · $$
Transatlantico Restaurant Hotel
Booking.com - Hotels
Ristorante La Scialuppa
Castel dell'Ovo
Chiuso temporaneamente
Ristorante - Club Nautico della Vela
Officina del Mare
Pesce · $$
Donna Luisella - Ristorante Napoletano