# Parallel circuit implementation of variational quantum algorithms - Supplementary Files Information

## Multi-objective QAOA

Though section 4 is mainly focused on analyzing the optimization problems to find slice Hamiltonians by looking at the constraints, we want to stress that this identification can be done by looking at other structures in the optimization problem. Indeed, by exploiting the method presented in [29] we can choose to include in $H_{\tilde{C}}$ also the objective function of the problem inspected. This can lead to a further decomposition of the problem. For instance, if we consider the VRP, we can see that if we consider $H_{\text{of}} + H_{\tilde{C}}$ to be the term to only simulate classically, we have that the slices are now the Hamiltonians

$$H_a = \sum_{s=0}^{n} \left(1 - \sum_{i=0}^{n} x_{a,i,s}\right)^2 .$$

Therefore, the quantum circuits only implement the one-hot constraint, which ensures that a vehicle can only be in one location at a given time step.

Nevertheless, considering the objective function as part of $H_{\tilde{C}}$ allows us to implement multi-objective constraint optimization problems. Let us consider the following mathematical program in the standard form:

$$\begin{aligned} \min_{x \in \mathbb{B}^n} \quad & (f_i(x))_{i=1,\dots,m} \\ \text{s.t.} \quad & \sum_{j=1}^{n} c_{ij} x_j = b_i \quad \forall i \in [k]. \end{aligned}$$

Let us now formalize the Hamiltonian that only implements the constraints. This is the slice Hamiltonian in this example and it is the only one that we identify. The Hamiltonian can be read as:

$$H = \sum_{i=0}^{k} \left(\sum_{j=1}^{n} c_{ij} x_j - b_i\right)^2 . \tag{17}$$

We can notice that, by following the same procedure as in fig. 2, we can implement a QAOA circuit that represents the Hamiltonian $H$ and we optimize the circuit over the function:

$$(f_i(x) + H)_{i=1,\dots,m} .$$

Since now we are evaluating a multi-objective function, we have to evaluate and identify the parameter based on the Pareto front. After the selection of the "optimal" point, the parameters are evaluated and the quantum circuit is updated. This allows the implementation of multi-objective combinatorial optimization problems without using a metric to reduce the multi-objective function to a QUBO problem.

**Slice Hamiltonian for the MaxCut problem.**

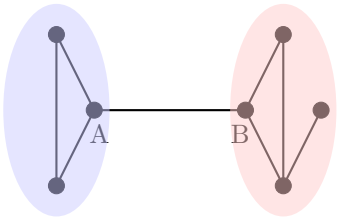



**Fig. 10** We consider the MaxCut problem over the considered graph. In the picture, we highlight the two slices that we build in this example. The label $A$ and $B$ are use to identify the edge representation in the Hamiltonian $H$ of the term $H_{\tilde{C}}$.

## Different slicing methods

As mentioned in section 4 the slicing method can be applied by considering any structure informed by the optimization problem, not only constraints or the objective function. Let us show this by considering an example. We consider the MaxCut problem on the graph shown in fig. 10. The Hamiltonian of the MaxCut problem [11] is define as

$$H = -\sum_{(i,j)\in E} \sigma_i^z \sigma_j^z,$$

where $E$ is the set of edges of the graph considered and $\sigma_i^z$ is the spin variables that take value $+1$ if the node $i$ is considered in one partition of the MaxCut and $-1$ otherwise.

In fig. 10 we can identify two different regions of the graph that are connected only by one edge. In this case, we can read the Hamiltonian in the following fashion:

$$H = -\sigma_A^z \sigma_B^z + \left( -\sum_{(i,j)\in E_b} \sigma_i^z \sigma_j^z - \sum_{(i,j)\in E_r} \sigma_i^z \sigma_j^z \right),$$

where $E_b$ is the set of edges identified by the blue region and $E_r$ is the set of edges identified by the red region. By using the same notation as section 2.1, we can easily see that $H_{\tilde{C}} = -\sigma_A^z \sigma_B^z$ and the two addends inside the brackets are the two slice Hamiltonians. In this case, we can apply the slicing method because of a topological property of the graph. We can observe that the graph is 1-connected, i.e. the graph becomes disconnected by leaving one edge out. As we already noticed, one can identify the edge $(A, B)$ as the edge that connects the two connected components defined by the edge sets $E_b$ and $E_r$.

It is important to note that this results in the same circuit cutting method for QAOA presented in [58]. The only difference is in the choice of the gates to leave out. In our framework, the choice is informed by the problem, while in [58] the optimal cut must be found introducing a problem that reduces to the graph partitioning problem, which is NP-hard [59].

The MaxCut problem can be considered a special example because the quadratic terms defined in the Hamiltonian, as well as the two-qubit gates of the QAOA ansatz,

has a one-to-one connection with the edges of the problem graph. Therefore the connected subgraphs correspond exactly to the slices of the circuit as well. However, in general, this example shows that we can exploit different properties of the graph defined by the optimization problem (e.g. the $k$-connectivity, as in this example) in order to apply the slicing method.

### Entanglement entropy analysis between slices

To apply the framework proposed in section 2.1, we must remove from the circuit 2-qubits gates. Thus, we remove quantum information from the circuit since the slices are classically separable and therefore there are no quantum correlations between them. We can analyze the amount of quantum correlation lost by computing the entanglement entropy between the slices. The entanglement entropy can be computed using the density matrix representation of the VQA parameterized quantum circuit. In general, the density matrix can be written as

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|,$$

and it describes a quantum system that can be in the state $|\psi_i\rangle$ with probability $p_i$. The entropy of the quantum system is described by $\rho$ using the Von Neumann entropy

$$S(\rho) = -\sum_j \eta_j \ln \eta_j,$$

where $\{\eta_j\}_j$ are the eigenvalues associated to the matrix $\rho$. Given a quantum system in the state $|\psi\rangle$ and a partition of its components in the subsystems $A$ and $B$, we can compute how much the subsystems $A$ and $B$ are entangled using the Von Neumann entropy. This can be obtained by computing the density matrix of the subsystem $A$ (or $B$ interchangeably) from the density matrix of the system $\rho = |\psi\rangle\langle\psi|$— the density matrix of the subsystem is called reduced density matrix. The reduced density matrix is

$$\rho_A = \mathrm{tr}_A(\rho) \coloneqq \sum_{\left\{|\phi_i^A\rangle\right\}} \left(\langle\phi_i^A| \otimes \mathbb{I}^B\right) \rho \left(|\phi_i^A\rangle \otimes \mathbb{I}^B\right),$$

where $\left\{|\phi_i^A\rangle\right\}$ is an orthonormal basis of the Hilbert space of the subsystem $A$. The entanglement entropy between the two subsystems is $S(\rho_A)$.

Therefore to compute the entanglement entropy between two slices we have to consider the density matrix of a parameterized quantum circuit. This can be obtained starting from the wavefunction representing the outcome of the circuit:

$$|\psi(\boldsymbol{\alpha})\rangle = U(\boldsymbol{\alpha}) |0\rangle^{\otimes_n},$$

where $U(\boldsymbol{\alpha})$ is the unitary matrix representing the parameterized quantum circuit of the VQA evaluated in the hyperparameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_q)$, $|0\rangle^{\otimes_n}$ is the initial state of the parameterized quantum circuit, and $n$ is the number of qubits involved in the circuit. The density operator of the state $|\psi(\boldsymbol{\alpha})\rangle$ is $\rho(\boldsymbol{\alpha}) = |\psi(\boldsymbol{\alpha})\rangle\langle\psi(\boldsymbol{\alpha})|$. Therefore, to

compute the entanglement entropy between the slices we have to compute the reduced density matrix over the subsystem of the slice from $\rho(\boldsymbol{\alpha})$. Thus, we evaluate $\rho(\boldsymbol{\alpha})$ with the hyperparameters trained with a parallel implementation of the VQA, compute the reduced density matrix of the slice, and evaluate its entanglement entropy.

We consider the entanglement entropy obtained by evaluating the QAOA circuit density matrix representation with the hyperparameters trained using pQAOA, single-slice pQAOA. We evaluate the different QAOA implementations with $p = 1$ and $p = 2$ using the hyperparameters trained on the 50 VRP instances solved in section 2.3. Then, to study how well our parallel implementation can approximate QAOA, we consider the entropy distribution amongst the instances and compare the difference between the entropy obtained from the hyperparameters trained with QAOA and its parallel implementations by considering each instance. Moreover, to understand whether low entropy between slices can suggest when it is suitable to apply the slicing, we analyze the relationship between entanglement entropy obtained by evaluating the hyperparameters trained with QAOA and the minimum energy obtained with its parallel implementation. The results are shown in fig. 11.

As we can see in fig. 11a, for $p = 1$ the distribution of the entropy between the slices appears similar among the different implementations. Furthermore, we can observe that in most instances, the entropy is large for both the implementation with hyperparameters trained with QAOA and its parallel implementations. Thus, we obtain a good approximation of QAOA by using our parallelization framework. However, we can see that we cannot identify a general behavior for all the cases considered and that the entropy of many considered instances is problem-dependent.

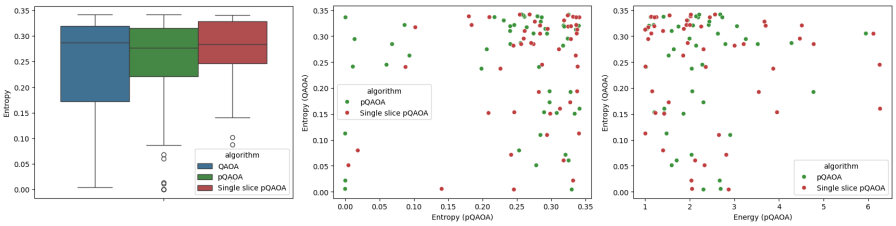
On the other hand, the results shown in fig. 11b present a different situation where the parallel implementations do not approximate QAOA with $p = 2$. In this case, not only we cannot characterize the entropy distribution, but we can observe that the hyperparameters optimized with QAOA follow a different trend than those trained with a parallel implementation of the VQA. We believe this is due to the number of gates used to implement the slices. As mentioned above, when the number of layers increases the number of removed gates from the circuit increases. Therefore, the approximation of the VQA circuit is less accurate. Nevertheless, as noted in section 2.3, even though the slices differ from the original circuit, we can still obtain better quality solutions by employing our parallelization framework.

Moreover, we can see that the entanglement entropy between the slices obtained by evaluating the hyperparameters trained with QAOA cannot be used to decide whether to apply the slicing. Both for $p = 1$ and $p = 2$ we cannot identify any trend between the low or high entanglement between the slices and the performance of pQAOA.

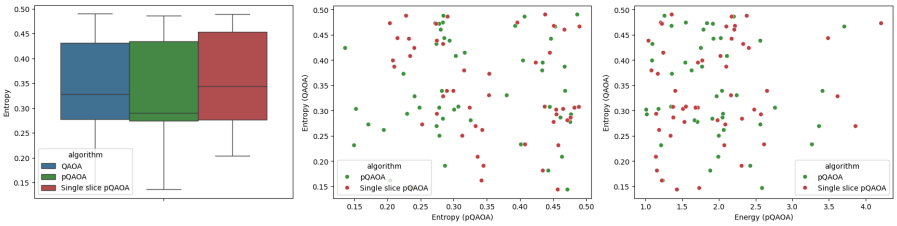
### Relation with the number of slices

The parallelization framework explained in section 2.1 can be seen as a method to approximate a VQA by moving the computational effort from the quantum circuit to the classical optimizer of the algorithm. This approximation depends on the number of slices used to apply the parallelization. Notice that there are two trivial cases: the framework is applied by using only one slice and, therefore the parallel implementation

**Analysis of the entropy between the two slices derived from the VRP Hamiltonian.**




(a) p=1




(b) p=2

**Fig. 11** We obtain the entropy between two slices by evaluating the QAOA quantum circuits as we shown in fig. 7, and by computing the bipartite entropy, considering the subsystems involved by the slices. We show the results for $p = 1$ and $p = 2$. We analyse the distribution of the entropy across the 50 instances solved (*left*) where the boxes represent the upper quartile (top line), median (central line) and lower quartile (bottom line), and the whiskers extend to 1.5 times the interquartile range; the relationship between the entanglement entropy between the slices by evaluating the hyperparameters trained by QAOA and the hyperparameters trained with pQAOA (*center*); and, the comparison between the entanglement entropy between the slice of the QAOA circuits evaluated with hyperparameters trained with QAOA and the energy of the results obtained with pQAOA (*right*). We can use these comparisons to understand whether the parallelization can approximate the behaviour of the entanglement entropy coming from the 2-qubits gates we removed to create the slices and if there is a correlation between the performance of pQAOA and the low entanglement entropy between the slices in QAOA. For $p = 1$, the pQAOA can approximate the entropy distribution. Conversely, for $p = 2$ the situation is different and pQAOA cannot approximate the behavior of QAOA as for $p = 1$. We believe this is due to the number of gates removed from the ansatz. If for $p = 1$ the circuit is similar except for the doors removed, by adding one layer the structure of the parameterized quantum circuit becomes significantly different. Therefore, the approximation of the entanglement cannot be as good as before. Nevertheless, this does not necessarily influence the performance of the algorithm as we showed in fig. 6.

is the direct implementation of the VQA circuit itself; and when we remove all the 2-qubit gates from the circuit obtaining slices consisting of only one qubit. We identify the latter as trivial because a classical computer can efficiently simulate a circuit with only one qubit. Therefore, when we apply the slicing by creating a slice for each qubit we obtain a classical approximation of the VQA circuit, equivalent to removing all quantum gates from the circuit.

Therefore, different choices can be made to apply the slicing, and different performances can be achieved. To study how the choice of the number of slices influences the performance of the VQA we consider different parallel implementations of QAOA

**Example of many ways of slicing directly a quantum circuit.**

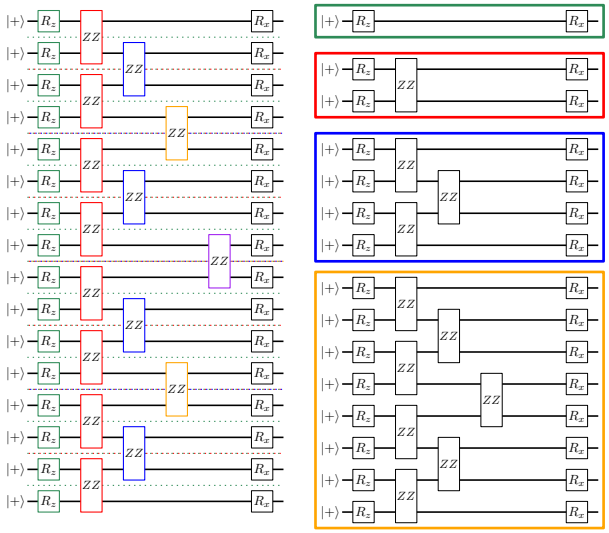



**Fig. 12** QAOA circuit implementation of the Hamiltonian eq. (18). On the right side, the slices obtained by removing the Hamiltonians eq. (20) (orange), eq. (21) (blue), eq. (22) (red), and eq. (23) (green) are shown. quantum gates representing the Hamiltonian eq. (18) represent the color that identifies the slices if the circuit is considered up to that gate. The dashed lines represent the action of the slicing on the circuit.

used to solve the following Hamiltonian:

$$H_{\text{toy}} = \sum_{i=0}^{15} h_i \sigma_i^z + \sum_{i=0}^{14} J_{i,i+1} \sigma_i^z \sigma_{i+1}^z, \tag{18}$$

where $\sigma_i^z$ is a spin variable that can take values $-1$ and $+1$, and $h_i, J_{i,i+1} \in [-1, +1]$ are randomly generated. Notice that this Hamiltonian is an Ising chain, i.e. the graph representing the Hamiltonian is a path. Thus, we consider each 2-body interaction in $H_{\text{toy}}$ to create slices of the problem. Due to the limited number of qubits, we can compute the ground state with a brute-force solver.

We show how to apply the slicing by using the same notation used in section 2.2.1, where $H_{\tilde{C}}$ is part of the Hamiltonian which is not implemented in the circuit but only optimized classically. We consider five different cases:

$$H_{\tilde{C}}^0 = 0, \tag{19}$$

$$H_{\tilde{C}}^1 = \sigma_7^z \sigma_8^z, \tag{20}$$

$$H_{\tilde{C}}^2 = \sum_{\substack{i=0 \\ 4|i+1}}^{14} \sigma_i^z \sigma_{i+1}^z, \tag{21}$$

**Comparison of pQAOA performance with different number of slices.**

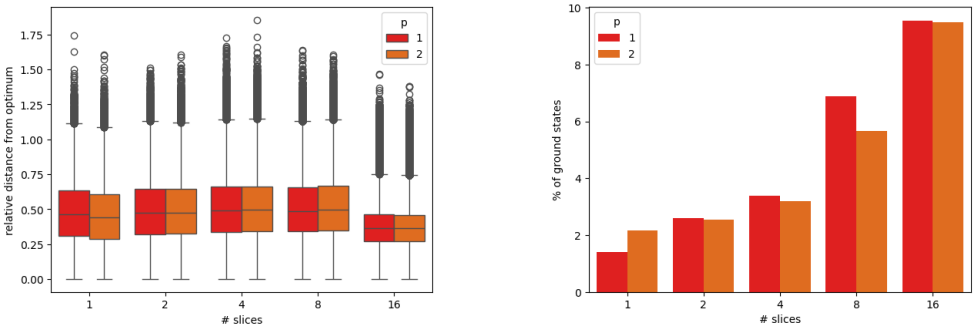



**Fig. 13** Performance of different implementations of single-slice pQAOA with different numbers of slices. The algorithm performance is related to the number of slices, where QAOA has only 1 slice. Notice that pQAOA with as many slices as qubits is a classical algorithm, since 1 qubit can be efficiently simulated, (in this case we have 16 slices). We refer to pQAOA when the slices consist of only 1 qubits as classical pQAOA. We consider the results in terms of: the relative distance from the optimum, computed with a brute force method (*left*) where the boxes represent the upper quartile (top line), median (central line) and lower quartile (bottom line), and the whiskers extend to 1.5 times the interquartile range. The circles stand for the outliers; and, the success rate, measured by the percentage of samples that are the ground state over the set of global samples obtained (*right*). We can note that when there is entanglement in the circuit (slice from 1 to 8) the results are comparable, while for the classical pQAOA the results are the best obtained over 5 different randomly generated instances of eq. (18). Furthermore, we note that the more effort is moved to the classical optimizer the more ground states are observed. Nevertheless, this analysis is based on a simple problem and therefore no general conclusion can be drawn.

$$H_C^3 = \sum_{\substack{i=0 \\ 2\nmid i}}^{14} \sigma_i^z \sigma_{i+1}^z, \tag{22}$$

$$H_C^4 = \sum_{i=0}^{14} \sigma_i^z \sigma_{i+1}^z, \tag{23}$$

where "$x \mid y$" ("$x \nmid y$") means that $x$ divides (does not divide) $y$. Notice that eq. (19) and eq. (23) are the trivial cases described above. The number of slices in the different implementations is: 1 for eq. (19), 2 for eq. (20), 4 for eq. (21), 8 for eq. (22), and, 16 for eq. (23)— we call this last implementation classical pQAOA. The graphical representation of the slices is given in fig. 12. We can see that by applying the slicing in the presented way we create identical slices.

To study the performance, we simulate QAOA with $p = 1, 2$ with the parallel implementations described in fig. 12. Excluding the direct implementation of QAOA, we implement single-slice pQAOA by using different slices obtained with eq. (20), eq. (21), eq. (22), eq. (23). We optimize the hyperparameter and use the solutions collected from the optimized pQAOA. In this experiment, we train the hyperparameters by collecting $100,000$ samples at each iteration to obtain the best performance possible. The results are shown in fig. 13 regarding the relative distance from the optimum and the percentage of ground states obtained. We can see that every non-trivial

pQAOA execution obtains a similar distribution of solution quality to QAOA, while we obtain better solution quality for classical pQAOA. Furthermore, we can note that the relative distance from the optimum increases by increasing the number of slices. Therefore, in this case, the more 2-qubits gates are removed from the quantum circuits the more the performance improves. Nevertheless, we believe this behavior could be related to the simple geometry of the problem. Since the problem is a chain of qubits, the choice of the value of the qubit depends only on its local interactions and does not depend on the other qubit values. Therefore, we expect these problems with local properties to be easy to solve with a black box optimizer.

Thus, more in-depth analysis must be done to give a general conclusion about the relation between the number of slices and the performance of the parallelization framework. The analysis of this relationship will be taken into account in future works.

## References


[1] Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by a quantum walk. In: Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, pp. 59–68 (2003)

[2] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)

[3] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)

[4] Simon, D.R.: On the power of quantum computation. SIAM journal on computing **26**(5), 1474–1483 (1997)

[5] Preskill, J.: Quantum computing in the nisq era and beyond. Quantum **2**, 79 (2018)

[6] Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., *et al.*: Variational quantum algorithms. Nature Reviews Physics **3**(9), 625–644 (2021)

[7] Yarkoni, S., Raponi, E., Bäck, T., Schmitt, S.: Quantum annealing for industry applications: introduction and review. Rep. Prog. Phys. **85**(10), 104001 (2022)

[8] Zhou, L., Wang, S.-T., Choi, S., Pichler, H., Lukin, M.D.: Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Physical Review X **10**(2), 021067 (2020)

[9] Lucas, A.: Ising formulations of many NP problems. Frontiers in Physics **2** (2014) https://doi.org/10.3389/fphy.2014.00005

[10] Glover, F., Kochenberger, G., Hennig, R., Du, Y.: Quantum bridge analytics i: a tutorial on formulating and using qubo models. Annals of Operations Research **314**(1), 141–183 (2022)

[11] Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)

[12] Streif, M., Leib, M.: Forbidden subspaces for level-1 quantum approximate optimization algorithm and instantaneous quantum polynomial circuits. Physical Review A **102**(4), 042416 (2020)

[13] Herrman, R., Lotshaw, P.C., Ostrowski, J., Humble, T.S., Siopsis, G.: Multi-angle quantum approximate optimization algorithm. Scientific Reports **12**(1), 6781 (2022)

[14] Skolik, A., Cattelan, M., Yarkoni, S., Bäck, T., Dunjko, V.: Equivariant quantum circuits for learning on weighted graphs. npj Quantum Information **9**(1), 47 (2023)

[15] Streif, M., Yarkoni, S., Skolik, A., Neukart, F., Leib, M.: Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm. Physical Review A **104**(1), 012403 (2021)

[16] Fitzek, D., Ghandriz, T., Laine, L., Granath, M., Kockum, A.F.: Applying quantum approximate optimization to the heterogeneous vehicle routing problem. arXiv preprint arXiv:2110.06799 (2021)

[17] Bravyi, S., Kliesch, A., Koenig, R., Tang, E.: Obstacles to variational quantum optimization from symmetry protection. Physical review letters **125**(26), 260505 (2020)

[18] Bennett, T., Matwiejew, E., Marsh, S., Wang, J.B.: Quantum walk-based vehicle routing optimisation. Frontiers in Physics **9**, 730856 (2021)

[19] Streif, M., Leib, M.: Training the quantum approximate optimization algorithm without access to a quantum processing unit. Quantum Science and Technology **5**(3), 034008 (2020)

[20] Egger, D.J., Mareček, J., Woerner, S.: Warm-starting quantum optimization. Quantum **5**, 479 (2021)

[21] Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G.H., *et al.*: The variational quantum eigensolver: a review of methods and best practices. Physics Reports **986**, 1–128 (2022)

[22] Barahona, F.: On the computational complexity of ising spin glass models. Journal of Physics A: Mathematical and General **15**(10), 3241 (1982)

[23] Fontana, E., Fitzpatrick, N., Ramo, D.M.n., Duncan, R., Rungger, I.: Evaluating the noise resilience of variational quantum algorithms. Phys. Rev. A **104**, 022403 (2021) https://doi.org/10.1103/PhysRevA.104.022403

[24] Bittel, L., Kliesch, M.: Training variational quantum algorithms is np-hard. Physical Review Letters **127**(12), 120502 (2021)

[25] Blekos, K., Brand, D., Ceschini, A., Chou, C.-H., Li, R.-H., Pandya, K., Summer, A.: A review on quantum approximate optimization algorithm and its variants. Physics Reports **1068**, 1–66 (2024)

[26] Zhu, L., Tang, H.L., Barron, G.S., Calderon-Vargas, F., Mayhall, N.J., Barnes, E., Economou, S.E.: Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. Physical Review Research

**4**(3), 033029 (2022)

[27] Bärtschi, A., Eidenbenz, S.: Grover mixers for qaoa: Shifting complexity from mixer design to state preparation. In: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 72–82 (2020). IEEE

[28] Bravyi, S., Kliesch, A., Koenig, R., Tang, E.: Hybrid quantum-classical algorithms for approximate graph coloring. Quantum **6**, 678 (2022)

[29] Hao, T., Shaydulin, R., Pistoia, M., Larson, J.: Exploiting in-constraint energy in constrained variational quantum optimization. arXiv preprint arXiv:2211.07016 (2022)

[30] Wang, Z.-D., Zheng, P.-L., Wu, B., Zhang, Y.: Quantum dropout for efficient quantum approximate optimization algorithm on combinatorial optimization problems. arXiv preprint arXiv:2203.10101 (2022)

[31] Kirby, W.M., Tranter, A., Love, P.J.: Contextual subspace variational quantum eigensolver. Quantum **5**, 456 (2021) https://doi.org/10.22331/q-2021-05-14-456

[32] Tüysüz, C., Clemente, G., Crippa, A., Hartung, T., Kühn, S., Jansen, K.: Classical splitting of parametrized quantum circuits. arXiv preprint arXiv:2206.09641 (2022)

[33] Albash, T., Lidar, D.A.: Adiabatic quantum computation. Reviews of Modern Physics **90**(1), 015002 (2018)

[34] Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Phys. Rev. E **58**, 5355–5363 (1998) https://doi.org/10.1103/PhysRevE.58.5355

[35] Johnson, M.W., Amin, M.H.S., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P., Chapple, E.M., Enderud, C., Hilton, J.P., Karimi, K., Ladizinsky, E., Ladizinsky, N., Oh, T., Perminov, I., Rich, C., Thom, M.C., Tolkacheva, E., Truncik, C.J.S., Uchaikin, S., Wang, J., Wilson, B., Rose, G.: Quantum annealing with manufactured spins. Nature **473**(7346), 194–198 (2011) https://doi.org/10.1038/nature10012

[36] Hauke, P., Katzgraber, H.G., Lechner, W., Nishimori, H., Oliver, W.D.: Perspectives of quantum annealing: methods and implementations. Rep. Prog. Phys. **83**(5), 054401 (2020)

[37] Boothby, K., Bunyk, P., Raymond, J., Roy, A.: Next-generation topology of d-wave quantum processors. arXiv preprint arXiv:2003.00133 (2020)

[38] Amin, M.H.: Searching for quantum speedup in quasistatic quantum annealers. Phys. Rev. A **92**, 052323 (2015) https://doi.org/10.1103/PhysRevA.92.052323

[39] Marshall, J., Rieffel, E.G., Hen, I.: Thermalization, freeze-out, and noise: Deciphering experimental quantum annealers. Phys. Rev. Appl. **8**, 064025 (2017) https://doi.org/10.1103/PhysRevApplied.8.064025

[40] Lanting, T., King, A.D., Evert, B., Hoskinson, E.: Experimental demonstration of perturbative anticrossing mitigation using nonuniform driver hamiltonians. Phys. Rev. A **96**, 042322 (2017) https://doi.org/10.1103/PhysRevA.96.042322

[41] Andriyash, E., Bian, Z., Chudak, F., Drew-Brook, M., D. King, A., G. Macready, W., Roy, A.: Boosting integer factoring performance via quantum annealing offsets. https://www.dwavesys.com/resources/publications

[42] Yarkoni, S., Wang, H., Plaat, A., Bäck, T.: Boosting quantum annealing performance using evolution strategies for annealing offsets tuning. In: Quantum Technology and Optimization Problems, pp. 157–168. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-14082-3_14

[43] Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature communications **5**(1), 4213 (2014)

[44] Liu, X., Angone, A., Shaydulin, R., Safro, I., Alexeev, Y., Cincio, L.: Layer vqe: A variational approach for combinatorial optimization on noisy quantum computers. IEEE Transactions on Quantum Engineering **3**, 1–20 (2022) https://doi.org/10.1109/TQE.2021.3140190

[45] Amaro, D., Modica, C., Rosenkranz, M., Fiorentini, M., Benedetti, M., Lubasch, M.: Filtering variational quantum algorithms for combinatorial optimization. Quantum Science and Technology **7**(1), 015021 (2022) https://doi.org/10.1088/2058-9565/ac3e54

[46] Saib, W., Wallden, P., Akhalwaya, I.: The effect of noise on the performance of variational algorithms for quantum chemistry. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 42–53 (2021). https://doi.org/10.1109/QCE52317.2021.00020

[47] Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J.M., Gambetta, J.M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature **549**(7671), 242–246 (2017) https://doi.org/10.1038/nature23879

[48] IBM Quantum: IBM Quantum technology roadmap

[49] Inc., D.-W.: D-Wave System Documentation. https://docs.dwavesys.com/docs/latest/. Accessed: 2023-11-23 (2009)

[50] Akshay, V., Rabinovich, D., Campos, E., Biamonte, J.: Parameter concentrations

in quantum approximate optimization. Physical Review A **104**(1), 010401 (2021)

[51] IBM Quantum: IBM transpiler overview

[52] Amin, M.H.: Searching for quantum speedup in quasistatic quantum annealers. Physical Review A **92**(5), 052323 (2015)

[53] Feld, S., Roch, C., Gabor, T., Seidel, C., Neukart, F., Galter, I., Mauerer, W., Linnhoff-Popien, C.: A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. Frontiers in ICT **6**, 13 (2019)

[54] Marsh, S., Wang, J.: A quantum walk-assisted approximate algorithm for bounded np optimisation problems. Quantum Information Processing **18**, 1–18 (2019)

[55] Cattelan, M., Yarkoni, S.: Modeling routing problems in qubo with application to ride-hailing. arXiv preprint arXiv:2212.04894 (2022)

[56] NVidia Corporation: NVIDIA cuQuantum Appliance. https://docs.nvidia.com/cuda/cuquantum/appliance/index.html

[57] Google Inc.: OR-Tools. https://developers.google.com/optimization

[58] Marshall, S.C., Gyurik, C., Dunjko, V.: High dimensional quantum learning with small quantum computers. arXiv preprint arXiv:2203.13739 (2022)

[59] Andreev, K., Räcke, H.: Balanced graph partitioning. In: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures. SPAA '04, pp. 120–124. Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1007912.1007931 . https://doi.org/10.1145/1007912.1007931