

Efficient Circuit Transpilation of Commuting Gates on 2D Lattices

Sabina Drăgoi^{1,2} and Daniel J. Egger²

¹*Institute for Theoretical Physics, ETH Zürich, Wolfgang-Pauli-Str. 27, 8093 Zürich, Switzerland*

²*IBM Quantum, IBM Research - Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland*

I. Introduction

Combinatorial optimization problems are ubiquitous across science and industry [1–7], yet many of them remain computationally challenging due to their NP-complete nature [8, 9]. Classical methods therefore rely on heuristics or approximation algorithms [10, 11] whose performance can degrade with problem size [12–15].

Quantum computing offers a promising alternative, with variational algorithms such as the Quantum Approximate Optimization Algorithm (QAOA) designed to tackle such problems by encoding them into Hamiltonians [16–18]. A key challenge of implementing QAOA is the depth of the resulting quantum circuits. The problem Hamiltonian typically induces interactions between many pairs of qubits, which are implemented on digital quantum computers using two-qubit gates. On hardware with limited connectivity, additional SWAP operations are required to route qubits into adjacent positions. This routing process significantly increases circuit depth and accumulated error, making efficient transpilation crucial.

Transpilation is the process of converting a high-level quantum circuit into a logically-equivalent one that the hardware can interpret. Finding an optimal transpilation for general circuits is itself an NP-complete problem [19–21]. However, this task can be simplified for circuits composed of commuting two-qubit gates, such as those arising in QAOA. Current state-of-the-art transpilers take advantage of this commutativity property to design a pre-determined network of SWAP gates, called a SWAP strategy, which can systematically generate all pairwise interactions [22, 23]. While this approach was proved to be practical on simple hardware connectivities, such as a line of qubits, designing SWAP strategies for higher connectivities is a considerably harder task. Furthermore, taking the pre-determined SWAP network approach can also be suboptimal for sparse graph problems which might require many SWAPs for a few gates.

We address these challenges by introducing a problem-dependent transpilation method for such circuits on two-dimensional lattice architectures. Our approach constructs SWAP sequences that reduce circuit volume and improve hardware performance. The outperformance of our method over the standard one improves with system size for both Random Regular graphs of degree 3 (RR3) and Erdős–Rényi (ER) graphs. Hardware experiments further confirm that our transpilation method is more noise robust, leading to improved performance when using QAOA to solve combinatorial optimization problems.

II. Quantum Approach

We consider the Max-Cut problem which is an unconstrained combinatorial optimization problem defined over n binary decision variables $x \in \{0, 1\}^n$ and often used to benchmark QAOA performance [24–27]. The goal is to split a set V of n nodes of a graph $G = (E, V)$ in two parts such that the sum of the edges connecting the two subsets is maximized. Formally, a decision variable x_i is assigned to each node $i \in V$ to indicate which side of the cut the node is on. Solving Max-Cut is then equivalent to finding the binary string x that maximizes

$$f_{\text{Max-Cut}}(x) = \sum_{(i,j) \in E} (x_i(1 - x_j) + x_j(1 - x_i)). \quad (1)$$

Quantum computers can tackle such problems by mapping $f(x)$ to an Ising Hamiltonian H_C whose ground state minimizes $f(x)$. This is done through the change of variables $x_i = \frac{1 - z_i}{2}$ and promoting z_i to Pauli Z operators Z_i [17]. Mapping $f(x)$ to a Hamiltonian [17] results in the cost operator

$$H_{\text{Max-Cut}} = \frac{1}{2} \sum_{(i,j) \in E} Z_i Z_j, \quad (2)$$

with a ground state that maps to the maximum cut.

Then, we use QAOA to find the ground state of H_C by sampling from an Ansatz

$$|\psi(\beta, \gamma)\rangle = \prod_{m=1}^p e^{-i\beta_m H_M} e^{-i\gamma_m H_C} |+\rangle^{\otimes n} \quad (3)$$

with optimized variational parameters (β, γ) . In QAOA, the mixing Hamiltonian H_M is chosen such that its ground state is the initial state. This ensures that an adiabatic evolution, obtained in the limit of $p \rightarrow \infty$, transforms the initial ground state of H_M to the ground state of H_C . Here, we consider a problem-independent mixer operator $H_M = -\sum_{i=0}^{n-1} X_i$, although other mixing Hamiltonians have been proposed [28, 29].

III. Grid-based SWAP Strategies

The transpilation problem consists of two main tasks: determining an initial qubit mapping and finding a sequence of SWAP layers that enables all required interactions. The inputs are the hardware connectivity, a ‘basis’ set of SWAP layers, and the interaction graph instance. A ‘basis’ of SWAP layers is defined as a set for which there exists an ordering of its layers that realizes any desired pairwise interaction, see Fig. 1(a). The state-of-the-art approach for transpiling on higher-connectivity

hardware first identifies the longest line of qubits and then applies problem-independent linear SWAP layers referred to here as the **linear** strategy.

We propose a transpilation method, called **greedy**, that constructs SWAP strategies *tailored* to the specific interaction graph. The method begins with an initial qubit mapping obtained via a SAT-based approach applied to a fixed SWAP strategy [23] (see Fig. 1(b)). Starting from this mapping, we iteratively build a sequence of SWAP layers by exploring the tree of possible qubit configurations up to a predefined depth $k_{\max}$.

At each step, we track which interactions can be executed and prioritize SWAP sequences that maximize the number of newly enabled interactions. If a sequence fully realizes all required interactions, it is accepted as a solution, with random tie-breaking if needed. Otherwise, we greedily extend the most promising partial sequences and repeat the exploration. This procedure yields a problem-dependent SWAP sequence that is typically shorter than fixed strategies. To further improve performance, we iteratively recompute the initial qubit mapping from the current SWAP sequence and repeat the search. Alternating between optimizing the mapping and the routing helps escape from suboptimal configurations.

IV. Results

In this section, we quantify the advantages of our transpilation method and identify the regimes where it provides the greatest benefit. For benchmarking, we consider only the time-evolution under the cost Hamiltonian. As a result, the number of SWAP layers and entangling gates corresponds to that of a depth-one QAOA circuit.

We first study RR3 graphs, a standard benchmark for QAOA. Our problem-dependent transpilation consistently outperforms existing approaches in terms of the number of SWAP layers, see Fig. 2(a). This metric is directly relevant, as it correlates with the number of CZ gates in the compiled circuit, Fig. 2(b). Importantly, the advantage persists across system sizes and becomes more pronounced for larger graphs, see Fig. 3(d).

We next consider ER graphs with edge probability q . Here, the reduction in CZ gates compared to a linear SWAP strategy increases with both graph size and density, see Fig. 2(c). However, denser graphs also lead to larger and noisier circuits. For hardware with error rates on the order of 10^{-3} , circuits exceeding $\sim 3,600$ gates become effectively unusable. The white and black dashed lines in Fig. 2(c) indicate the feasibility limits for the linear and our method, respectively, with our approach extending the accessible regime between them.

We probe this regime experimentally by running QAOA for Max-Cut on IBM's square-lattice hardware for 80-node instances near the feasibility boundary, including both RR3 graphs (edge density ≈ 0.037) and ER graphs, see highlighted squares. Performance is evaluated using the mean approximation ratio, defined as the ratio between the average cost over quantum samples and the optimal classical cost.

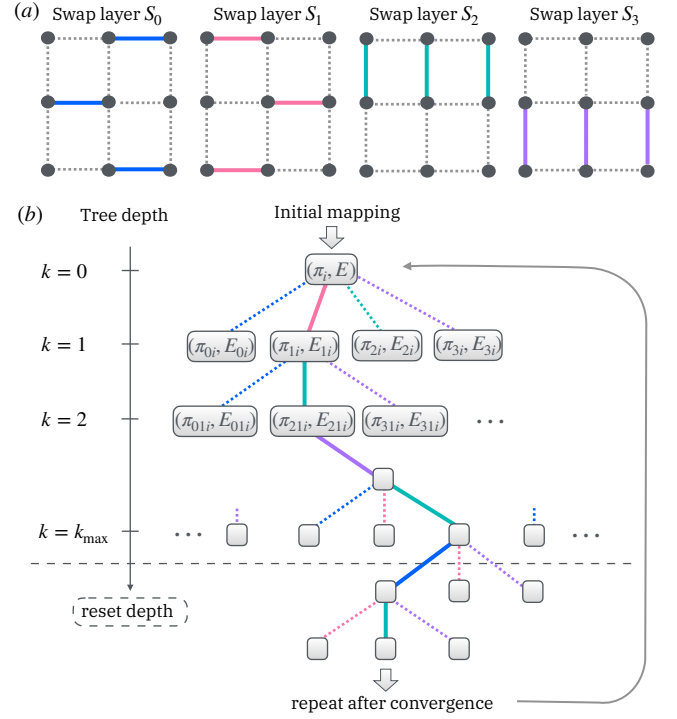


Figure 1: (a) The square ‘basis’ set of 2D SWAP layers for a 3×3 square lattice, introduced in [22]. (b) Conceptual overview of the problem-dependent circuit transpilation algorithm. A tree search over SWAP layer sequences is performed, where branches correspond to ‘basis’ SWAP layers and nodes encode qubit layouts π and remaining interactions E . For example, at depth $k=1$, the possible qubit orderings are denoted by $\pi_{0i}, \pi_{1i}, \pi_{2i}, \pi_{3i}$, where the subscripts indicate the respective SWAP sequence. The tree is explored up to a fixed depth $k_{\max}$, after which the leaf minimizing the number of remaining edges is selected and used to continue the search until all interactions are implemented, yielding a SWAP layer sequence. The outcome can be further refined via re-optimization of the initial qubit mapping and repeating this process.

In noiseless MPS simulations, increasing the number of QAOA layers p improves performance, as expected, see solid lines in Fig. 3(a). On hardware, however, this trend can reverse for larger systems due to more noise in deeper circuits, see dashed and dotted lines for ≥ 48 qubits. Despite this, our greedy method consistently outperforms the standard linear SWAP strategy across all tested system sizes. This advantage is further illustrated by the cumulative distribution functions (CDFs) of sampled cost values, Fig. 3(b). Since higher cost values are desirable, a right-shifted CDF indicates better performance. Our method yields distributions that are consistently shifted toward higher values compared to the linear strategy.

Finally, we quantify the improvement as the relative increase in approximation ratio over the linear method, Fig. 3(c). This improvement grows with circuit complex-

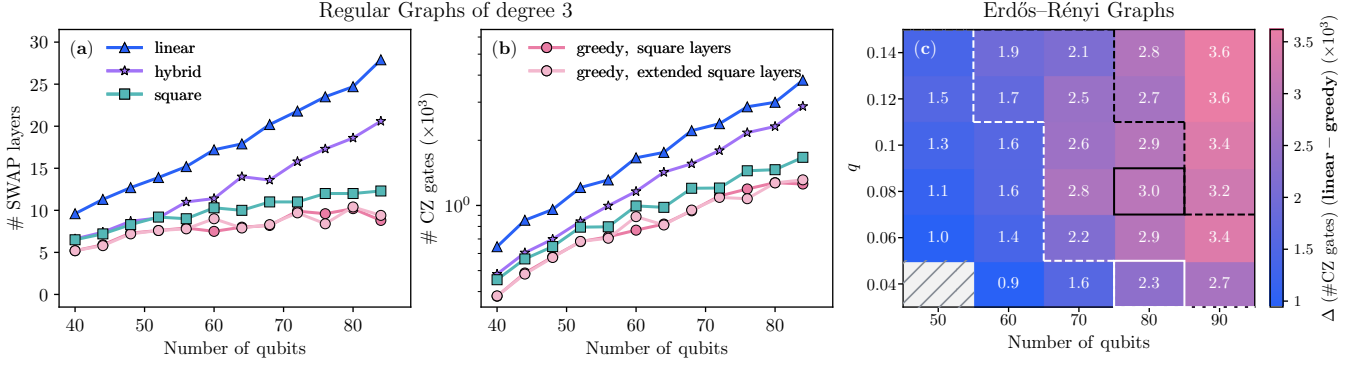


Figure 2: Gate scaling versus graph size and edge density for time-evolution circuits on a square lattice under different SWAP strategies. The *linear* strategy is the current state of the art; the *hybrid* and *square* strategies (the latter from [22]) are discussed in the full version text. The performance of the *greedy* strategy depends on the choice of SWAP-layer basis (pink curves). In (c), dashed lines indicate feasibility thresholds ($\sim 3,600$ CZ gates) for the linear (white) and greedy (black) strategies; the region between them marks instances enabled by our approach. The black square highlights the hardware experiment in Fig. 3. Data points are averaged over 10 instances.

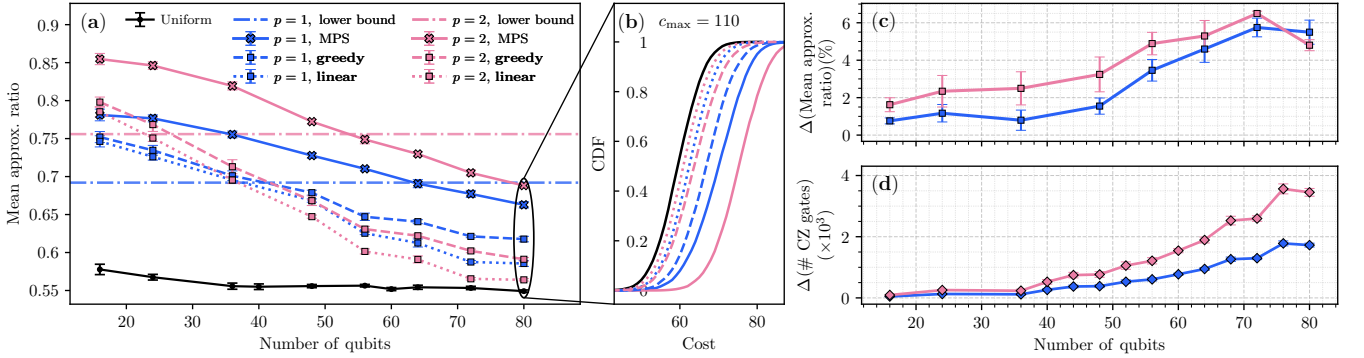


Figure 3: Hardware performance analysis of QAOA circuits for Max-Cut with $p = 1$ (blue) and $p = 2$ (pink), transpiled using *linear* (dashed) and *greedy* (dotted) strategies. Lower bounds in (a) are theoretical guarantees [30, 31] and become difficult to resolve with our approximate MPS simulator at larger system sizes. Results are averaged over 10 graph instances, with 10,000 samples per circuit. No error mitigation or post-selection is applied.

ity, as measured by system size and CZ gate count, reflecting the increasing number of gates avoided by our method, see Fig. 3(d).

V. Conclusions and Discussion

We introduce a transpilation method for circuits of commuting two-qubit gates, common in quantum algorithms for combinatorial optimization and Ising systems. Our method produces significantly shallower circuits than the current best practices by constructing problem-dependent sequences of SWAP layers, rather than relying on fixed routing patterns. By minimizing the number of SWAP layers required to realize all interactions, we effectively reduce circuit depth and the number of entangling gates. Although this comes at the cost of additional classical pre-processing, scalability is maintained by limiting the exploration depth, enabling improvements for systems with up to 90 qubits.

Numerically, we find that the reduction in the CZ gate count increases with graph complexity. Hardware experiments confirm that these reductions translate into improved performance, with gains of up to 7% in approximation ratios for both Max-Cut and maximum independent set (MIS) at depth-one and depth-two QAOA. Here, we only presented the results on Max-Cut, but in the presentation, we would also present results for MIS which show a similar behavior.

Future work may consider designing improved sets of SWAP layers. While our method relies on a predefined set, different choices can lead to varying performance, and identifying optimal or near-optimal sets remains an open problem. Extending this approach to other hardware topologies is another promising avenue. Overall, our results highlight the importance of problem-aware transpilation and suggest that such strategies are key to improve the performance of quantum algorithms on noisy devices.

-
- [1] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments* (Yale University Press, 1959).
- [2] B. Pascariu, M. Sama, P. Pellegrini, A. D'Ariano, J. Rodriguez, and D. Pacciarelli, Formulation of train routing selection problem for different real-time traffic management objectives, *J. Rail Transp. Plan. Manag.* **31**, 100460 (2024).
- [3] N. S. Blunt, J. Camps, O. Crawford, R. Izsák, S. Leontica, A. Mirani, A. E. Moylett, S. A. Scivier, C. Sunderhauf, P. Schopf, *et al.*, Perspective on the Current State-of-the-Art of Quantum Computing for Drug Discovery Applications, *J. Chem. Theory Comput.* **18**, 7001–7023 (2022).
- [4] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems* (Springer, 2016).
- [5] N. Bansal *et al.*, Resource Allocation in Cloud Computing, *IEEE Transactions on Parallel and Distributed Systems* **23**, 1417 (2012).
- [6] Y. Cao, J. Romero, and A. Aspuru-Guzik, Potential of quantum computing for drug discovery, *IBM J. Res. Dev.* **62**, 6:1–6:20 (2018).
- [7] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications* (SIAM, 2014).
- [8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, 1979).
- [9] R. M. Karp, On the Computational Complexity of Combinatorial Problems, *Networks* **5**, 45–68 (1975).
- [10] IBM ILOG CPLEX. V20.1: User's manual for cplex, <https://www.ibm.com/docs/en/icos/22.1.2?topic=optimizers-users-manual-cplex>.
- [11] Gurobi Optimization LLC., <https://www.gurobi.com>.
- [12] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms* (Cambridge University Press, 2011).
- [13] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Dover, 1998).
- [14] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, 1988).
- [15] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency* (Springer, 2003).
- [16] E. Farhi, J. Goldstone, and S. Gutmann, A Quantum Approximate Optimization Algorithm (2014) [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [17] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* **2**, 5 (2014).
- [18] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
- [19] H. Zou, M. Treinish, K. Hartman, A. Ivrii, and J. Lishman, LightSABRE: A Lightweight and Enhanced SABRE Algorithm (2024) [arXiv:2409.08368](https://arxiv.org/abs/2409.08368).
- [20] G. Li, Y. Ding, and Y. Xie, Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19 (Association for Computing Machinery, New York, NY, USA, 2019) p. 1001–1014.
- [21] D. Maslov, S. M. Falconer, and M. Mosca, Quantum Circuit Placement, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **27**, 752 (2008).
- [22] J. Weidenfeller, L. C. Valor, J. Gacon, C. Tornow, L. Bello, S. Woerner, and D. J. Egger, Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware, *Quantum* **6**, 870 (2022).
- [23] A. Matsuo, S. Yamashita, and D. J. Egger, A SAT Approach to the Initial Mapping Problem in SWAP Gate Insertion for Commuting Gates, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **106**, 1424 (2023).
- [24] T. Koch, D. E. Bernal Neira, Y. Chen, G. Cortiana, D. J. Egger, R. Heese, N. N. Hegade, A. G. Cadavid, R. Huang, T. Itoko, *et al.*, Quantum Optimization Benchmark Library – The Intractable Decathlon (2025) [arXiv:2504.03832](https://arxiv.org/abs/2504.03832).
- [25] 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, *Phys. Rev. X* **10**, 021067 (2020).
- [26] S. Brandhofer, D. Braun, V. Dehn, G. Hellstern, M. Hüls, Y. Ji, I. Polian, A. S. Bhatia, and T. Wellens, Benchmarking the performance of portfolio optimization with QAOA, *Quantum Inf. Process.* **22**, 25 (2022).
- [27] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, Benchmarking the quantum approximate optimization algorithm, *Quantum Inf. Process.* **19**, 1–24 (2020).
- [28] F. G. Fuchs, K. O. Lye, H. M. Nilsen, A. J. Stasik, and G. Sartor, Constraint Preserving Mixers for the Quantum Approximate Optimization Algorithm, *Algorithms* **15**, 202 (2022).
- [29] Z. He, R. Shaydulin, S. Chakrabarti, D. Herman, C. Li, Y. Sun, and M. Pistoia, Alignment between initial state and mixer improves QAOA performance for constrained optimization, *Npj Quantum Inf.* **9**, 121 (2023).
- [30] J. Wurtz and D. Lykov, Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs, *Physical Review A* **104**, 052419 (2021).
- [31] 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 (2021) [2110.14206](https://arxiv.org/abs/2110.14206).