Transfer learning of optimal QAOA parameters in combinatorial optimization

J. A. Montañez-Barrera,[1, *] Dennis Willsch,[1, 2] and Kristel Michielsen[1, 3, 4]

[1] *Jülich Supercomputing Centre, Institute for Advanced Simulation, Forschungszentrum Jülich, 52425 Jülich, Germany*
[2] *Faculty of Medical Engineering and Technomathematics, University of Applied Sciences Aachen, 52428 Jülich, Germany*
[3] *AIDAS, 52425 Jülich, Germany*
[4] *RWTH Aachen University, 52056 Aachen, Germany*

Solving combinatorial optimization problems (COPs) is a promising application of quantum computation, with the Quantum Approximate Optimization Algorithm (QAOA) being one of the most studied quantum algorithms for solving them. However, multiple factors make the parameter search of the QAOA a hard optimization problem. In this work, we study transfer learning (TL), a methodology to reuse pre-trained QAOA parameters of one problem instance into different COP instances. This methodology can be used to alleviate the necessity of classical optimization to find good parameters for individual problems. To this end, we select small cases of the traveling salesman problem (TSP), the bin packing problem (BPP), the knapsack problem (KP), the weighted maximum cut (MaxCut) problem, the maximal independent set (MIS) problem, and portfolio optimization (PO), and find optimal β and γ parameters for p layers. We compare how well the parameters found for one problem adapt to the others. Among the different problems, BPP is the one that produces the best transferable parameters, maintaining the probability of finding the optimal solution above a quadratic speedup over random guessing for problem sizes up to 42 qubits and $p = 10$ layers. Using the BPP parameters, we perform experiments on IonQ Harmony and Aria, Rigetti Aspen-M-3, and IBM Brisbane of MIS instances for up to 18 qubits. The results indicate that IonQ Aria yields the best overlap with the ideal probability distribution. Additionally, we show that cross-platform TL is possible using the D-Wave Advantage quantum annealer with the parameters found for BPP. We show an improvement in performance compared to the default protocols for MIS with up to 170 qubits. Our results suggest that there are QAOA parameters that generalize well for different COPs and annealing protocols.

Keywords: QUBO; transfer learning; knapsack; bin packing; portfolio optimization; TSP; maxcut; MIS; quantum optimization; QAOA; combinatorial optimization.

I. INTRODUCTION

Solving COPs is perceived as one of the major applications for the near future of quantum computation. There are three main reasons for this. First, COPs can be effectively encoded in Hamiltonians, where the ground state corresponds to the optimal solution of the problem [1, 2]. Second, COPs have practical applications and are hard to solve [3]. Third, quantum algorithms to solve these problems need few resources and can be tested on current state-of-the-art quantum hardware [4–6].

One of the most studied quantum algorithms for solving COPs is QAOA [7]. QAOA consists of p layers, each of which includes the COP cost Hamiltonian encoded in a parametric unitary gate with parameters γ_i and a "mixer" parametric unitary gate with parameters β_i. In this context, the parameters are adjusted to improve the probability of finding good solutions to the problem. Different techniques have been proposed to improve the performance of QAOA. For example, warm-start [8–10] is used to adjust QAOA in order to start it from solutions obtained by classical algorithms or recursive QAOA [11] which uses the QAOA to produce sequences of reduced problems. In [12, 13] are reviews of different techniques used in QAOA. To some extent, QAOA can be seen as a trotterized quantum annealing protocol where the number of layers p determines the precision of the solution [14, 15].

The most used technique to find the γ and β parameters in QAOA consists of a subroutine of classical optimization where QAOA provides samples used to obtain the expectation value of the cost Hamiltonian. Subsequently, a classical solver produces new γ and β parameters to minimize the expectation value. This process is repeated until a good set of parameters is found or other stopping criteria is reached. There are many limitations to this approach, e.g., finding the optimal set ofparameters has been proved to be an NP-Hard problem [16, 17]. Additionally, the energy landscape of the parameterized QAOA has many local minima where classical solvers can easily get trapped [16, 18]. Moreover, implementations on real hardware face an even greater challenge, the noise inherent in current quantum devices makes the search for the minima of the objective function unfeasible after only a few QAOA layers [19, 20].

An alternative strategy that partially overcomes these problems and has gained attention recently is the use

[*] Corresponding author: J. A. Montañez-Barrera; j.montanez-barrera@fz-juelich.de

of transferred parameters [21–23]. The idea is to find a good set of parameters using the classical optimization loop for small instances of a problem, where it is easier to find good parameters, and then, reuse them on larger instances. For instance, in [24], TL of 3-regular graphs in the MaxCut problem was initially introduced as an alternative to do extensive classical optimization for the QAOA parameters. Subsequently, [25] extended this concept to non-isomorphic unweighted graphs in the MaxCut problem and showed optimal QAOA parameter concentration for large graphs. In [26], different properties of random graphs in the MaxCut problem were identified as indicative characteristics of parameter transferability. In [27], numerical evidence supporting an approximation ratio exceeding the worst-case scenario by Goemans-Williamson (GW) was presented for parameters transferred in 3-regular graphs of MaxCut, specifically for $p < 12$. Furthermore, [6] provided indications of TL capabilities in the context of the weighted MaxCut problem. However, the prospects of TL between different COPs have not been systematically investigated to this point.

In this paper, we extend the study of TL capabilities of optimal QAOA parameters across different COPs. This is motivated by the fact that finding a good set of parameters that works for different COPs, independent of their construction or problem size, can be used in larger problems or as an initial guess for doing a subsequent classical optimization on specific problem instances. This initial guess is therefore not biased by the given problem class or specific instance. To this end, we select random instances of TSP, KP, BPP, PO, MaxCut, and MIS. We first use the Constrained Optimization BY Linear Approximation (COBYLA) [28] optimization method to find β and γ parameters for problems with up to 20 qubits.

We study the transfer of those parameters to (i) the same COP with up to 42 qubits and (ii) other random instances of completely different COPs. We thereby demonstrate that parameters can not only be transferred to larger instances of the same problem but also to completely different problems. We use the probability of finding the ground state, $probability(^*x)$, as the performance metric of the TL methodology. Of all the COPs studied, BPP is the one that shows the best TL capabilities.

Furthermore, we study the practical TL performance on various quantum technology platforms, namely Aspen-M3 from Rigetti, Harmony and Aria from IonQ, and ibm_brisbane from IBM with problem sizes 8, 14, and 18 qubits. We present solutions for the MIS using QAOA with p=10 and TL from BPP. Our results suggest that even in the 14 qubit case for QAOA with p=10, which corresponds to 640 CNOT gates, a positive net gain of TL may still be observable using IonQ Aria.

Additionally, we explore "cross-platform TL", by transferring the QAOA protocol to a quantum annealing protocol. To this end, we use a D-Wave quantum annealer [29, 30], D-Wave Advantage, with a modified custom anneal schedule that reflects the learned TL parameters to mimic the QAOA β and γ parameters separately. We study MIS from 100 to 170 qubits and find that the TL protocol of the β parameters performs consistently better than the default D-Wave Advantage annealing protocol for this particular problem.

The rest of the paper is organized as follows. Section II provides a description of the COPs used in this work, the TL methodology, the postprocessing technique for the real hardware implementation to mitigate some of the noise, and a description of the cross-platform TL approach. In Sec. III we present our results and a discussion. Finally, Sec. IV contains our conclusions. The source code for the results shown here can be found at `https://jugit.fz-juelich.de/qip/transfer-learning-QAOA`.

II. TRANSFER LEARNING IN QAOA

In the context of QAOA, we refer to TL as the use of pre-optimized $\gamma = [\gamma_0, \gamma_1, ..., \gamma_{p-1}]$ and $\beta = [\beta_0, \beta_1, ..., \beta_{p-1}]$ for parameters on problems that were not used for the optimization. In this methodology, the first step is to find an optimal set of parameters that works well for a specific problem. Then, we test if the optimized parameters work well on different instances of the same and other problems.

We study random instances of TSP, BPP, MIS, KP, PO, and MaxCut using QAOA with $p = 10$. We employ a quantum annealing initialization of the QAOA parameters [19, 31]. To find the minimum of the cost function, COBYLA is used with a maximum number of iterations given by max_iter $= 20N_qp$, where N_q is the number of qubits needed by the problem and p is the number of QAOA layers. Fig. (1) shows the methodology used for (a) the initialization of the γ and β parameters on the problem selected, (b) the loop of self-optimization where the parameters are updated improving the expectation value of the cost Hamiltonian of the problem, and (c) the TL of the parameters from the problem (BPP) on a new problem − in this case, the MIS. If the TL is successful, the QAOA circuit should sample good solutions for the new problem.

We pick 5 random instances for each problem size. For the TSP, we use instances with 3, 4, 5, and 6 cities (9, 16, 25, and 36 qubits), where the distances between cities are randomly chosen from a normal distribution with a mean value of 10 and a standard deviation of 0.1. This set of values is chosen because it has been shown to produce hard instances of the TSP [32]. In the BPP, we consider scenarios involving 3, 4, 5, and 6 items (12, 20, 30, and 42 qubits). The weight of each item is randomly chosen from integer values between 1 and 10, and 20 is the maximum weight of the bins. The MaxCut, MIS, KP, and PO problem sizes are 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, and 42. For MaxCut problems, we use randomly weighted edges with weights between 0 and 1 and a probability of having an edge between any two

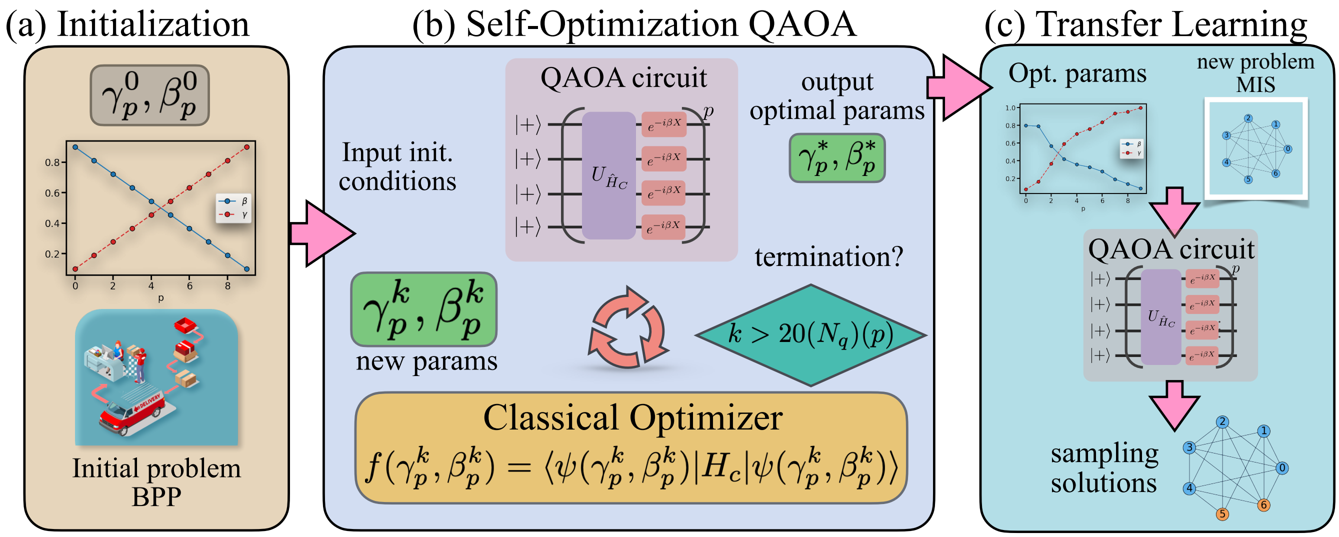

FIG. 1. Example of the TL methodology for transferring the parameters from BPP to MIS. (a) Quantum annealing initialization of the QAOA parameters for $p = 10$ layers using the BPP, (b) self-optimization step using QAOA, (c) final β and γ parameters transferred to the MIS problem.

vertices as 70%. In MIS problems, edges between nodes are randomly selected with a 50% probability of having an edge. For KP problems, item values are randomly chosen from integer values between 5 to 63, weights from 1 to 20, and the maximum weight is set to half of the sum of all weights. Finally, for PO, correlation matrix values are chosen randomly from $[-0.1, 0, 0.1, 0.2]$, asset costs varying between 0.5 and 1.5, and the budget is set to half of the total assets cost. For large problems, we simulate them using JUQCS–G software [15] on JUWELS Booster, a cluster of 3744 NVIDIA A100 Tensor Core GPUs, integrated into the modular supercomputer JUWELS [33, 34].

For the inequality constraints in the KP, PO, and BPP, we use the unbalanced penalization approach [35, 36]. Once the QUBO is generated with quadratic penalization terms and translated to the Ising Hamiltonian representation, the Hamiltonian is normalized in all the cases. This step is helpful to bring the problem Hamiltonian to a similar range of energy. A similar procedure is used by D-Wave annealers with the auto_scale functionality [37]. In Appendix A, we explain in detail the problems and the parameters used in this work.

To quantify the performance of TL, we use the probability of success, *probability*$(^*x)$. This is calculated from the final state vector $|\psi\rangle$ (after the QAOA protocol has been implemented) given by

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle, \tag{1}$$

where n is the number of qubits involved. The probability of success is calculated by

$$probability(^*x) = \sum_{x_i \in {^*x}} |\alpha_{x_i}|^2, \tag{2}$$

where $^*x$ is the set of all the optimal solutions of the given problem. The set of $^*x$ solutions has been previously obtained using the classical solver CPLEX [38]. Note that the *probability*$(^*x)$ is preferred here instead of the approximation ratio r as it has been shown that r can be a misleading metric in constrained problems [39].

Mitigation: Hamming distance 1

Our approach uses the Hamming distance 1 strategy as a post-processing technique to reduce errors in real quantum devices. This involves applying a bitflip to each position within the output bit-string, to mitigate single-qubit bitflips. The computational overhead of this post-processing method is linear, $O(NN_q)$, where N represents the number of samples and N_q is the number of qubits. It is important to note that the success of this method in improving the probability of the ground state relies on the large probability of obtaining the ground state compared to the number of samples used. Also, the error must be low enough to have only a single qubit error in the samples.

Cross-platform TL

In addition to TL between COPs, we also investigate the possibility of applying TL across platforms. In the case of quantum annealing, we want to test the capabilities to transfer the QAOA parameters for solving COPs using D-Wave Advantage. The default quantum annealing protocol on D-Wave Advantage, as proposed by Johnson et al. [40], is represented by

$$H(s) = -\frac{A(s)}{2}\sum_i \sigma_x^i + \frac{B(s)}{2}\,H_c, \qquad (3)$$

where $s \in [0,1]$ is a parameter that represents normalized time, H_c is the problem cost Hamiltonian (see Appendix A for details), $A(s)$ is the annealing protocol associated to the mixer, $B(s)$ is the annealing protocol of the cost Hamiltonian, and σ_x^i is the Pauli-x matrix for qubit i. The D-Wave device has hardware-dependent, fixed annealing functions $A(s)$ and $B(s)$ that cannot be changed. They are documented in spreadsheets that can be found for each device in the online D-Wave documentation. What can be changed by the user, however, is the mapping $s(t)$ between the normalized annealing fraction $s \in [0,1]$ and the real-time $t \in [500\,\mathrm{ns}, 2000\,\mu\mathrm{s}]$. The value of s is given by $s_i = t_i/t_f$, where t_i is the instantaneous real time and t_f is the total annealing time. D-Wave Advantage allows modification of the annealing schedule by specifying a maximum of 12 points for the relation $s_i = f(t_i)$ between a normalized time and instantaneous real-time. This flexibility enables us to implement a custom schedule for $A(s)$ or $B(s)$. For example, Fig. 2 shows a new relationship between the $s(t_i)$ and the time t_i. Hence, what originally happened at $t = 5.57\mu s$ in the default schedule now happens at $t = 2.2\mu s$ in the modified schedule.‘ We utilize parameters from the TL parameters of QAOA to modify the quantum annealing schedule on a D-Wave quantum annealer.

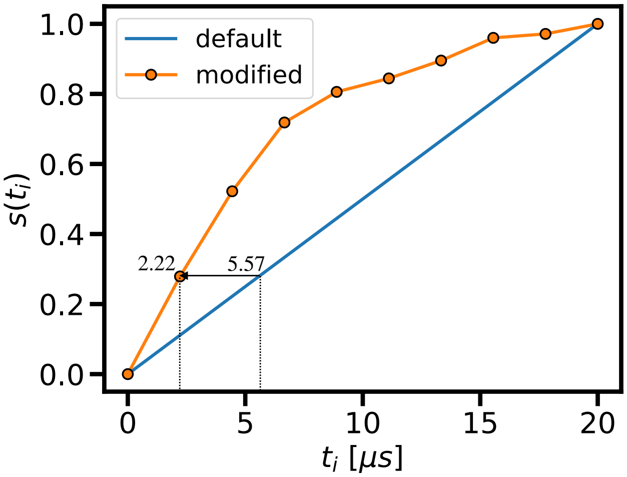

FIG. 2. D-Wave modified schedule example. The circles represent the modified point in the realtion between s_i and t_i.

III. RESULTS

In this section, we first study, by numerical simulation, TL from one COP to another COP. Using the best parameters, we then perform experiments on various quantum hardware devices. Finally, we investigate cross-platform TL, i.e., we learn parameters using a gate-based quantum computing model and then transfer them to a physical device implementing the quantum annealing model.

The initialization used for all problems was a linear ramp quantum annealing scheme (see Fig. 3 (a)). Fig. 3 (b) shows the final γ and β parameters of a random instance of the BPP. These parameters are used in subsequent plots to show the capabilities of TL from the BPP.

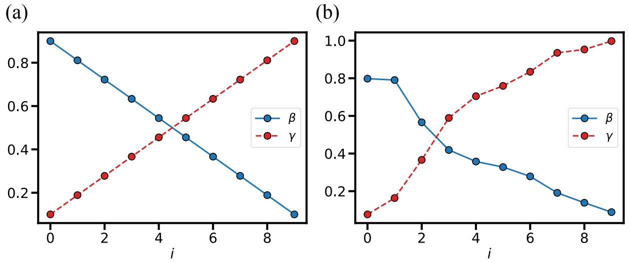

FIG. 3. Example for the QAOA parameter optimization of the BPP. (a) Quantum annealing initialization of the QAOA parameters for $p = 10$ layers, and (b) final β_i and γ_i for $i = 0, ..., p-1$ angles for the BPP with 3 items (12 qubits).

Figure 4 shows the classical optimization evolution for (a) BPP and (b)-(c) MaxCut problem using QAOA with $p = 10$. The cost represents the Hamiltonian energy at the given iteration. The optimization process effectively helps to improve the γ and β parameters reducing the average cost for both problems. The final schedules in the inset of Fig. 4(a)-(b) seem to differ in magnitude but some characteristics from the initial annealing schedule are conserved, i.e., β goes down to 0 and γ goes up to some value. Fig. 4-(c) shows the cost for the different seeds of problems sizes 10 and 14 qubits. For problem size 10 qubits, the optimization process seems not to have an effect after 500 iterations while for 3 out of 5 cases of the 14-qubit problems the optimization is still effective after 1000 iterations. This suggests that the number of iterations to find QAOA parameters is not only affected by p but also by the number of qubits.

Figure 5 shows a comparison between TL from a BPP with 3 items vs. self-optimization for the mean value of the optimal $probability(*x)$ of the different COPs. Self-optimization refers to the optimization of the γ and β parameters for each specific problem using COBYLA with a maximum number of iterations given by $max\_iter = 20pN_q$, where N_q is the number of qubits and p the number of QAOA layers. The election of max_iter is arbitrary with the main objective of setting a maximum number of iterations equal for problems with the same size. However, it could be beneficial for some of them to have more iterations while for others there is no improvement after a few hundred iterations.

The guiding black dotted line indicates a quadratic speedup over random guessing, i.e., a reduction in the search space to $\mathrm{O}(2^{N_q/2})$. The trend for all problems using QAOA is better than a quadratic speedup over ran-

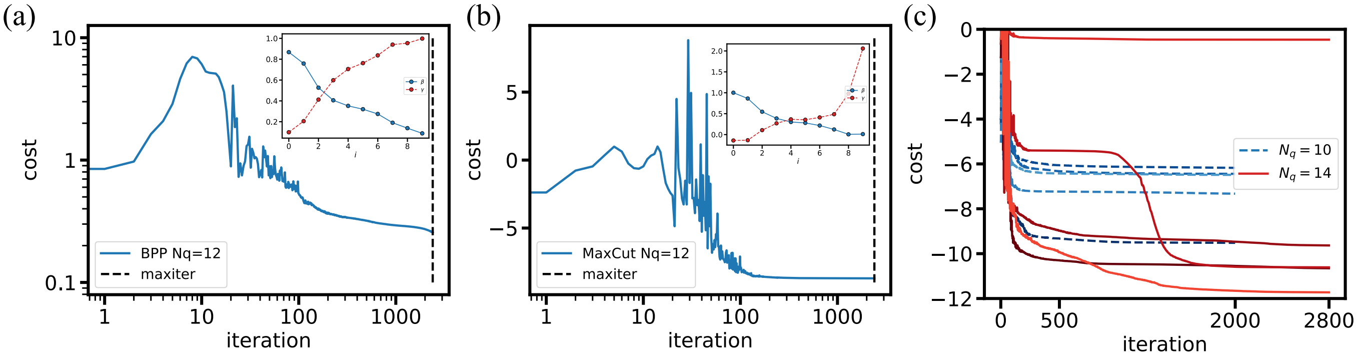

FIG. 4. (a)-(b)Classical optimization of the cost function value at each iteration for a random 12 qubit (3 items) BPP problem and 12 nodes MaxCut problem. (a) BPP (b) MaxCut. The inset plot shows the final parameters found for each problem. The maximum number of iterations is 2400. (c) MaxCut cost function versus the number of iterations for the classical optimization for problem sizes 10 and 14 qubits. Different lines represent the different seeds.

dom guessing. Comparing self-optimization and transfer learning, the probability of success is slightly better for the self-optimization method. However, this improvement does not fully compensate for the resources spent in the self-training step. For instance, the number of samples required to observe the optimal solution with a 99% probability, $N^{99\%}_{samples} = \log(1-0.99)/\log(1-probability(^*x))$, for the 14-qubit (18-qubit) MaxCut, is on average $N^{99\%}_{samples} = 16$ (28) for self-optimization and $N^{99\%}_{samples} = 20$ (41) for TL. The largest difference in number of samples is for 20-qubit BPP with $N^{99\%}_{samples} = 339$ for self-optimization and $N^{99\%}_{samples} = 848$ for TL. The number of iterations times the samples per iteration needs to be taken into account to get an estimate of the resources needed.

Figure 6 shows the results of applying TL to larger BPP problems with up to 42 qubits, i.e., we take the resulting 12-qubit parameters and apply the same QAOA schedule to significantly larger problems. The mean probability of success is above a quadratic speedup over random guessing for all the problems, which is a good indication of the generalization capabilities of the transferred BPP parameters. The best performance and scaling trend are given by the KP. This scaling is not necessarily related to the TL capabilities because self-optimization shows a similar performance up to the size tested. The improvement could be related to the KP energy structure. The lowest performance is for the MaxCut problem with an error bar going below the quadratic speedup over random guessing. For this problem, the scaling below 18 qubits seems similar to that of the KP, however, as the problem grows in size, the scaling trend changes considerably. It could be related to the maximum values of β and γ, and these values could be rescaled as the problem size grows, similar to the rescaling in [21]. The other problems seem to have similar performance that holds as the problem size grows.

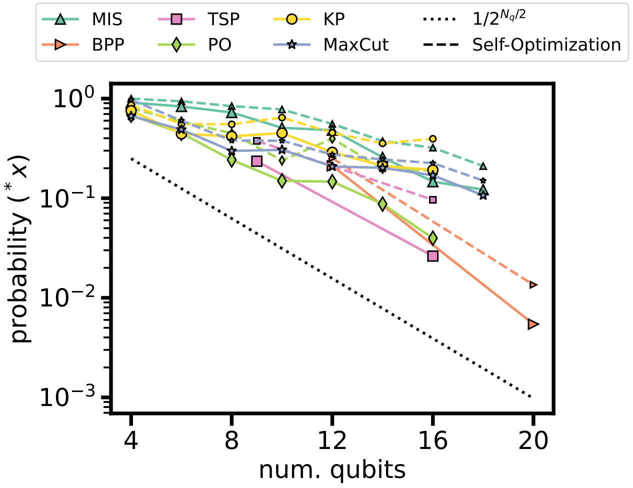

FIG. 5. Comparison between TL (solid line) and self-optimization (dashed line) for different COPs (see legend). Here, each marker represents the mean value over 5 random cases. The dashed lines with small markers represent the problems optimized using the procedure in Sec. II, and the solid lines with big markers represent the results of applying TL from the BPP. The quadratic speedup over random guessing (black dotted line) is presented as a guiding line.

Figure 7 shows the results of applying TL from different COPs to the 5 random instances from 4 to 18 qubits of (a) the MIS and (b) MaxCut. In Fig. 7 (a), we see that the best performance is obtained by TL parameters from an MIS instance (triangle-up) or a BPP instance (triangle-right). The first can be explained by the fact that different instances of the same problem (MIS) have similar Hamiltonian structures. However, the favorable results for the BPP constitute an interesting empirical observation with no obvious explanation for its generalization capabilities. Figure 7(b) shows that this observation also holds when transferring to the MaxCut problem, i.e., the best performance is obtained by TL from MaxCut to MaxCut (star) or again from BPP to MaxCut (triangle-right).

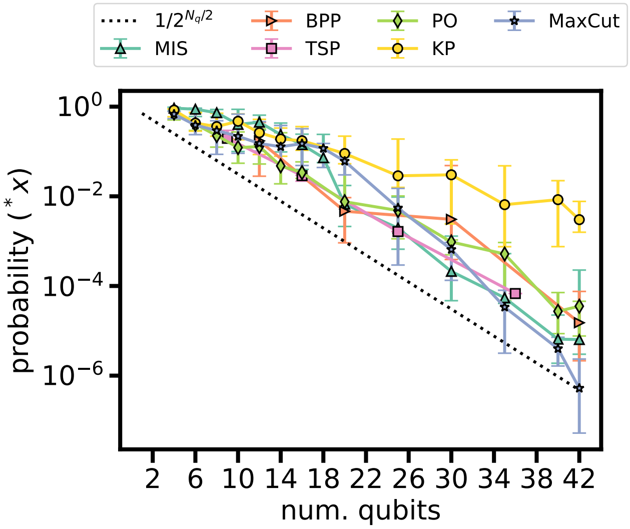

FIG. 6. TL from BPP of 3 items to different COPs. Shown is the mean value of 5 random instances of each problem size of the different COPs. Markers represent the mean value. Error bars represent quartiles Q1 and Q3. The guiding line is the same as in Fig. 5.

Solutions on quantum hardware

Next, we show how TL behaves when executing the QAOA algorithm for $p = 10$ on real quantum hardware using the BPP parameters from Fig. 3 (b). The problem used is the MIS for random problems with sizes 8, 14, and 18 qubits. The number of samples used is 1000 for the 8 qubit devices and 5000 for the 14 and 18 qubit devices.

In Fig. 8 (a), we show results for solving an 8-qubit MIS problem on Rigetti's Aspen-M-3 (2.9%, 27.9%), IBM's Brisbane (4.7% raw, 32.5%), IonQ's Harmony (3%, 21.2%), IonQ's Aria-1 (34.8%, 81.5%), an ideal simulator (89.1%, 94.0%), and random sampling (2.4%, 21%). In this problem, there are 240 CNOT gates on a fully connected device. We can see that at these large circuit depths, Aria is the only device with a distribution resembling the ideal case; all others are similar to a random bitstring generator.

In Fig. 8 (b), we show results for solving a 14-qubit MIS problem on Rigetti's Aspen-M-3 (0.0%, 0.2%), IBM's Brisbane (0.06%, 0.16%), IonQ's Aria-1 (4.46%, 12.3%), the ideal simulator (19.8%, 30.7%), and a Random Sampler (0%, 0.22%). The probability of connection between different nodes is set to 40%, and 640 CNOT gates are applied on a fully connected quantum device. Once again, the results indicate that the probability distribution of the ideal case is very different from the Brisbane and Aspen-M-3 results, whereas for Aria the resemblance with the ideal case is still observable.

Finally, in Fig. (8)-(c) we show results for solving an 18-qubit MIS problem on Rigetti's Aspen-M-3 (0.03%, 0.34%), IBM's Brisbane (0.0%, 0.03%), the ideal simulator (24.1%, 34.8%), and a Random Sampler (0.0%, 0.04%). In this case, the probability of connection between different nodes is 40%, $p = 10$, and the problem requires 1020 CNOT gates on a fully connected device (which corresponds to 3657 ECR gates after transpilation to IBM's Brisbane device). Although IonQ's Aria would also have enough qubits to run this case, testing it was prevented by a limitation in the number of 1 and 2-qubit gates to 950 and 650, respectively. In this case, there is a slight improvement in the Aspen-M-3 result compared to the Random Sampler, but it is not significant enough to conclude that partial information about the probability distribution is recoverable.

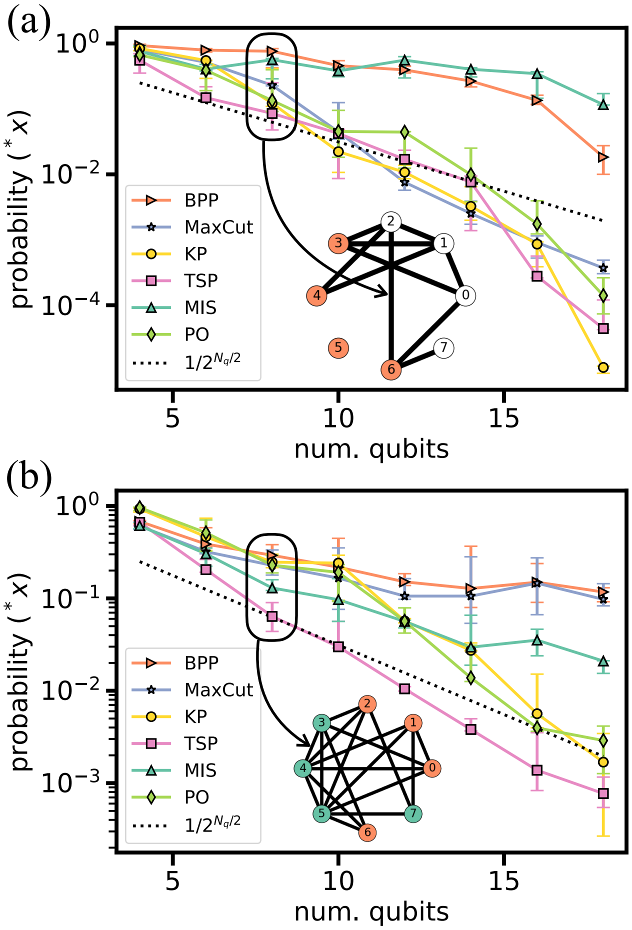

FIG. 7. TL from different COPs to (a) the MIS problem, (b) the MaxCut problem. Using the γ_p and β_p parameters from different COPs, 5 random instances for each problem size of the MIS are solved. Markers represent the median value and error bars represent the Q1 and Q3 quartiles. The insets represent one instance of the respective problem solved using the TL parameters of the other COPs.

Cross-platform TL

In this section, we study cross-platform TL by transferring the γ and β parameters of the 3 items BPP that

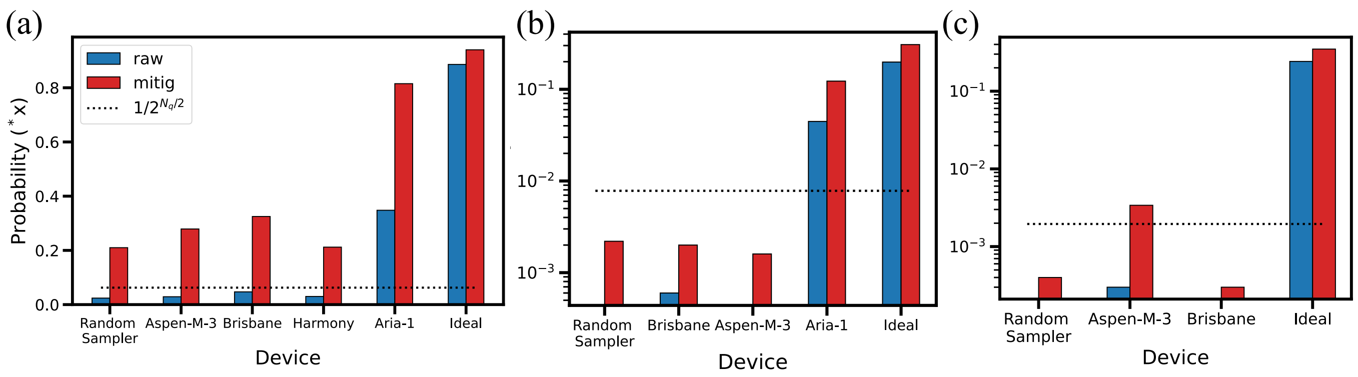

FIG. 8. Success rate to find the optimal solution for different MIS problem sizes using real devices (a) MIS with 8 qubits with 89.1% of ideal probability (b) MIS with 14 qubits and 19.8% ideal probability (c) MIS with 18 qubits and 24.1% ideal probability. The blue bars represent the raw sampling results using the different devices while the red bars are the percentage of bitstrings with optimal solution and Hamming distance 1 from the optimal solution. The dashed line represents a quadratic improvement in the probability of success compared to random guessing.

have shown good TL capabilities to a different quantum computing platform, namely a D-Wave Advantage quantum annealer. Figure 9 shows the two modified annealing schedules for $A(s)$ and $B(s)$, respectively. Note that we modify D-Wave Advantage $A(s)$ and $B(s)$ schedules separately because the device does not allow the modification of both of them at the same time. Fig. 9-(a) shows the TL of the γ parameters of the BPP to the schedule $B(s)$ in D-Wave Advantage. Fig. 9-(b) shows the TL of the β parameters of the BPP to the schedule $A(s)$ in D-Wave Advantage. A further restriction in the D-Wave Advantage schedule is that the initial and final annealing points must correspond to the $A(s)$ and $B(s)$ points.

Figure. 10 shows the TL to the MIS problem for problem sizes 100, 125, 150, 160, and 170 qubits. The results are obtained using 5000 samples for each problem size. Equation A18 is the QUBO formulation used to implement the problem and to evaluate the cost function. The green dotted lines are the results using the D-Wave default annealing schedule, the red triangle line is the TL of the mixer Hamiltonian parameters, and the blue circle line is the TL of the cost Hamiltonian parameters. Figure (10)-(a) shows the mean cost with error bars representing the standard deviation of the 5000 sample costs. Figure (10)-(b) shows the minimum cost of the 5000 samples for each problem size. These results show a consistent improvement in the distribution of solutions using TL of β QAOA parameters to the $A(s)$ parameters of the mixer Hamiltonian schedule of D-Wave Advantage both in terms of average and minimum value.

IV. CONCLUSIONS

We have presented transfer learning (TL) of QAOA parameters in the context of COP, a methodology that involves using pre-optimized QAOA parameters to solve different COPs. This method therefore does not require extra steps of classical optimization, but further parameters optimization using the TL parameters as a starting point could be beneficial. We show that the BPP has great generalization capabilities, i.e., the parameters for small instances of BPP are good for larger instances of the same problem and effective for instances of other COPs. The successful TL between instances of BPP to different COPs can be intuitively pictured as the consequence of optimizing an effective, digitized annealing schedule that works within the limit of a small number of layers (p = 10). This is the opposite of what happens with other COPs, where the parameters found do not perform well on other problems.

We test TL using KP, BPP, MIS, MaxCut, TSP, and PO. First, we do self-optimization for these problems, exploring random instances with up to 20 qubits. Then, we select the case with the best performance to find optimal solutions to the other problems. In our case, the parameters used are those of the BPP for 3 items (12 qubits). We use those parameters in different instances of the same and other problems for up to 42 qubits and find that for all of them, the probability of finding the ground state is above the quadratic speedup over random guessing. This suggests that there are β and γ parameters that generalize well over different COPs in their QUBO formulation.

Next, we show that coherent outputs for problem sizes up to 14 qubits are still present in current quantum technology. We test 3 different instances of the MIS problem with 8, 14, and 18 qubits with the devices used being Rigetti's Aspen-M-3, IBM's Brisbane, IonQ's Harmony, and Aria. In the case of 8 qubits, a direct comparison between the two generations of IonQ technology is possible, Harmony with a success probability of 3.0% and Aria 34.8%. This means an order of magnitude improvement between these two generations of IonQ trapped ions for sampling optimal solutions. There is still room for improvement and benchmarking with this methodology is a

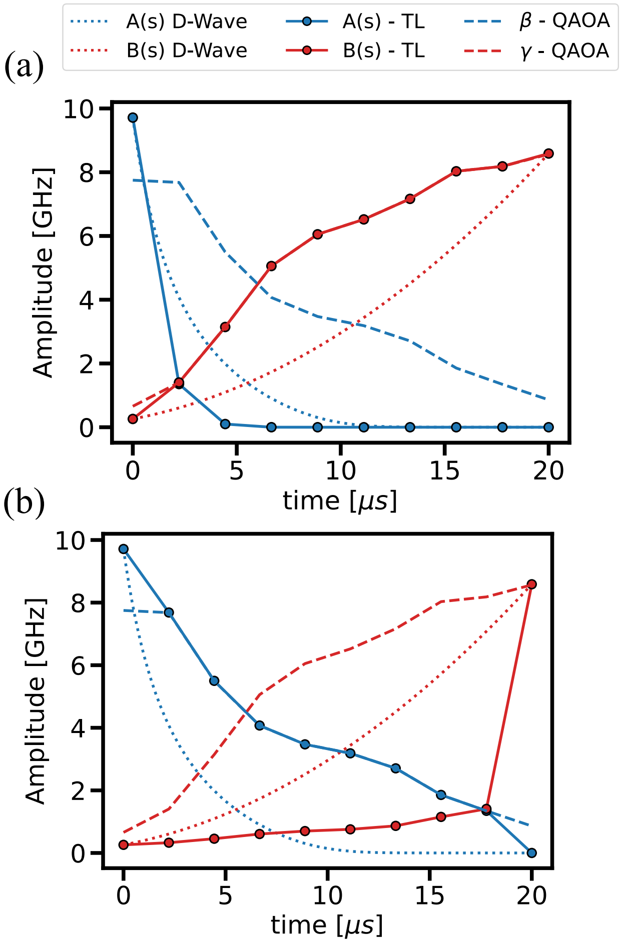

FIG. 9. Example for cross-platform TL parameters on the D-Wave Advantage protocol. (a) TL to the cost Hamiltonian schedule, and (b) TL to the mixer Hamiltonian schedule. The dotted line represents the default schedule used by D-Wave, the solid line is the new schedule with TL applied, and the dashed line reflects the BPP-QAOA schedule. The blue (red) color represents the mixer (cost) Hamiltonian protocol.

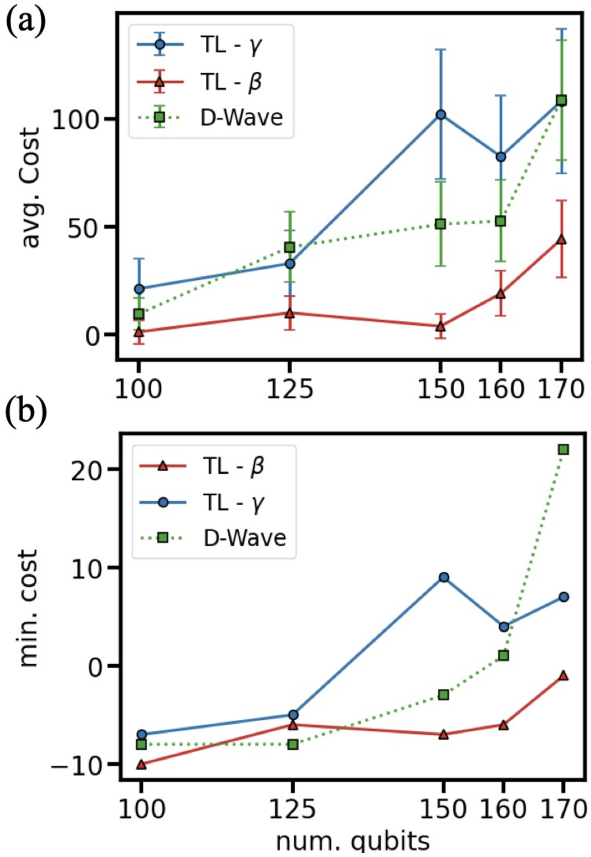

FIG. 10. Cross-platform TL for the MIS using BPP parameters (a) average value using 5000 samples on D-Wave Advantage (b) minimum cost of the 5000 samples.

promising tool to see the evolution of quantum technology for sampling.

Though self-optimization gives a better probability($*x$) than TL for every COP, this improvement does not fully compensate for the resources needed for the optimization loop needed to find the parameters. TL is also convenient for the classical simulation of large problems that typical computers cannot handle ($N_q > 30$). Compared to the self-optimization method it only requires one iteration compared to the hundreds of self-optimization. In the case of $N_q > 35$, only high-performance computing (HPC) systems can handle the simulations, making self-optimization computationally too expensive to even be considered. Self-optimization issues are more notorious in real quantum hardware where the cost of finding QAOA parameters is in terms of the number of samples and circuits used and classical solvers find it hard to work with noisy quantum hardware [41]. This makes TL a useful alternative that can compensate for the problems associated with the classical optimization loop even if TL parameters are not optimal.

Finally, we show that cross-platform TL is possible. We use D-Wave Advantage to test MIS problems between 100 and 170 qubits using the QAOA parameters of the BPP, Fig. 3. Two cases are tested, one with the modification of the mixer Hamiltonian B(s) and one with the modification of the cost Hamiltonian annealing protocol A(s). We find a consistent improvement in terms of the minimum and average cost using the mixer Hamiltonian modified schedule.

A future direction could be to extended the analysis of transfer learning in the context of Higher-Order Unconstrained Binary Optimization (HUBO) formulations of COPs, e.g., the one presented in [42].

ACKNOWLEDGMENTS

J. A. Montanez-Barrera acknowledges Unitary Fund for their support in the development of the combinatorial optimization library used in this work within OpenQAOA SDK. We acknowledge AWS Amazon Braket for the time used across the quantum devices Aspen-M3, Harmony, and Aria-1. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. J. A. Montanez-Barrera acknowledges support by the German Federal Ministry of Education and Research (BMBF), funding program Quantum Technologies - from basic research to market, project QSolid (Grant No. 13N16149). D. Willsch acknowledges support from the project Jülich UNified Infrastructure for Quantum computing (JUNIQ) that has received funding from the German Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the State of North Rhine-Westphalia. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

[1] A. Lucas, Ising formulations of many NP problems, Frontiers in Physics **2**, 1 (2014), arXiv:1302.5843.

[2] G. Kochenberger, J. K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, The unconstrained binary quadratic programming problem: A survey, Journal of Combinatorial Optimization **28**, 58 (2014).

[3] M. Ohzeki, Breaking limitation of quantum annealer in solving optimization problems under constraints, Scientific Reports **10**, 1 (2020), arXiv:2002.05298.

[4] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, Ben Chiaro, R. Collins, W. Courtney, S. Demura, A. Dunsworth, D. Eppens, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, S. Habegger, A. Ho, S. Hong, T. Huang, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, C. Jones, D. Kafri, K. Kechedzhi, J. Kelly, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, M. Lindmark, M. Leib, O. Martin, J. M. Martinis, J. R. McClean, M. McEwen, A. Megrant, X. Mi, M. Mohseni, W. Mruczkiewicz, J. Mutus, O. Naaman, C. Neill, F. Neukart, M. Y. Niu, T. E. O'Brien, B. O'Gorman, E. Ostby, A. Petukhov, H. Putterman, C. Quintana, P. Roushan, N. C. Rubin, D. Sank, A. Skolik, V. Smelyanskiy, D. Strain, M. Streif, M. Szalay, A. Vainsencher, T. White, Z. J. Yao, P. Yeh, A. Zalcman, L. Zhou, H. Neven, D. Bacon, E. Lucero, E. Farhi, and R. Babbush, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, Nature Physics **17**, 332 (2021), 2004.04197.

[5] P. Niroula, R. Shaydulin, R. Yalovetzky, P. Minssen, D. Herman, S. Hu, and M. Pistoia, Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer, Scientific Reports **12**, 10.1038/s41598-022-20853-w (2022).

[6] R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, and T. S. Humble, Parameter Transfer for Quantum Approximate Optimization of Weighted MaxCut, ACM Transactions on Quantum Computing **4**, 1 (2023), arXiv:2201.11785.

[7] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028 [quant-ph].

[8] D. J. Egger, J. Marecek, and S. Woerner, Warm-starting quantum optimization, Quantum **5**, 10.22331/q-2021-06-17-479 (2021), arXiv:2009.10095.

[9] R. Tate, M. Farhadi, C. Herold, G. Mohler, and S. Gupta, Bridging Classical and Quantum with SDP initialized warm-starts for QAOA, ACM Transactions on Quantum Computing **4**, 1 (2023), arXiv:2010.14021.

[10] R. Tate, J. Moondra, B. Gard, G. Mohler, and S. Gupta, Warm-Started QAOA with Custom Mixers Provably Converges and Computationally Beats Goemans-Williamson's Max-Cut at Low Circuit Depths, Quantum **7**, 10.22331/q-2023-09-26-1121 (2023), arXiv:2112.11354.

[11] J. R. FinŽgar, A. Kerschbaumer, M. J. Schuetz, C. B. Mendl, and H. G. Katzgraber, Quantum-Informed Recursive Optimization Algorithms, PRX Quantum **5**, 1 (2024), arXiv:2308.13607.

[12] A. Abbas, A. Ambainis, B. Augustino, A. Bärtschi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen, N. Franco, F. Fratini, B. Fuller, J. Gacon, C. Gonciulea, S. Gribling, S. Gupta, S. Hadfield, R. Heese, G. Kircher, T. Kleinert, T. Koch, G. Korpas, S. Lenk, J. Marecek, V. Markov, G. Mazzola, S. Mensa, N. Mohseni, G. Nannicini, C. O'Meara, E. P. Tapia, S. Pokutta, M. Proissl, P. Rebentrost, E. Sahin, B. C. B. Symons, S. Tornow, V. Valls, S. Woerner, M. L. Wolf-Bauwens, J. Yard, S. Yarkoni, D. Zechiel, S. Zhuk, and C. Zoufal, Quantum optimization: Potential, challenges, and the path forward (2023), arXiv:2312.02279 [quant-ph].

[13] Z. He, R. Shaydulin, D. Herman, C. Li, R. Raymond, S. H. Sureshbabu, and M. Pistoia, Parameter setting heuristics make the quantum approximate optimization algorithm suitable for the early fault-tolerant era (2024), arXiv:2408.09538 [quant-ph].

[14] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics **3**, 625 (2021), arXiv:2012.09265.

[15] D. Willsch, M. Willsch, F. Jin, K. Michielsen, and H. De Raedt, GPU-accelerated simulations of quantum annealing and the quantum approximate optimization algorithm, Computer Physics Communications **278**, 108411 (2022), arXiv:2104.03293.

[16] L. Bittel and M. Kliesch, Training Variational Quantum Algorithms Is NP-Hard, Physical Review Letters **127**, 120502 (2021), arXiv:2101.07267.

[17] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte, Reachability deficits in quantum approximate optimization, Phys. Rev. Lett. **124**, 090504 (2020).

[18] V. Kremenetski, T. Hogg, S. Hadfield, S. J. Cotton, and N. M. Tubman, Quantum alternating operator ansatz (qaoa) phase diagrams and applications for quantum chemistry (2021), arXiv:2108.13056 [quant-ph].

[19] L. Zhou, S. T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, Physical Review X **10**, 1 (2020), arXiv:1812.01041.

[20] G. Koßmann, L. Binkowski, L. van Luijk, T. Ziegler, and R. Schwonnek, Deep-circuit qaoa (2023), arXiv:2210.12406 [quant-ph].

[21] R. Shaydulin, C. Li, S. Chakrabarti, M. DeCross, D. Herman, N. Kumar, J. Larson, D. Lykov, P. Minssen, Y. Sun, Y. Alexeev, J. M. Dreiling, J. P. Gaebler, T. M. Gatterman, J. A. Gerber, K. Gilmore, D. Gresh, N. Hewitt, C. V. Horst, S. Hu, J. Johansen, M. Matheny, T. Mengle, M. Mills, S. A. Moses, B. Neyenhuis, P. Siegfried, R. Yalovetzky, and M. Pistoia, Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem, Science Advances **10**, 10.1126/sciadv.adm6761 (2024).

[22] J. Basso, E. Farhi, K. Marwaha, B. Villalonga, and L. Zhou, The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model, Leibniz International Proceedings in Informatics, LIPIcs **232**, 1 (2022), arXiv:2110.14206.

[23] S. Boulebnane and A. Montanaro, Solving boolean satisfiability problems with the quantum approximate optimization algorithm (2022), arXiv:2208.06909 [quant-ph].

[24] F. G. S. L. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances (2018), arXiv:1812.04170 [quant-ph].

[25] P. C. Lotshaw, T. S. Humble, R. Herrman, J. Ostrowski, and G. Siopsis, Empirical performance bounds for quantum approximate optimization, Quantum Information Processing **20**, 10.1007/s11128-021-03342-3 (2021).

[26] A. Galda, X. Liu, D. Lykov, Y. Alexeev, and I. Safro, Transferability of optimal QAOA parameters between random graphs, Proceedings - 2021 IEEE International Conference on Quantum Computing and Engineering, QCE 2021 , 171 (2021), arXiv:2106.07531.

[27] J. Wurtz and D. Lykov, The fixed angle conjecture for qaoa on regular maxcut graphs (2021), arXiv:2107.00677 [quant-ph].

[28] M. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, Advances in Optimization and Numerical Analysis **275**, 51 (1994).

[29] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, *et al.*, Quantum annealing with manufactured spins, Nature **473**, 194 (2011).

[30] A. D. King, J. Raymond, T. Lanting, R. Harris, A. Zucca, F. Altomare, A. J. Berkley, K. Boothby, S. Ejtemaee, C. Enderud, E. Hoskinson, S. Huang, E. Ladizinsky, A. J. R. MacDonald, G. Marsden, R. Molavi, T. Oh, G. Poulin-Lamarre, M. Reis, C. Rich, Y. Sato, N. Tsai, M. Volkmann, J. D. Whittaker, J. Yao, A. W. Sandvik, and M. H. Amin, Quantum critical dynamics in a 5,000-qubit programmable spin glass, Nature **617**, 61–66 (2023).

[31] S. H. Sack and M. Serbyn, Quantum annealing initialization of the quantum approximate optimization algorithm, Quantum **5**, 1 (2021), arXiv:2101.05742.

[32] P. Cheeseman, B. Kanefsky, and W. M. Taylor, Where the Really Hard Problems Are, The 12nd International Joint Conference on Artificial Intelligence , 331 (1991).

[33] D. Krause, JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre, Journal of large-scale research facilities JLSRF **5**, A135 (2019).

[34] D. Alvarez, JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre, J. of Large-Scale Res. Facil. **7**, A183 (2021).

[35] J. A. Montañez-Barrera, D. Willsch, A. Maldonado-Romo, and K. Michielsen, Unbalanced penalization: a new approach to encode inequality constraints of combinatorial problems for quantum optimization algorithms, Quantum Science and Technology **9**, 025022 (2024).

[36] J. A. Montanez-Barrera, P. van den Heuvel, D. Willsch, and K. Michielsen, Improving Performance in Combinatorial Optimization Problems with Inequality Constraints: An Evaluation of the Unbalanced Penalization Method on D-Wave Advantage, 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) **01**, 535 (2023), arXiv:2305.18757.

[37] D-Wave Systems Inc., Solver parameters (2024), accessed: 2024-08-27.

[38] C. Bliek, P. Bonami, and A. Lodi, Solving mixed-integer quadratic programming problems with ibm-cplex : a progress report (2014).

[39] S. Schulz, D. Willsch, and K. Michielsen, Guided quantum walk, Physical Review Research **6**, 1 (2024), arXiv:2308.05418.

[40] M. W. Johnson, P. Bunyk, F. Maibaum, E. Tolkacheva, A. J. Berkley, E. M. Chapple, R. Harris, J. Johansson, T. Lanting, I. Perminov, E. Ladizinsky, T. Oh, and G. Rose, A scalable control system for a superconducting adiabatic quantum optimization processor, Superconductor Science and Technology **23**, 1 (2010), arXiv:0907.3757.

[41] X. Bonet-Monroig, H. Wang, D. Vermetten, B. Senjean, C. Moussa, T. Bäck, V. Dunjko, and T. E. O'Brien, Performance comparison of optimization methods on variational quantum algorithms, Physical Review A **107**, 10.1103/PhysRevA.107.032407 (2023), arXiv:2111.13454.

[42] A. Glos, A. Krawiec, and Z. Zimborás, Space-efficient binary optimization for variational quantum computing, npj Quantum Information **8**, 1 (2022).

[43] M. C. Giron, G. Korpas, W. Parvaiz, P. Malik, and J. Aspman, Approaching Collateral Optimization for NISQ and Quantum-Inspired Computing (May 2023), IEEE Transactions on Quantum Engineering **4**, 10.1109/TQE.2023.3314839 (2023).

[44] J. Yan, Y. Lu, L. Chen, S. Qin, Y. Fang, Q. Lin, T. Moscibroda, S. Rajmohan, and D. Zhang, Solving the batch stochastic bin packing problem in cloud: A chance-constrained optimization approach, in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22 (ACM, 2022) p. 2169–2179.

[45] M. Kroes, L. Petrica, S. Cotofana, and M. Blott, Evolutionary bin packing for memory-efficient dataflow inference acceleration on FPGA, GECCO 2020 - Proceedings of the 2020 Genetic and Evolutionary Computation Conference , 1125 (2020), arXiv:2003.12449.

Appendix A: Supplementary Material

In this section, a description of the different problems used in this work is presented. For each problem, the constraints are encoded using penalization terms. We use squared penalty terms for equality constraints, and the unbalanced penalization approach [35] for inequality constraints.

1. QUBO Formulation

One approach to represent combinatorial optimization problems is through the Quadratic Unconstrained Binary Optimization (QUBO) formulation. The QUBO formulation expresses the problem as a quadratic objective function that depends on binary variables. The objective is to minimize this function by determining the values of the binary variables, subject to certain constraints. Combinatorial problems that can be represented by the QUBO formulation have functions of the form

$$f(\mathbf{x}) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} q_{ij} x_i x_j. \tag{A1}$$

Here, n represents the number of variables, $q_{ij} \in \mathbb{R}$ are coefficients associated with the specific problem and $x_i \in \{0, 1\}$ are the binary variables. It is important to note that in this formulation, $x_i x_i \equiv x_i$ and $q_{ij} = q_{ji}$.

A general form of a combinatorial optimization problem that can be solved using Quantum Processing Units (QPUs) is characterized by a cost function

$$f(\mathbf{x}) = 2\sum_{i=0}^{n-1}\sum_{j>i} q_{ij} x_i x_j + \sum_{i=0}^{n-1} q_{ii} x_i, \tag{A2}$$

and additionally, linear equality constraints are given by

$$\sum_i c_i x_i = C, \quad c_i \in \mathbb{Z}, \tag{A3}$$

and linear inequality constraints are given by

$$\sum_i w_i x_i \leq W, \quad w_i \in \mathbb{Z}, \tag{A4}$$

can be added. Here, q_{ij}, c_i, and w_i are parameters of the problem. To transform problems with constraints into the QUBO formulation, the constraints are usually incorporated as penalization terms. The equality constraints are included in the cost function using a penalization term of the form

$$\lambda_0 \left(\sum_i c_i x_i - C \right)^2. \tag{A5}$$

Here, λ_0 is a penalization coefficient that should be chosen appropriately to obtain sufficient solutions that satisfy the equality constraint, and C is a constant value given by the constraint.

For the inequality constraints, we use the *unbalanced penalization* [35] encoding which is a heuristic method for including inequality constraints as penalization terms in the QUBO formulation of combinatorial optimization problems. The method has been shown to outperform the slack variables encoding for the TSP, BPP, KP, and collateral optimization [35, 36, 43]. Starting from Eq.(A4), the method adds a penalization term $\xi(\mathrm{x})$ to the objective function given by

$$\xi(\mathrm{x}) = -\lambda_1 h(\mathrm{x}) + \lambda_2 h(\mathrm{x})^2, \tag{A6}$$

where $h(\mathrm{x}) = W - \sum_i w_i x_i$ and $\lambda_{1,2}$ are penalization coefficients that should be chosen to guarantee that the constraint is fulfilled. The term $\xi(\mathrm{x})$ is unbalanced, meaning that it imposes a larger penalization for negative values of $h(\mathrm{x})$ (i.e., when the constraint is not satisfied) than for positive values. The QUBO formulation using the unbalanced penalization approach is given by

$$\min_x \left(2\sum_{i,j>i} q_{ij}x_i x_j + \sum_i q_{ii}x_i + \lambda_0 \left(\sum_i c_i x_i - C \right)^2 - \lambda_1 h(x) + \lambda_2 h(x)^2 \right). \tag{A7}$$

The parameters for the different problems studied in this work are shown in Table I. In general, this method does not guarantee that the optimal solution is encoded in the ground state of the Ising Hamiltonian. For the probability of the BPP, PO, and KP in Figs. 5 and 6, we choose as optimal solution the ground state of the new Ising Hamiltonian that in the majority of the cases is the optimal solution of the original problem. The last step to represent the QUBO problem as an Ising Hamiltonian is to change the x_i variables to spin variables $z_i \in \{1, -1\}$ by the transformation $x_i = (1 - z_i)/2$. Note that Eq.(A7) can ultimately be rewritten as Eq.(A2) plus a constant value. Hence, Eq.(A2) represented in terms of the Ising model reads

$$H_c(\mathrm{z}) = \sum_{i=0}^{n-1}\sum_{j>i}^{n-1} J_{ij}z_i z_j + \sum_{i=0}^{n-1} h_i z_i + \text{offset}, \tag{A8}$$

where J_{ij} and h_i are real coefficients that represent the combinatorial optimization problem, and the offset is a constant value. Since the offset does not affect the location of the optimal solution, it can be left out for the sake of simplicity. In the following subsection, the COPs are presented. These problems can be translated into the Ising Hamiltonian representation following the methodology presented in this section. The last step we use to solve the problems using QAOA is to normalize the Hamiltonian by the maximum weight in the Hamiltonian, i.e., $\max\{J_{ij}, h_i\}$.

TABLE I. Parameters $\lambda_{0,1,2}$ for the TSP, BPP, KP, PO, and MIS used to translate the combinatorial optimization problems into the QUBO representation using the unbalanced penalization approach (Eq. (A7)). For all equality constraints of each problem, we use the same λ_0, and for the inequality constraints the same $\lambda_{1,2}$.

	λ_0	λ_1	λ_2
TSP	23	-	-
BPP	15	4.2	0.4
KP	-	0.96	0.04
PO	-	0.97	0.06
MIS	-	1	1

2. Combinatorial optimization problems

a. Traveling salesman problem

The TSP is a well-known combinatorial optimization problem that aims to determine the shortest possible route to visit a given set of cities and return to the starting city. This problem has various practical applications, including route planning, circuit board drilling, and DNA sequencing. A QUBO formulation of the TSP can be obtained using a time encoding of the route that the traveler passes on a Hamiltonian cycle [1]. For the asymmetric and symmetric forms, this TSP formulation requires n^2 variables for n cities (we note that in principle, one can reduce this to $(n-1)^2$ variables by fixing the starting point). It needs $2n$ equality constraints and avoids complications associated with sub-tours. The TSP formulation is given by

$$\min \sum_{t=0}^{n-1}\sum_{i=0}^{n-1}\sum_{j\neq i,j=0}^{n-1} c_{ij}x_{i,t}x_{j,t+1}, \tag{A9}$$

subject to the set of constraints,

$$\sum_{i=0}^{n-1} x_{i,t} = 1 \qquad \forall t = 0, ..., n-1, \tag{A10}$$

and

$$\sum_{t=0}^{n-1} x_{i,t} = 1 \qquad \forall i = 0, ..., n-1. \tag{A11}$$

Equation (A10) expresses that at every time t exactly one city is visited, while Eq.(A11) expresses that every city i is visited at exactly one time. For this problem, we use instances with 3, 4, 5, and 6 cities, where the distances between cities c_{ij} are randomly chosen from a normal distribution with a mean equal to 10 and a standard deviation equal to 0.1.

b. Knapsack Problem

The KP involves selecting a subset of items from a larger set, each with a certain weight w_i and value v_i, in such a way that the total weight does not exceed a given limit while maximizing the total value. Formally, the cost function is

$$\max \sum_{i=0}^{n-1} v_i x_i, \tag{A12}$$

subject to the single inequality constraint,

$$\sum_{i=0}^{n-1} w_i x_i \leq W, \tag{A13}$$

where $x_i = 1, 0$ indicates that an item is included or not, and W is the maximum weight. We select item values v_i ranging from 5 to 63 randomly, weights w_i from 1 to 20 randomly, and maximum weight $W = \frac{1}{2} \sum_i w_i$.

c. Portfolio Optimization

The goal of PO is to create a balanced portfolio out of a selection of financial assets, which should maximize future returns while taking into account the total risk of the investment. The information we have about the assets is their past returns μ_i and the covariances between assets σ_{ij}, with which the problem can be formulated as follows

$$\max \sum_{i=0}^{n-1} \mu_i x_i - q \sum_{i=0}^{n-1} \sigma_{ij} x_i x_j, \tag{A14}$$

subject to the inequality constraint

$$\sum_{i=0}^{n-1} c_i x_i \leq B, \tag{A15}$$

where (similar to the KP), the x_i indicates whether an asset is selected as part of the portfolio or not, and B is the total budget. The factor q controls how much risk is taken. If it is small, the second term in Eq.(A14) becomes negligible and the returns μ_i will be dominant in determining the optimal solutions. In the main text, problem sizes ranging from 4 to 42 are presented. The values of the expected return μ_i are randomly chosen between 0 and 1. The correlation matrix $\sigma_{i,j}$ is selected randomly from the set $[-0.1, 0, 0.1, 0.2]$. Asset costs c_i are randomly chosen between 0.5 and 1.5. The budget is set as $B = \frac{1}{2} \sum_i c_i$.

d. Maximal Independent Set

The MIS problem asks to find the largest subset of vertices of a graph, such that no two vertices in the subset are adjacent. This subset is then called independent. Formally, for an undirected graph $G = (V, E)$, the problem formulation is

$$\max \sum_{v \in V} x_v, \tag{A16}$$

subject to

$$x_u + x_v \leq 1 \qquad \forall (u, v) \in E, \tag{A17}$$

where the binary variable x_v determines whether a vertex is included in the subset or not. We select problem sizes between 4 and 42 qubits and the probability of having an edge between nodes of 50%. The constraints in this case are added to the QUBO formulation as $2x_ix_j$ if an edge is present. Therefore, the QUBO formulation for this problem is given by

$$\min \sum_{i \in V} -x_i + \sum_{i,j \in E} 2x_ix_j. \tag{A18}$$

e. Bin Packing Problem

The BPP involves the efficient packing of a collection of items into the minimum number of bins, where each item has an associated weight and the bins have a maximum weight capacity. This problem finds applications in various real-world scenarios like container scheduling [44], and FPGA chip design [45] among others. The BPP is classified as an NP-hard problem due to its computational complexity. The problem can be formulated as follows, minimize the total number of bins used given by the objective function

$$\min \sum_{j=0}^{m-1} y_j, \tag{A19}$$

subject to the following constraints. Each bin's weight capacity should not be exceeded

$$\sum_{i=0}^{n-1} w_i x_{ij} \leq W y_j \quad \forall j = 0, ..., m-1, \tag{A20}$$

and each item can only be assigned to one bin

$$\sum_{j=0}^{m-1} x_{ij} = 1 \quad \forall i = 0, ..., n-1. \tag{A21}$$

Binary variables indicating item-bin assignments and bin utilization

$$x_{ij} \in 0, 1 \quad \forall i = 0, .., n-1 \quad \forall j = 0, .., m-1, \tag{A22}$$

$$y_j \in 0, 1 \quad \forall j = 0, .., m-1. \tag{A23}$$

In the above equations, n represents the number of items (nodes), m represents the number of bins, w_i is the weight of the i-th item, W denotes the maximum weight capacity of each bin, and x_{ij} and y_j are binary variables representing the presence of item i in bin j and the use of bin j, respectively. The objective function in Eq.(A19) aims to minimize the number of bins used, while Eq.(A20) enforces the constraint on bin weight capacity. Eq.(A21) ensures that each item is assigned to only one bin, and Eqs. (A22) and (A23) define the binary nature of variables x_{ij} and y_j. The main text considers scenarios involving 3, 4, 5, and 6 items. The weight of each item w_i is randomly chosen from 1 to 10, and 20 is the maximum weight W of the bins. The Lagrange multipliers $\lambda_{0,1,2}$ in Eq.(A7) for this problem are 15, 4.2, and 0.4, respectively.

f. Maximum Cut

The MaxCut problem involves determining the partition of the vertices in an undirected graph such that the total weight of the edges between the two sets is maximized. For an undirected graph $G = (V, E)$, the problem is formulated as

$$\max \sum_{(i,j) \in E} w_{ij}(x_i + x_j - 2x_ix_j), \tag{A24}$$

where w_{ij} represents the weight of the edge between vertices i and j, and x_i and x_j are binary variables that determine the partition of vertices. The goal is to maximize the sum of edge weights over all edges in the cut. The binary variables x_i and x_j take values of 0 or 1, indicating the membership of vertices in different sets of the partition. If x_i and x_j are different, the edge contributes to the objective function.