Hybrid quantum optimization in the context of minimizing traffic congestion

Jedwin Villanueva,[1, *] Gary J Mooney,[1, †] Bhaskar Roy Bardhan,[2]
Joydip Ghosh,[2] Charles D Hill,[1, 3] and Lloyd C L Hollenberg[1, ‡]

[1]*School of Physics, University of Melbourne, Parkville, VIC, 3010, Australia.*
[2]*Research and Advanced Engineering, Ford Motor Company, Dearborn, MI 48124, USA.*
[3]*School of Mathematics and Statistics, University of Melbourne, VIC, Parkville, 3010, Australia.*
(Dated: April 14, 2025)

Traffic optimization on roads is a highly complex problem, with one important aspect being minimization of traffic congestion. By mapping to an Ising formulation of the traffic congestion problem, we benchmark solutions obtained from the Quantum Approximate optimization Algorithm (QAOA), a hybrid quantum-classical algorithm. In principle, as the number of QAOA layers approaches infinity, the solutions should reach optimality. On the other hand, short-depth QAOA circuits are known to have limited performance. We show that using tailored initialization techniques encourages the convergence to the desired solution state at lower circuit depths with two and three QAOA layers, thus highlighting the importance of adapting quantum algorithms in the noisy intermediate scale (NISQ) quantum computing era. Moreover, for NISQ devices with limited qubit connectivity and circuit depth, we introduce a heuristic noise-resilient variant of QAOA predicated on the elimination of long-range 2-qubit interactions in the QAOA layers whilst the cost function is unaltered. Our results show that this QAOA variant is surprisingly effective, outperforming QAOA on a physical IBM Quantum computer device.

I. INTRODUCTION

As quantum computer technology continues to traverse the noisy intermediate scale quantum (NISQ) regime towards full-scale error-corrected systems [1–3], the implementation of quantum algorithms for real-world applications such as optimization has received much attention [4–6]. The benchmark approach is the quantum approximate optimization algorithm (QAOA) [7] typically applied to problems encoded in a quadratic unconstrained binary optimization (QUBO) formulation or equivalently an Ising formulation [8]. QAOA has also shown promise in addressing binary optimization problems with higher order terms [9]. It remains to be seen whether such generic quantum optimisation approaches in this nascent NISQ era can compete with highly developed classical heuristics that have enjoyed many decades of evolution, e.g. for the travelling salesman problem (TSP) [10–15] as well as more general QUBO problems [16]. Nevertheless, the implementation, benchmarking and improvement of quantum optimization approaches for real-world problems on NISQ devices is a topic of great interest [17–20]. Within these confines, a key question focuses on improving implementation of quantum optimization algorithms on realistic hardware in the NISQ era, potentially informing how such calculations might be better performed on full-scale error corrected systems. In this paper we investigate implementations of the QAOA approach in the context of an important real-world problem – minimization of traffic congestion. Through numerical simulation in the first instance, we study the effect of initialization and the use of precomputed parameters to avoid barren plateau issues in the classical minimization steps. The effect of noise on the solutions was studied via implementation on IBM Quantum devices, and we introduce a hardware-informed QAOA ansatz compression strategy to also avoid noise. Under different metrics that quantify the quality of solutions and runtime, we examine how QAOA solutions scale over a range of problems of up to 23 cars.

Since the proposal of QAOA, the performance and limitations of the algorithm have been intensively studied in general classes of problems which are typically in the form of MaxCut [21–26]. Generally, the quality of the QAOA solution scales with the circuit depth, along with the number of variational parameters. One of the main challenges come from the noise-free barren plateau (BP) problem [27] which ultimately makes the classical optimization of the QAOA circuits NP-Hard [28]. The noise-induced barren plateau (NIBP) is a similar BP phenomenon causing vanishing gradients in the presence of noise [29], further obscuring the global optimum during classical optimization. To alleviate the difficulty of parameter optimization, various initialization techniques have been proposed and investigated [30–32], including the use of nearly-optimal precomputed parameters [33–37]. To further reduce the classical burden, we investigate the algorithm's performance using precomputed parameters to omit the optimization of the QAOA algorithm. This approach largely avoids the BP and the NIBP problems. Other methods of omitting the optimization are possible for short-depth QAOA circuits via analytical means [38, 39], as well as for QAOA circuits with low connectivity via reverse causal cones which provide ideal measurements via classical simulation of reduced quantum circuits [40].

Despite rapidly improving quantum hardware [41–45], another key factor to the performance of QAOA on a quantum processor is the noise induced by large-depth circuits, restricting the size of reliable quantum computation. Even a single layer of QAOA on a general QUBO instance can result in densely connected QAOA circuits. Running such circuits requires the use of SWAP gates in order to realize the interactions of non-connected qubits on the hardware. This impacts heavily on the performance [46], prompting a surge of research aiming to improve or compress the QAOA ansatz [47–54] and implement advanced error mitigation techniques [3, 9]. Here, we introduce and investigate a heuristic for reducing noise on physical hardware by removing 2-qubit interaction gates that require SWAP gates. As this fundamentally changes the underlying variational ansatz, we re-optimized the compressed circuits with respect to the original problem Hamiltonian while introducing extra variational parameters.

This paper is organized as follows: in Section II, we introduce the traffic problem and its mapping to QUBO form. In Section III, we benchmark baseline QAOA, approaches to initialization, and the use of precomputed QAOA parameters. Results from running on hardware are given in Section IV. In Section V we implement connectivity-forced QAOA ansatz compression tailored to the hardware as well as various measures of run time for comparison analyses. Conclusions are given in Section VI.

II. BACKGROUND

A. Traffic Problem Statement

The problem of reducing congestion on roads can be modeled as a binary combinatorial optimization problem by defining decision variables corresponding to the route taken by each car, as previously done in [55]. Given a set of cars, we assigned each car a set of possible routes to travel from their origin to destination. A route is a path consisting of edges corresponding to road segments. Each route begins and ends respectively at origin and destination nodes, which were always chosen to be located at intersections for simplicity. An example problem is shown in Fig 1. The decision variables q_{ij} for car i taking route j can be assigned as,

$$q_{ij} = \begin{cases} 1, & \text{if car } i \text{ taking route } j \text{ is True} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

A cost is assigned to each segment, depending on which cars have routes that travel on that segment. Namely, for the k-th road segment in a road network, labelled s_k, the cost is the sum

$$c(s_k) = d_k \cdot \left(\sum_i \sum_{j \in R_i} q_{ij} \right)^2. \quad (2)$$

where $R_i = \{j|\ \text{car } i \text{ route } j \text{ contains } s_k\}$ is the set of routes for car i that contain segment s_k, with $d_k > 0$ a positive weight for the segment, such as the segment's physical length. The prefactor d_k can be set to 1 if all segments are equally weighted. If multiple cars travel on a segment s_k, then $c(s_k)$ will increase quadratically with the number of non-zero binary variables q_{ij}. This disincentivises multiple cars traveling on the segment. The total traffic congestion contribution to the cost over the network can be taken as,

$$A = \sum_k c(s_k). \quad (3)$$

minimizing this term corresponds to minimizing the congestion over the whole road network. However, this cost function is trivially minimized by setting every variable to zero. We thus enforce appropriate constraints stating that each car must take exactly one route. The constraint for each car takes the form

$$B_i = 1 - \sum_j q_{ij} = 0, \quad (4)$$

noting that the sum is over all routes j of car i, so that only one $q_{ij} = 1$ within the set of routes of a car. The constraints can be incorporated into the the cost function by minimizing $(B_i)^2 > 0$. The total constraint contribution to the cost function can be obtained by summing over all cars,

$$B = \sum_i (B_i)^2 > 0. \quad (5)$$

The traffic problem can now be formulated as a quadratic unconstrained binary optimization (QUBO) problem using the cost function,

$$C = A + \lambda B, \quad (6)$$

where the factor $\lambda > 0$ needs to be sufficiently large to penalize violation of constraints. As a heuristic choice, we set λ so that it scales the range of possible cost values of λB to be equal to the range of A. That is, λ is chosen such that

$$\max_{q_{ij}}(A) - \min_{q_{ij}}(A) = \max_{q_{ij}}(\lambda B) - \min_{q_{ij}}(\lambda B),$$

using the observation

$$\text{argmin}_{q_{ij}}(\lambda B) = \text{argmin}_{q_{ij}}(A) = \{0, 0, ...0\} \text{ and,}$$
$$\text{argmax}_{q_{ij}}(\lambda B) = \text{argmax}_{q_{ij}}(A) = \{1, 1, ...1\}.$$

We found that this method of choosing λ resulted in robust solutions that successfully satisfied the constraints in all of our experiments.

B. Mapping to QAOA

The first step in mapping the QUBO problem to the QAOA is to transform the cost function of binary vari-
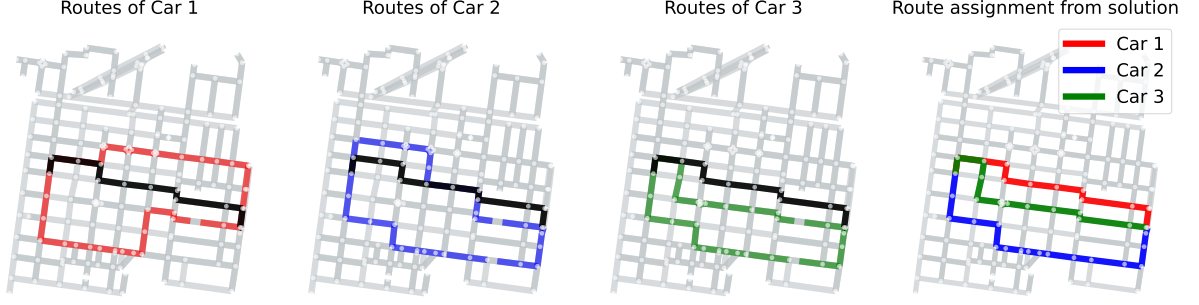
FIG. 1. A problem of 3 cars, 3 routes each (9 variables) showing the designated routes. The black route is a common route between all cars chosen to be the shortest route. The dashed lines show two dissimilar, alternative routes (detours). A solution to the QUBO is plotted in the rightmost panel, showing minimal traffic congestion. Provided the cars are traveling in the same direction, the problem can be interpreted as depicting either direction of travel.

ables to a quantum Ising Hamiltonian consisting of Pauli-Z operators Z_u with eigenvalues $z_u \in \{-1, 1\}$. The mapping is given by,

$$q_{ij} \mapsto \frac{1}{2}(1 - Z_u), \qquad (7)$$

where u represents a re-indexing of the pairs of indices (i, j), ordered in ascending order with j increasing first and then i. Under this mapping, the QUBO cost function is transformed to the Ising Hamiltonian

$$H_C = \sum_{uv} J_{uv} Z_u Z_v + \sum_u h_u Z_u + c, \qquad (8)$$

up to a scalar constant c. The Ising Hamiltonian can be visually represented as a graph, where edges are weighted according to J_{uv} and nodes are weighted according to h_u. The expectation values of the Ising Hamiltonian with respect to a superposition of quantum states give the average cost of those states weighted by the state probabilities. By minimizing the expectation value with respect to a variational ansatz, it is possible approximate the ground state of H_C, resulting in a quantum state that maximises the probability of the ground state. The QAOA ansatz is defined as,

$$|\Psi(\vec{\gamma}, \vec{\beta})\rangle = \prod_{l=1}^{p} e^{-i\beta_l H_B} e^{-i\gamma_l H_C} |+\rangle^{\otimes N}, \qquad (9)$$

where $\vec{\gamma} = (\gamma_1, \gamma_2, ...)$ and $\vec{\beta} = (\beta_1, \beta_2, ...)$ are the variational parameters, and N is the total number of qubits. $H_B = -\sum_u X_u$ is a mixer Hamiltonian consisting of Pauli-X operators summed for each qubit u for which the initial state $|+\rangle^{\otimes N}$ is the ground state of H_B. The two unitary operators in each of the p factors in the product are time-evolution operators of H_C and H_B. The initial state $|+\rangle^{\otimes N}$ can be prepared on a quantum computer as

$$|+\rangle^{\otimes N} = H^{\otimes N} |0\rangle^{\otimes N}. \qquad (10)$$

The variational method is applied by minimizing the expectation value of the Ising Hamiltonian

$$\langle H_c \rangle = \langle \Psi(\vec{\gamma}, \vec{\beta})| H_c |\Psi(\vec{\gamma}, \vec{\beta})\rangle \qquad (11)$$

with respect to the variational parameters $\vec{\gamma}$ and $\vec{\beta}$. A representative quantum circuit of the QAOA ansatz is shown in Fig 2. The Ising Hamiltonian can be normalised by some non-zero positive constant which is equivalent to a re-scaling of the parameter vector $\vec{\gamma}$. We perform normalization such that the average absolute values of coefficients, J_{uv} and h_u is equal to 1. This was done so that $-\pi \leq \gamma_i \leq \pi$ is a suitable search space, since values outside this range are likely to be too large and structure from the problem Hamiltonian encoded in the circuit is dominated. Without rescaling, the parameter landscape exhibits the narrow basin of attraction in [35], where optimization features are narrow and is thus difficult for traditional optimizers. We also use the translational symmetry $\beta_i \mapsto \beta_i \pm \pi$ which leaves the ansatz unchanged in order to restrict $-\pi/2 \leq \beta_i < \pi/2$.

III. QAOA INITIALISATION

The performance of QAOA is dependent on the variational parameters that define the final QAOA circuit. However, finding the optimal QAOA parameters can be computationally difficult, with studies showing that the classical search for optimal parameters is in fact NP-Hard [28] via gradient descent methods with randomly initialized QAOA parameters. We implemented various initialization techniques from literature, and benchmarked their performance on the traffic problems against random initialization.

Some of the first initialization techniques that were proposed involve incrementally adding a QAOA layer, using the optimal parameters found from the p-layer QAOA to give an initial guess to the $(p+1)$-layer QAOA. This means that at each step, the number of variational parameters increases by 2. We considered two

methods based on this approach called INTERP and FOURIER [30]. INTERP uses the observation that the optimal p-layer QAOA parameters tend to follow the same scheduling as the optimal $(p+1)$-layer QAOA, thus an initial guess to the $(p+1)$-layer QAOA is a linearly interpolation from the optimal p-layer QAOA parameters. FOURIER is a re-parametrisation of the variational parameters via a discrete sine/cosine transformation,

$$\gamma_i = \sum_{k=1}^{q} u_k \sin\left(\left[k - \frac{1}{2}\right]\left[i - \frac{1}{2}\right]\frac{\pi}{p}\right), \qquad (12)$$

$$\beta_i = \sum_{k=1}^{q} v_k \sin\left(\left[k - \frac{1}{2}\right]\left[i - \frac{1}{2}\right]\frac{\pi}{p}\right), \qquad (13)$$

optimizing over the FOURIER parameters $\vec{u}$ and $\vec{v}$ instead. The $(p+1)$-layer QAOA parameters is initialized by using the same FOURIER parameters of the optimal p-layer QAOA. optimization is also performed with respect to the FOURIER parameters. Another promising initialization technique we considered is Trotterised Quantum Annealing (TQA) [31], utilising the connection between QAOA and quantum annealing. In particular, the continuous time evolution of a linear-ramp quantum annealing schedule is discretised via Trotterisation, resulting in the alternating unitary operators of QAOA and a formula for the QAOA parameters given by,

$$\gamma_j = \frac{j}{p}\Delta t, \ \beta_j = (1 - \frac{j}{p})\Delta t, \qquad (14)$$

which involves the Trotter time step Δt as a parameter. The Trotter time step is related to the total quantum annealing time T and the number of QAOA layers p, as $T = p\Delta t$. The optimal value for Δt is not known and varies between different QAOA instances. Similar to the presence of QAOA parameter concentration - which has been observed on certain problem classes as instance sizes increase [56, 57] - it was observed that the best values for Δt also concentrate [31]. In particular, an optimal concentration value of $\Delta t \sim 0.75$ was found for typical weighted and unweighted MaxCut QAOA instances.

A. Performance Metric: Approximation Measure

We want a way of measuring the quality of the QAOA quantum states $|\psi\rangle = |\Psi(\vec{\gamma}, \vec{\beta})\rangle$, as well as a single computational basis state $|\psi\rangle = |z\rangle$. We define the instance specific approximation measure

$$R_{\text{true}} := \frac{E_{\max} - \langle H_C\rangle}{E_{\max} - E_{\min}}, \qquad (15)$$

where $E_{\max}$ is the maximum energy eigenvalue of H_C, $E_{\min}$ is the minimum energy eigenvalue, and $\langle H_C\rangle = \langle\psi|H_C|\psi\rangle$ is the expected energy with respect to the target quantum state. The lowest energy state corresponds to $R_{\text{true}} = 1$, and the highest energy state corresponds to

$R_{\text{true}} = 0$. For larger problems, $E_{\max}$ and $E_{\min}$ is classically difficult to calculate. To reduce computational demands, we use a metric that substitutes $E_{\min}$ with a close approximation of the ground state energy and $E_{\max}$ with the mean energy derived from a uniform distribution of states. We define this as the approximation measure *relative to random*,

$$R_{\text{random}} := \frac{E_{\text{random}} - \langle H_C\rangle}{E_{\text{random}} - \overline{E}_{\min}} = \frac{c - \langle H_C\rangle}{c - \overline{E}_{\min}}. \qquad (16)$$

where E_{random} is the average energy of the 2^N possible states. To obtain the simplified equation on the right of Eq (16), we note that E_{random} can be calculated as,

$$E_{\text{random}} = \langle +^N|H_C|+^N\rangle, \text{ where } |+^N\rangle := |+\rangle^{\otimes N}, \qquad (17)$$

and since $\langle +^N|Z_i|+^N\rangle = 0$ and $\langle +^N|Z_iZ_j|+^N\rangle = 0$, the contributions of the Pauli terms of H_C vanish, leaving only the scalar constant c in Eq (8). To approximate $\overline{E}_{\min}$, we use a state-of-the-art commercial solver, Gurobi [58], which is highly likely to be optimal for the small problem instances in this work. In summary, a positive value $R_{\text{random}} > 0$ represents an expected energy better than randomly guessing, and a value close to 1 indicates near-optimality. While the initial measure $R_{\text{true}} > 0$ for any possible quantum state, the adjustment to R_{true} means that this new measure can be negative for some states for which its average performance is worse than randomly guessing.

B. Numerical Experiment Setup

In the following, traffic problems of various numbers of cars and routes were randomly generated using the Open Street Map Python package called OSMnx [59] which was used to obtain road distances of a region within the city of Melbourne, Australia. For a given number of cars and number of routes, we fixed the starting and destination nodes which were the road intersections and used a k-shortest paths algorithm to generate 1000 routes for each instance. For each car we then randomly selected routes according to a uniform distribution, in addition to the shortest possible path. Once the required number of routes were selected for each car, the traffic instance was formulated as a QUBO problem and mapped to QAOA. The QAOA circuits were then simulated without noise using Qiskit [60], an open-source quantum framework. The following shows an outline of the preparation of the traffic instances and QAOA solutions, as schematically shown in Fig 2.

1. *QUBO Formulation.* Prepare multiple routes from an origin node to a destination node. Randomly assign to each car a number of routes (see Fig 1 for an example). Define the QUBO objective function to minimize.

2. *Convert to Ising Formulation.* Use the mapping described in Eq(7) to obtain the Ising Hamiltonian in Eq(8).

3. *Construct p-layer QAOA Ansatz.* Construct the variational state in Eq(9) on a quantum computer. The final QAOA circuit is then obtained by the following two steps.

   A) *Initialize $2p$ QAOA parameters.* initialize QAOA parameters for the QAOA circuit optimization. We used random initialization and three existing techniques found in the literature on variational quantum algorithms.

   B) *Classical optimization.* Use a classical optimizer to find the optimal QAOA parameters such that the ansatz minimizes the expected energy of the Ising Hamiltonian. We used an off-the-shelf gradient descent optimizer, SciPy's L-BFGS-B, however most optimizers should be sufficient.

4. *Measure the QAOA State.* Repeatedly run and measure the optimized QAOA circuit on a quantum computer, obtaining a distribution of binary strings. There are two choices when taking a single solution state from the distribution. Under the assumption that the QAOA state prepares the ground state with high probability, the first option is to take the most occuring string in the distribution. The second option is to evaluate the cost of each binary string and choose the lowest energy string at the cost of extra classical processing. We consider both solutions in Section V.

We also investigated the performance of using precomputed initial parameters without the classical optimization in Step 3B. As a practical consideration, this avoids the BP and NIBP problems, and reduces the number of quantum circuit executions required. Methods used to calculate optimal precomputed parameters are generally computationally intensive, however this overhead can be considered worthwhile. Once the parameters are optimized, they can frequently be applied to multiple graphs with similar properties [34]. Additionally, methods exist to adapt these parameters to achieve near-optimal results for graphs with varied structure [35, 36].

In order to generate precomputed parameters for our traffic problems, first we generated traffic problems of varying sizes. We then measure the mean and standard deviations of the Ising coefficients resulting from those traffic problems. We then randomly generated 100 Ising Hamiltonians of 9-qubits with coefficients J_{uv} and h_u chosen from a normal distribution with the same mean and standard deviation calculated from the generated traffic problems. Since the traffic problems typically result in fully connected Ising Hamiltonian graphs attributing to the constraint terms, we also used completely connected Ising Hamiltonians for this step. We then perform

and optimize the QAOA circuits, and used the median of those 100 optimized parameters as the precomputed parameters of all the traffic problems in our numerical experiments.

C. Numerical Results: QAOA Benchmark

1. RANDOM and TQA Initialization

The importance of initialization techniques is investigated by benchmarking TQA initialization against a naive technique of randomized (RANDOM) initialization, where QAOA parameters were drawn at random from a uniform distribution. The performance of each initialization technique was benchmarked for 9-qubit instances of the traffic congestion problem with 3 cars and 3 routes, with $p = 2$ and $p = 3$ layers of QAOA. We initialized 50 QAOA parameters for each approach, totaling 100 initializations. For RANDOM, the QAOA parameters were drawn from the ranges $\gamma_i \in [-\pi, \pi)$ and $\beta_i \in [-\pi/2, \pi/2)$. For TQA, the Trotter time steps Δt were 50 equally spaced values in the range $\Delta t \in [0.1, 1)$. The minimum value of $\Delta t = 0.1$ avoids initializing values which are too small which is equivalent to setting $\gamma_i = 0, \beta_i = 0$ in the optimization. Each RANDOM and TQA initialized QAOA parameters are then optimized via Scipy's L-BFGS-B optimizer with a fixed maximum number of 150 iterations, which was sufficient for convergence for both initialization approaches.

We plot the averages over 50 initialized parameters over all 20 traffic instances in Fig 4. The shaded regions show the range between minimum and maximum average convergence, and the solid line corresponds to the median traffic instance. Across the 20 traffic instances, it is clear that TQA initialization outperforms random initialization on average. In fact, this suggests that randomly guessing Δt is better than randomly guessing the parameters. We also plot the convergence of each parameter for a single traffic instance in Fig 3, where each run is colored to distinguish between the 50 different initialized parameters. For this instance, TQA had more parameters which converged to a higher approximation measure. By observing the averaged approximation measures over the 50 initialized parameters, the TQA initialization method shows a distinct advantage in solution quality over RANDOM. The difference is more significant with $p = 3$ layers of QAOA.

2. Performance of Initialization Strategies

We observed that initializing parameters with tailored techniques performs well even without optimization. This is particularly useful for precomputed parameters, which may be obtainable through noiseless simulations by looking at smaller instances or sub-graphs of ensembles of traffic instances, justified from parameter
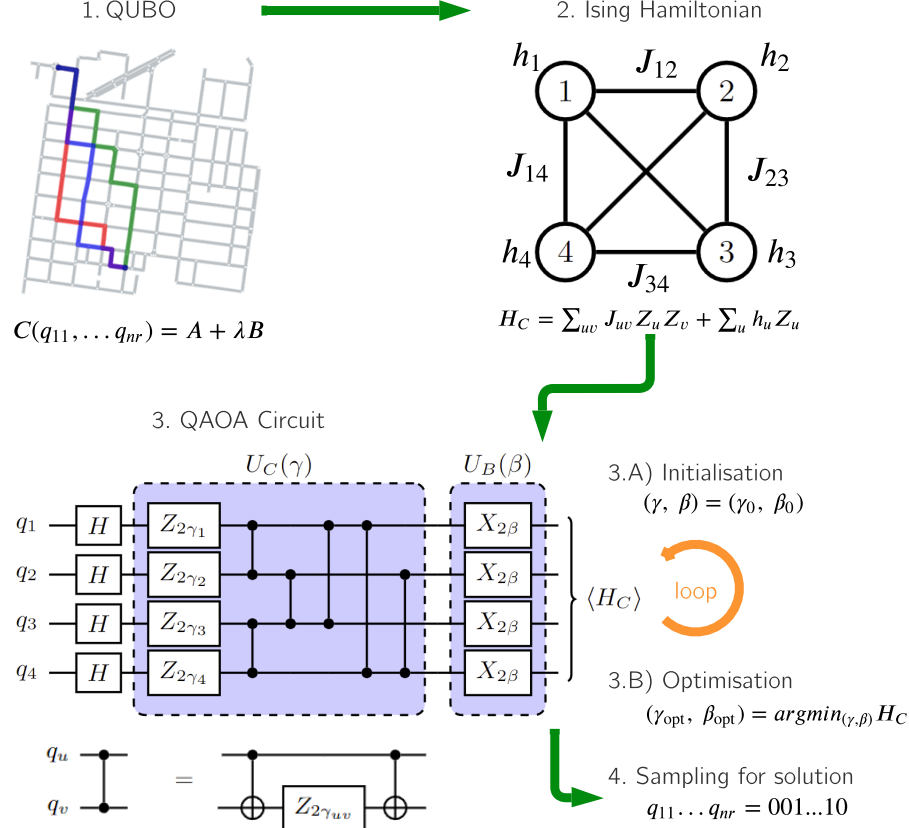
FIG. 2. Schematic of the QAOA algorithm for the traffic congestion problem. 1) An instance of the traffic problem is formulated as a QUBO problem. 2) The QUBO is mapped to a quantum Ising Hamiltonian by mapping binary variables to Pauli-Z operators, such that the problem solution is encoded as the ground state. 3) A parameterized QAOA circuit is constructed using the Ising Hamiltonian, consisting of layers of alternating unitaries $U_C(\gamma)$ and $U_B(\beta)$. The decomposition of the ZZ-rotation gates is shown below. The angles for the single-qubit gates are shown as subscripts where $\gamma_u := \gamma h_u$ and $\gamma_{uv} = \gamma J_{uv}$. 3) The QAOA classical optimization loop occurs by repeatedly measuring the energy and classically updating the parameters, represented by the circular arrow. 3A) The parameters are initialized. 3B) The classical optimization loop updates the parameters (γ, β) until they converge $(\gamma_{opt}, \beta_{opt})$, minimizing the expectation of the Hamiltonian. 4) The optimized QAOA state is sampled from the quantum device to find an approximate solution to the original problem.

concentration in MaxCut instances independent of problem size [56, 57].

We benchmarked the performance of initial parameters against the optimal parameters obtained after optimization using $p = 2$ QAOA. For INTERP and FOURIER, the optimal parameters for $p = 1$ were initially calculated via a global grid search, followed by gradient descent to further refine the optimal parameters. Then, the $p = 2$ parameters for INTERP and FOURIER were calculated from the optimal $p = 1$ parameters. Additionally, we performed gradient descent from the precomputed parameters to obtain the optimal parameters for $p = 2$ QAOA. For each of the parameters, we took the probability distributions of the QAOA states to construct a density plot showing the approximation measure vs probability of a single state. In an ideal scenario where $r_q = 1$ for the QAOA state, this would correspond to a 100% probability of measuring the ground state, and hence a singular

point at $(Prob = 1, r_q = 1)$. However, this is generally not the case, especially for low-depth QAOA circuits. Instead better performance is indicated by a greater probability of measuring states with higher approximation measures, which would correspond to higher concentrations of data-points on the top-right of the plots.

The density plots of the 20 9-qubit traffic instances are plotted in Fig 5. For each instance, the optimal QAOA state always had the lowest average energy even though the highest probability state is generally not the ground state. In the plots, this corresponds to the states with $y = 0.8 \sim 0.9$ approximation measures having the highest probabilities $x \geq 0.03$. For some instances, there are states with approximation measures as low as 0.6 with high probability. Interestingly, FOURIER is most similar to the optimized state, but the reason for this is not explored due to the practical consideration that FOURIER still requires classical optimization of the $p-1$-
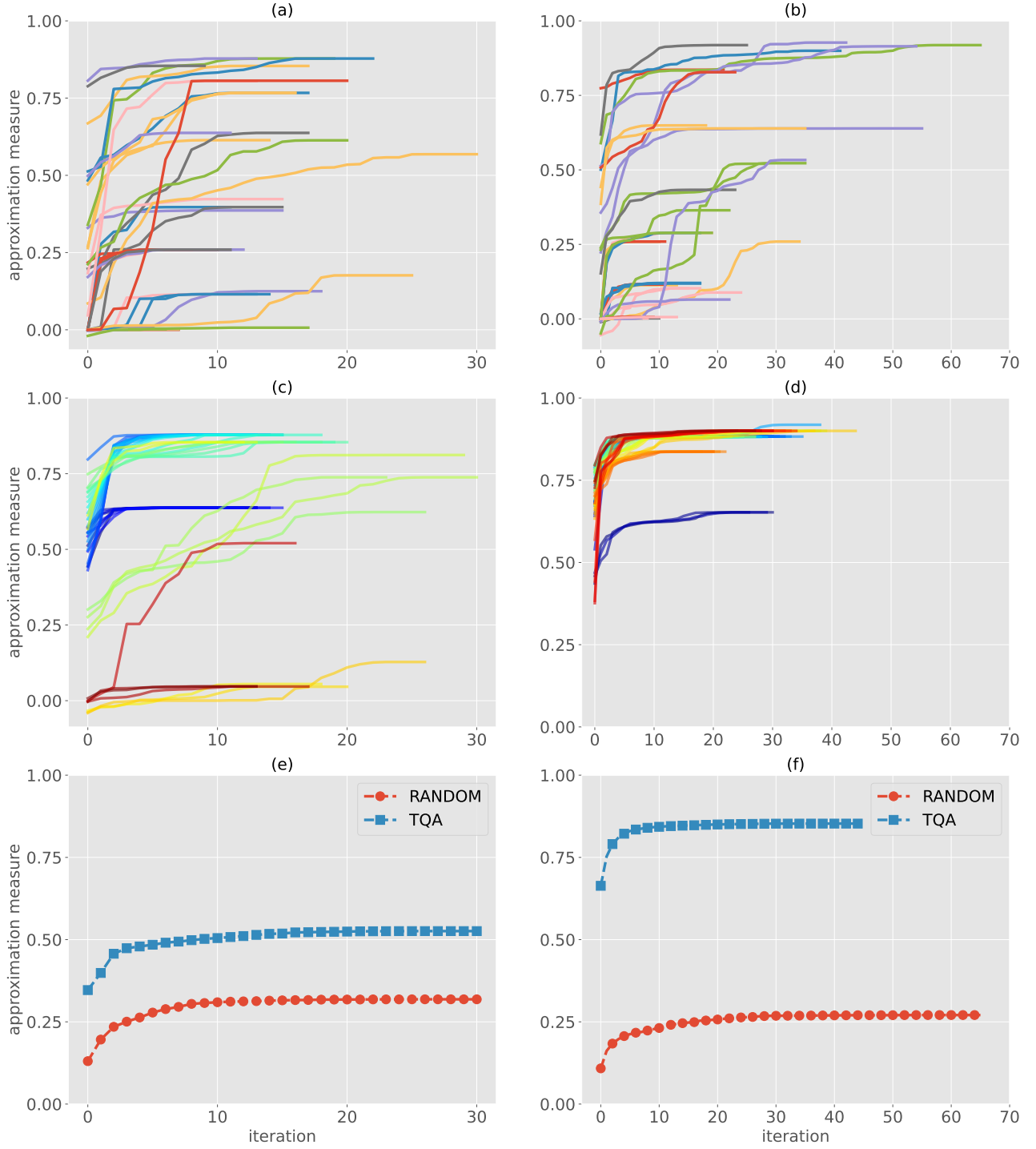
FIG. 3. Plots for a single traffic problem instance that show the convergence of the approximation measure for the evolving QAOA states during the optimization process using 50 QAOA parameters from (a)(b) RANDOM and (c)(d) TQA initialization techniques, and (e)(f) comparisons in performance between the two initialization techniques calculated as the average approximation measure over the corresponding 50 initializations. The line colors distinguish between different initializations. The left and right columns indicate values for $p = 2$ and $p = 3$ QAOA layers respectively.

layer QAOA while precomputed parameter initialization does not. In contrast, precomputed parameters seem to have the most reliable performance between all instances, as it tends to have the ground state with highest probability than other states. INTERP is the least performing method, suggesting that it is likely not useful for directly sampling the QAOA state without optimization.

IV. NISQ HARDWARE RESULTS

A. Performance of QAOA on NISQ hardware

Here we investigated the expected QAOA runtime scaling and naively compare with runtime scaling of the Gurobi classical solver, which use highly complex mathematical models to solve such integer problems [58]. Knowing the challenges of noisy optimisation due to NIBPs, we performed noiseless optimization of the QAOA circuits via simulations. The pre-optimised QAOA circuits are then executed on real quantum processors from IBMQ. This limits the destructive impact of noise observed in our numerical experiments to the final QAOA circuits only. We also observed that in most cases, precomputed parameters perform almost as well as optimized parameters on noisy quantum computers, since the prepared state is likely close to a mixed state. Thus, using pre-optimised parameters from noiseless simulations is valid in this noisy regime under the assumption of the availability of precomputed parameters, which is heavily discussed in previous sections. Additionally, to minimize circuit depths, we only perform QAOA with $p = 1$ layers. In order to see how performance scales with increasing problem size, we incremented the size of the traffic problems by adding a car with two possible routes each, beginning from a trivial case of 1 car with 2 routes. Each increment increases the qubit requirement by 2.

We first present the results of QAOA on simulators vs real devices to show how impact of noise on performance. In Fig 6, we plot the approximation measure on the left, as well as the probability of the ground state on the right, against the number of cars. For the approximation measure, we use two values, the average approximation measure over the whole QAOA distribution, labeled *avg*, and the maximal approximation measure of any one state in the distribution, labeled *best*, corresponding to the single best performing state observed in the QAOA distribution. The numerical experiment was performed on both ibmq_qasm_simulator (simulation) and ibmq_kolkata (hardware).

In the noiseless case, we see that the approximation measure of QAOA is ∼1 for the best solution state, and ∼0.7 for the average over the distribution. Unsurprisingly, the performance of the QAOA state significantly deteriorated on the noisy hardware. This is likely attributed to the increased circuit depths and CNOT counts of the densely connected QAOA circuits when transpiled to the physical IBM Quantum's ibmq_kolkata

with a heavy hex architecture [60](Fig. 4). The average approximation measure plateaus at some positive value, suggesting that even as the problem size increases, the QAOA ditribution will output a solution better than random guessing on average. Furthermore, the best solution state sampled from the noisy QAOA circuit reports a value no smaller than 0.8. However, when looking at the observed probability of the ground state, we see an exponential decrease in the noiseless case, but it is not as fast as the exponential increase of the search space dimension 2^N. Note that for problems with more than 11 cars (22 qubits), there were already instances where the ground state is not measured from the 10000 shots, suggesting more shots are required to measure the ground state. In the presence of noise, the rate of decrease in probability is much worse, first approaching the exponential rate of the search space dimension before dropping to a zero probability for instances with more than 10 qubits (5 cars). This means that the output solution of such an algorithm will almost always never be the exact solution for some large enough problem size with only a fixed number of shots. This is in contrast to classical algorithms, namely Goemans-Williamson with polynomial time scaling and a guarantee of *minimum* performance. Instead, the noisy QAOA(p=1) state has a limit on the *maximum* performance of the output state.

V. CONNECTIVITY FORCED QAOA ANSATZ

Since the previous section shows the destructive impact of noise, we heuristically counteracted noise in the simplest way possible by removing all RZZ gates that will induce SWAP gates according to the connectivity of the hardware, as in Fig 7. On its own, such a method is ill-defined since there are many ways to map the problem variables to the physical qubits on the device, with each different mapping resulting in different RZZ gates on the hardware. Nevertheless, we will show that such a modification to circuits will result in better performing solutions under the restrictions of noise and limited connectivity on the hardware. The circuits with excised gates are referred to as connectivity-forced QAOA (CF-QAOA) circuits. The removal of the RZZ gates are done to entirely remove the necessity of SWAP gates, which is expected to destroy most quantum properties of the QAOA circuit. The CF-QAOA circuit itself is a QAOA circuit containing layers of RZ, RZZ, and RX gates in alternating fashion however it will correspond to a different underlying Ising Hamiltonian with less ZZ terms. Nevertheless, we treat CF-QAOA as a variational ansatz with respect to the original problem Hamiltonian and as such, optimisation is performed by minimising the expectation including all the ZZ terms from the original problem.

From the variational principle, removing any gate from an optimal QAOA circuit will result in a higher energy state, since this will change the prepared state. To counteract this increase in energy, we introduce extra vari-
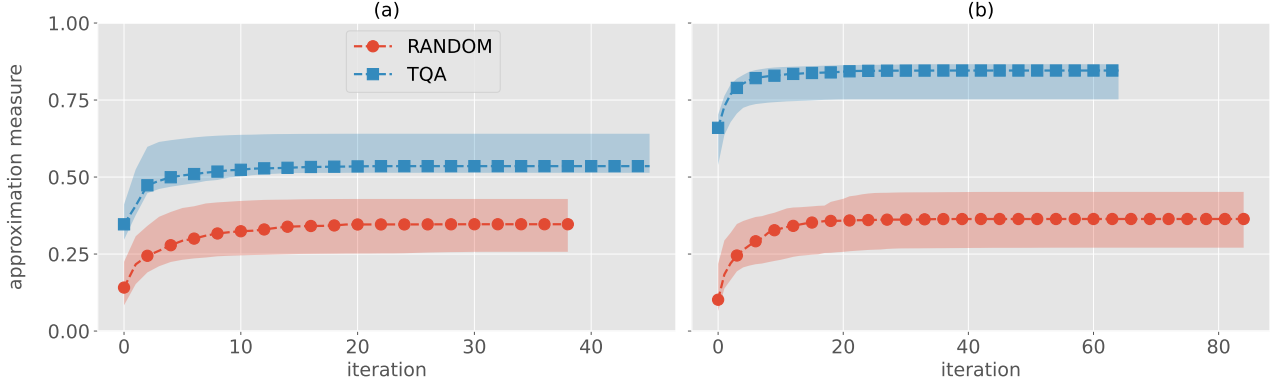
FIG. 4. Plots of comparisons of the RANDOM and TQA initialization techniques for (a) $p = 2$ and (b) $p = 3$ QAOA layers, taking the average approximation measure over the corresponding 50 initializations for each problem instance. The shaded region shows the range of recorded values for each technique.

ational freedom in the circuit without introducing additional gates by assigning the remaining rotation gates their own variational parameters in the CF-QAOA circuit in the same way that the multi-angle QAOA (ma-QAOA) [54] does for QAOA. We refer to these circuits as *CF-maQAOA*. Additional variational parameters could mean that the optimization landscape is susceptible to more local minima. More sophisticated analysis would be required to precisely determine the classical burden added by the additional variational parameters. For QAOA and CF-QAOA, the number of parameters is always $2p$, independent of the problem size. However, in ma-QAOA, the number of parameters also depends on the problem size per layer, increasing up to $O(p(N^2+N))$ corresponding to the $O(N^2)$ maximal scaling of the number of terms in an Ising Hamiltonian. However, for CF-maQAOA, the number of parameters in each layer is bounded by the qubit connectivity. In the case of the heavy hex architecture, CF-maQAOA scales only linearly in the problem size, scaling as $O(p(2.5N + N))$ assuming an average connectivity of $d = 2.5$. We used the optimized QAOA parameters to initialize both CF-QAOA and CF-maQAOA circuits.

A. Circuit Depths

As in Section IV, we performed optimization without noise and compared the performance of QAOA, CF-QAOA and CF-maQAOA running both on noiseless simulators and real hardware. We also performed readout error mitigation on the noisy circuits using Mthree [61]. We first report the circuit depths and CNOT counts in the QAOA circuits after transpiling to the heavy hex architecture, shown in Fig 8(a)(b). Both CF-QAOA and CF-maQAOA reach a constant depth in contrast to a continually increasing circuit depth for QAOA. The rate of increase for the number of CNOTs is also much slower for both the CF-QAOA and CF-maQAOA.

B. Performance of Modified Circuits

Next, we report our numerical and empirical results for the average approximation measures of QAOA, CF-QAOA and CF-maQAOA in Fig 8(c)(d). Without noise in Fig 8(c), the average performance of CF-QAOA (orange) is worse than QAOA (blue). This is expected due to the damaging removal of the RZZ gates. However, on the noisy hardware, CF-QAOA outperforms QAOA. CF-maQAOA (green) increases the performance in the noiseless case to almost ideal in the noiseless case, likely due to the additional variational freedom. In the noisy case of Fig 8(d), there is still improvement from the CF-QAOA performance but it is no longer close to ideal. CF-maQAOA is the best performing in both scenarios.

We examined the approximation measures of potential solution states that can be obtained from the distribution upon repeated executions of the circuit. We report two candidate solutions, the *best performing state* in Fig 8(e) with the highest approximation measure of the realized distribution, and the *most probable state* in Fig 8(f) with the most occurrences. The QAOA implementation resulted in a maximal performance of no less than 0.84, which agrees with previous results in Fig 6(a). For CF-QAOA, the maximal performance is no less than 0.96, and for CF-maQAOA it is 0.98. Using the most probable state, the results are again consistent from simulations in 6(b) where performance approaches that of random guessing. However, for CF-QAOA and CF-maQAOA, this is not the case, with better than random performance for the larger instances.

C. Runtime

Finally, we use the observed probabilities of the noisy QAOA distributions to estimate the expected runtime scaling. From previous considerations, we recorded vanishing ground state probability with a fixed number of
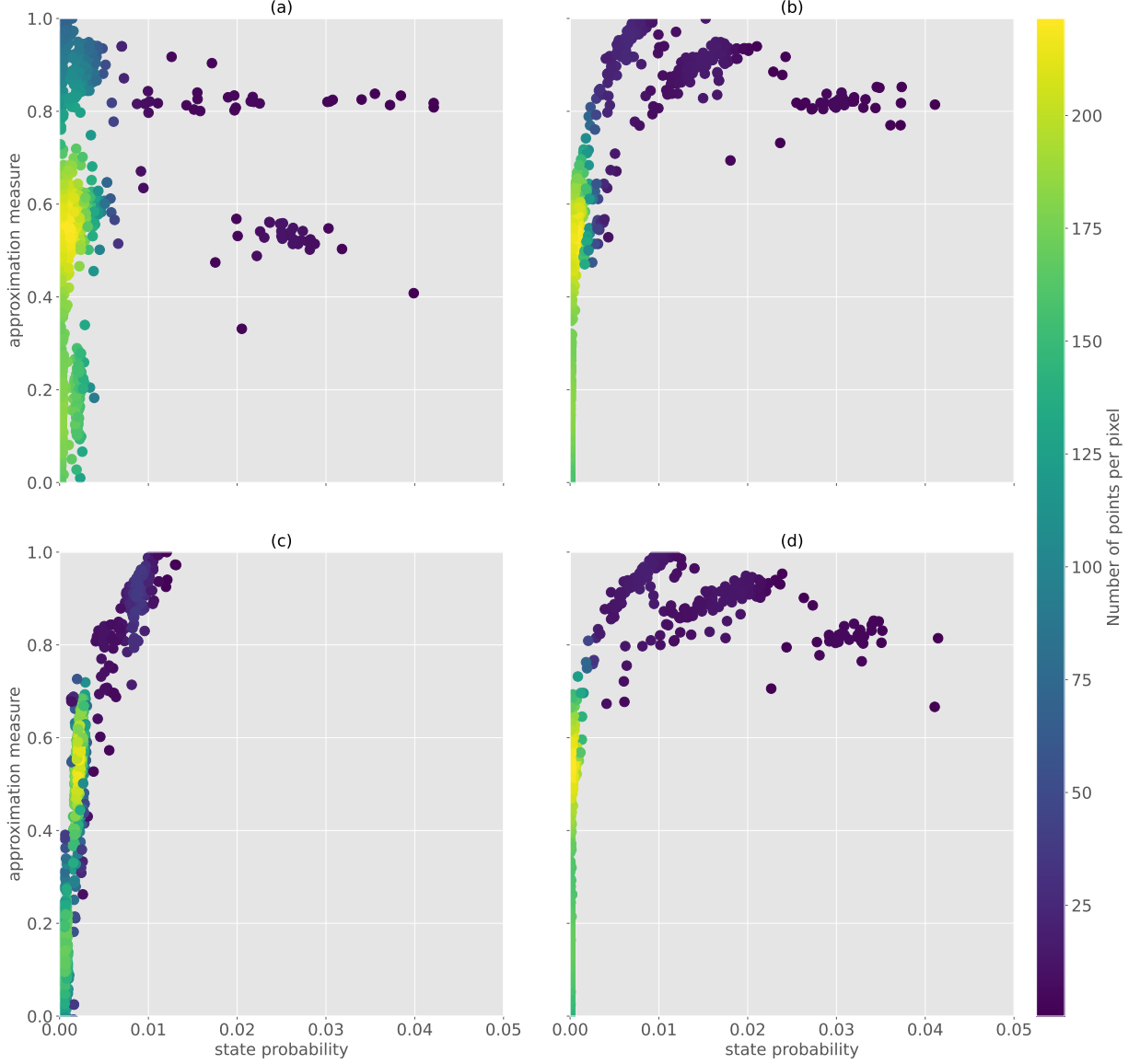
FIG. 5. Density plots of binary states from the QAOA($p = 2$) state probability distribution parameterised with (a) INTERP, (b) FOURIER, (c) precomputed *initial* parameters, and (d) *optimal* parameters, plotting the probability and approximation measures of each binary state. The optimal parameters are optimized with Scipy L-BFGS-B, initialized with the precomputed parameters. By superimposing 20 9-qubit traffic instances, there are $20 \times 2^9 = 10240$ data points for the $2^9 = 512$ possible binary states of each instance, with brighter regions showing higher concentration of points. The red dotted vertical line shows the background probability of $1/2^9 \approx 0.002$, equivalent to sampling the QAOA initial state $|+\rangle^{\otimes N}$.

shots as problem size increased. Thus, it is not possible to use these recorded probabilities of the ground state to estimate a runtime for general instances, as a zero probability corresponds to an infinite runtime. To circumvent this issue, we relax the constraint of requiring the exact ground state, and instead accept an approximate solution state with more than 0.8 approximation measure. Solution states with this lower bound are always obtainable in our experiments, as shown in Fig 8(e). Thus, instead of the ground state probability, the runtime measure is

calculated based on the *total probability* of states with an approximation measure greater than 0.8, plotted in Fig 9(a). Under this relaxation, we no longer use the dimension scaling of 2^N to approximate the probability of randomly guessing the ground state as $1/2^N$ but instead take the total number of states in the solution space with approximation measure greater than 0.8 divided by 2^N, labeled as *random* in the plots. We see again that the total probability for QAOA on a physical device is scaling similarly to random, even with this relaxation of approx-
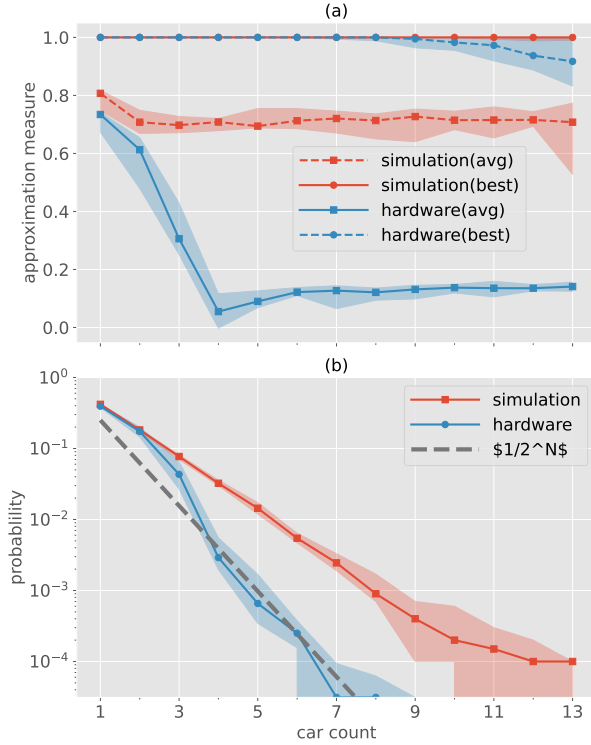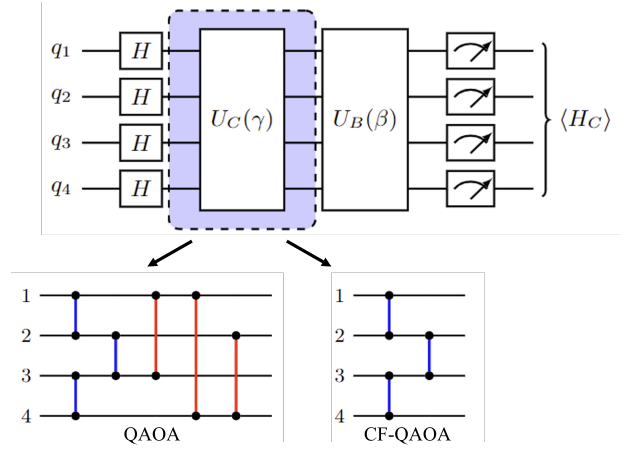
FIG. 7. The modified unitary of the connectivity forced QAOA circuits (CF-QAOA), showing the removal of the red RZZ gates from the original QAOA circuits to implement on a 1d linear array of physical qubits.

obtained by solving the inequality $p_{\text{total}} \geq 0.99$ for K,

$$K_{99} \geq \frac{\log(1 - 0.99)}{\log(1 - p_{\text{single}})}. \tag{19}$$

The estimated circuit execution runtime is then

$$T = t_{\text{single}} \times K_{\min}, \tag{20}$$

where t_{single} is the time required for a single shot. We compared these estimated runtimes with the recorded runtimes of the Gurobi algorithm which were executed on a single CPU core with a clock speed of 3.10GHz. We normalised each metric with respect to the minimal observed value which always occurred for the smallest trivial instance in order to obtain a metric that is agnostic of hardware speeds. In other words, we took the ratio T/T_0 of the estimated runtime T against the runtime T_0 of the easiest, smallest instance. For the quantum heuristics, the ratio clearly becomes independent of t_{single} due to the proportionality assumption. However, note that using this metric is still not a fair comparison between classical and quantum methods, in particular since the Gurobi solution was always found to be optimal with an exact approximation measure of 1 for such small instances. In contrast, the quantum heuristics only had meaningful values when we accepted approximate solutions with more than 0.8 approximation measure. Moreover, this is not a direct comparison as the quantum circuits provide only an *estimated* runtime from the observed probabilities but in reality the runtime would depend on a multitude of factors which were not considered in the experiments. In particular, the unpredictable nature of the classical optimisation loop meant that it is difficult to estimate a runtime for general instances. On the other hand, the *exact* runtime of the classical Gurobi solver can be directly measured from input to output and is a good indication for other like instances. Nevertheless, we use the exact runtime scaling of Gurobi to

FIG. 6. Results with 10000 shots using the noiseless Qiskit Aer simulator (matrix product state) labeled *simulation*, and 32000 shots on IBM Quantum ibmq_montreal labeled *hardware*. The plots show the (a) approximation measure and (b) probability of the ground state against increasing problem size. We use two values for the approximation measure. The first is the expected value over QAOA distribution labeled *avg*(blue), and the second is the approximation measure of the *best* solution state(orange) with the lowest energy in the distribution. In all cases, there were 2 routes assigned to each car, thus the number of variables/qubits is double the number of cars. Each point shows the median of 10 problem instances, and the shaded region is the range of all recorded values over the 10 problem instances.

imate solution states. CF-QAOA improves on this, and CF-maQAOA generally provides the most improvement but with more variance in the measured probabilities.

To calculate the expected runtime scaling, we first calculate the expected number of shots required to have a 99% probability of obtaining a state with at least 0.8 approximation measure, then multiply that by the runtime of each shot. For K independent shots of a quantum circuit, the total probability of measuring a state at least once is

$$p_{\text{total}} = 1 - (1 - p_{\text{single}})^K, \tag{18}$$

where p_{single} is the probability of measuring that state in a single shot. To include all states with approximation measure greater than 0.8 we take the sum total of all state probabilities. The number of shots required to have a 99% probability of outputting such a state can be
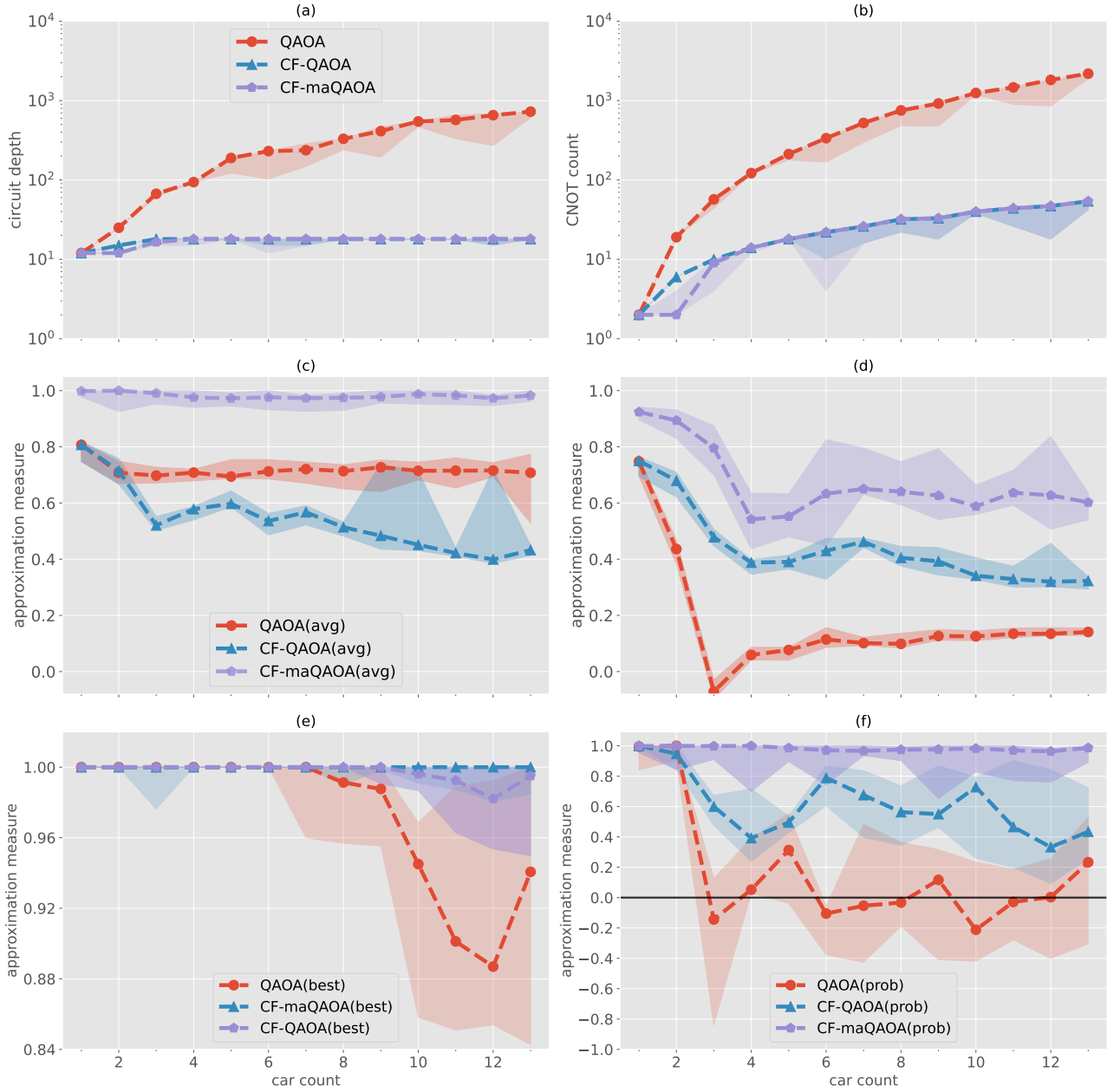
FIG. 8. The transpiled (a) circuit depth and (b) CNOT counts for QAOA, CF-QAOA and CF-maQAOA implementations. Average approximation measure of three QAOA implementations from (c) noiseless simulations, and (d) physical hardware on *ibmq_kolkata*. The approximation measure of (e) the single best solution state and (f) the most probable solution state obtained from the QAOA distributions with 32000 circuit executions. For approximation measure, we plot horizontal lines for maximal approximation measure $y = 1$ and random guessing approximation measure $y = 0$ as benchmarks. We remark that dotted lines indicate a *noiseless* simulation whereas solid lines indicate results from a physical device. The shaded region indicates the whole range of values over 10 traffic instances and the marked point indicate the median value.

see how the quantum heuristics compared. The runtimes are plotted in Fig 9(b). We note that each datapoint for CF-maQAOA has a large variance in the runtime ratio. These are likely due to a variety of factors, most likely attributed to the removal of RZZ gates, the optimisation of additional variational parameters, and the reduced noise in the circuits. All of these affect the final

results in highly non-trivial ways. For further analysis, one could reduce the impact of these experimental errors by either increasing the number of shots in each quantum circuit execution, adding to the number of independent experiments by generating more problem instances, or repeating the same experiment over multiple time periods. Gurobi had the most consistent trend in runtime ratio
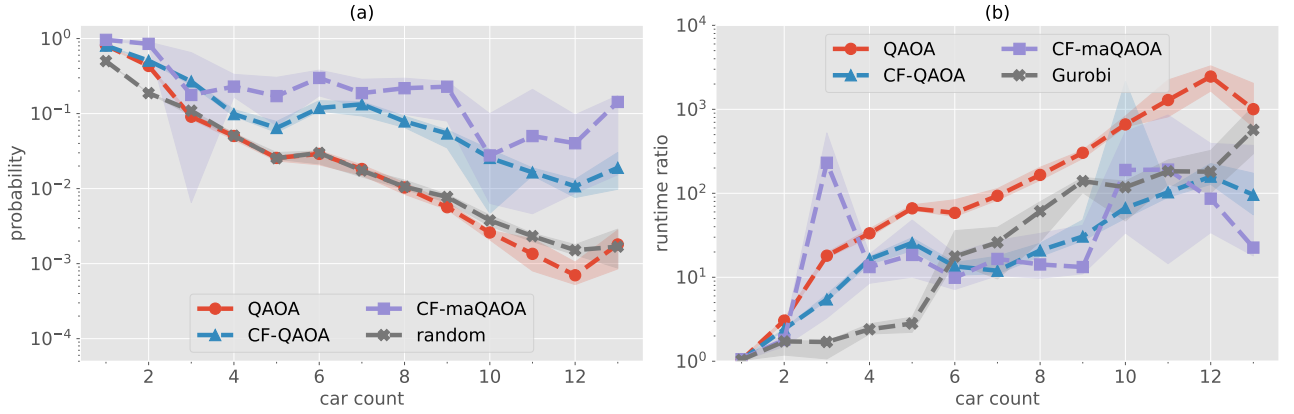
FIG. 9. In (a), the datapoints show the observed total probability of states with more than 0.8 approximation measure from the realized QAOA distributions executed on *ibmq_kolkata*. As a benchmark, we compare against random sampling of the equal superposition state. In (b) the ratios of the estimated runtime using the probabilities in (a) and, as a benchmark, the ratio of the exact recorded runtime of the Gurobi solver running on a single CPU of 3.10GHz clock speed. All ratios are normalized against the smallest recorded value within sample, thus each curve starts at ~ 1. Due to large variance between instances, we restrict the shaded region to the inter-quartile range (IQR) of instances for each car count for visual clarity.

with unsurprisingly an exponential growth for increasing numbers of cars. QAOA appeared to scale similarly to Gurobi although often an order of magnitude higher. The runtime ratio for CF-QAOA performed consistently lower than for QAOA. Moreover, the scaling seemed to compete closely with that of Gurobi. The runtime ratio for CF-maQAOA performed better than other methods although is not consistent.

It is still too early to conclude whether the connectivity-forced heuristic can compete with classical algorithms like Gurobi in the long run. However, we observed that it is possible to make changes to quantum circuits in non-intuitive ways that can bring about better performance closer to that of Gurobi. It is still unclear how much entanglement is lost when removing two-qubit gates in the connectivity-forced QAOA variants. Further work could analyze the entanglement properties of such circuits, as well as its classical simulability. In another direction, it is possible to devise a better compression strategy for the ansatz. For instance, incorporating the problem constraints directly into the mixer and initial states could drastically reduce the number of terms in the Hamiltonian, reducing the number of two-qubit gates

VI. CONCLUSION

In our numerical and device experiments, we investigated the performance of the Quantum Approximate Optimization Algorithm (QAOA) on traffic problems constructed as Quadratic Unconstrained Binary Optimisation (QUBO) problems. Classical simulations were used to investigate QAOA performance on traffic problems under different parameter initializations. In our numerical experiment, the Trotterized Quantum Annealing (TQA) initialization outperformed random initialization (RI).

We observed that QAOA with precomputed parameters gives high-quality solutions to traffic instances, a result known to hold for MaxCut instances [33–37].

On physical hardware, circuit noise destructively impacted the performance due to deep circuits after compilation. We introduced a connectivity-forced circuit compression heuristic (CF-QAOA) whereby long-range RZZ gates over non-connected qubits on the hardware are removed. Compared to QAOA, this heuristic improved approximation measures of the final solution state when run on physical quantum devices. Introducing extra variational parameters into the subsequent connectivity-forced QAOA circuits (CF-maQAOA) provided a trade-off between increased performance and additional classical optimization. We observed that the expected runtime scaling of QAOA and CF-QAOA scaled similarly to Gurobi. Compared to QAOA, the expected runtime scaling on physical quantum hardware was lower for CF-QAOA and even lower for CF-maQAOA, although with a higher variance. Compared to baseline QAOA, while CF-QAOA does worse in noise-free simulations, it has higher performance on physical hardware. With additional classical finetuning, the CF-maQAOA variant provides a potential improvement to CF-QAOA in both performance and runtime scaling.

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

[2] Q. Xu, G. Zheng, Y.-X. Wang, *et al.*, Autonomous quantum error correction and fault-tolerant quantum computation with squeezed cat qubits, npj Quantum Information **9**, 78 (2023).

[3] Y. Kim, A. Eddins, S. Anand, *et al.*, Evidence for the utility of quantum computing before fault tolerance, Nature **618**, 500 (2023).

[4] E. Osaba, E. Villar-Rodriguez, I. Oregi, and A. Moreno-Fernandez-de Leceta, Hybrid quantum computing-tabu search algorithm for partitioning problems: preliminary study on the traveling salesman problem, in *2021 IEEE Congress on Evolutionary Computation (CEC)* (IEEE, 2021) pp. 351–358.

[5] S. Gonzalez-Bermejo, G. Alonso-Linaje, and P. Atchade-Adelomou, GPS: A new TSP formulation for its generalizations type QUBO, Mathematics **10**, 416 (2022).

[6] Ö. Salehi, A. Glos, and J. A. Miszczak, Unconstrained binary models of the travelling salesman problem variants for quantum optimization, Quantum Information Processing **21**, 67 (2022).

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

[8] A. Lucas, Ising formulations of many NP problems, Frontiers in physics **2**, 5 (2014).

[9] N. Sachdeva, G. S. Harnett, S. Maity, *et al.*, Quantum optimization using a 127-qubit gate-model IBM quantum computer can outperform quantum annealers for nontrivial binary optimization problems, arXiv preprint arXiv:2406.01743 (2024).

[10] D. S. Johnson and L. A. Mcgeoch, *The traveling salesman problem: a case study in local optimization* (John Wiley and Sons, 1995).

[11] D. S. Johnson and L. A. McGeoch, Experimental analysis of heuristics for the STSP, in *The traveling salesman problem and its variations* (Springer, 2002) pp. 369–443.

[12] M. L. Fredman, D. S. Johnson, L. A. McGeoch, and G. Ostheimer, Data structures for traveling salesmen, Journal of Algorithms **18**, 432 (1995).

[13] K. Helsgaun, An effective implementation of the Lin–Kernighan traveling salesman heuristic, European journal of operational research **126**, 106 (2000).

[14] S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, Journal of the ACM (JACM) **45**, 753 (1998).

[15] D. Applegate, W. Cook, and A. Rohe, Chained Lin-Kernighan for large traveling salesman problems, Informs journal on computing **15**, 82 (2003).

[16] I. Dunning, S. Gupta, and J. Silberholz, What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO, INFORMS Journal on Computing **30**, 608 (2018).

[17] S. Brandhofer, D. Braun, V. Dehn, *et al.*, Benchmarking the performance of portfolio optimization with QAOA, Quantum Information Processing **22**, 25 (2022).

[18] P. K. Barkoutsos, G. Nannicini, A. Robert, *et al.*, Improving variational quantum optimization using CVaR, Quantum **4**, 256 (2020).

[19] M. Streif, S. Yarkoni, A. Skolik, *et al.*, Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm, Physical Review A **104**, 012403 (2021).

[20] M. T. Khumalo, H. A. Chieza, K. Prag, and M. Woolway, An investigation of IBM quantum computing device performance on combinatorial optimisation problems, Neural Computing and Applications , 1 (2022).

[21] J. Wurtz and P. Love, MaxCut quantum approximate optimization algorithm performance guarantees for p¿1, Physical Review A **103**, 042612 (2021).

[22] G. G. Guerreschi and A. Y. Matsuura, QAOA for MaxCut requires hundreds of qubits for quantum speed-up, Scientific reports **9**, 6903 (2019).

[23] R. Shaydulin and Y. Alexeev, Evaluating quantum approximate optimization algorithm: A case study, in *2019 tenth international green and sustainable computing conference (IGSC)* (IEEE, 2019) pp. 1–6.

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

[25] V. Akshay, H. Philathong, M. E. Morales, and J. D. Biamonte, Reachability deficits in quantum approximate optimization, Physical review letters **124**, 090504 (2020).

[26] M. Alam, A. Ash-Saki, and S. Ghosh, Analysis of quantum approximate optimization algorithm under realistic noise in superconducting qubits, arXiv preprint arXiv:1907.09631 (2019).

[27] J. R. McClean, S. Boixo, V. N. Smelyanskiy, *et al.*, Barren plateaus in quantum neural network training landscapes, Nature communications **9**, 4812 (2018).

[28] L. Bittel and M. Kliesch, Training variational quantum algorithms is np-hard, Physical review letters **127**, 120502 (2021).

[29] S. Wang, E. Fontana, M. Cerezo, *et al.*, Noise-induced barren plateaus in variational quantum algorithms, Nature communications **12**, 6961 (2021).

[30] L. Zhou, S.-T. Wang, S. Choi, *et al.*, Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices, Physical Review X **10**, 021067 (2020).

[31] S. H. Sack and M. Serbyn, Quantum annealing initialization of the quantum approximate optimization algorithm, quantum **5**, 491 (2021).

[32] R. Shaydulin, I. Safro, and J. Larson, Multistart methods for quantum approximate optimization, in *2019 IEEE high performance extreme computing conference (HPEC)* (IEEE, 2019) pp. 1–8.

[33] F. G. Brandao, M. Broughton, E. Farhi, *et al.*, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances, arXiv preprint arXiv:1812.04170 (2018).

[34] J. Wurtz and D. Lykov, Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs, Physical Review A **104**, 052419 (2021).

[35] R. Shaydulin, P. C. Lotshaw, J. Larson, *et al.*, Parameter transfer for quantum approximate optimization of weighted maxcut, ACM Transactions on Quantum Computing **4**, 1 (2023).

[36] A. Galda, X. Liu, D. Lykov, *et al.*, Transferability of optimal QAOA parameters between random graphs, in *2021*

IEEE International Conference on Quantum Computing and Engineering (QCE) (IEEE, 2021) pp. 171–180.

[37] M. Alam, A. Ash-Saki, and S. Ghosh, Accelerating quantum approximate optimization algorithm using machine learning, in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, 2020) pp. 686–689.

[38] A. Ozaeta, W. van Dam, and P. L. McMahon, Expectation values from the single-layer quantum approximate optimization algorithm on Ising problems, Quantum Science and Technology **7**, 045036 (2022).

[39] S. Hadfield, T. Hogg, and E. G. Rieffel, Analytical framework for quantum alternating operator ansätze, Quantum Science and Technology **8**, 015017 (2022).

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

[41] Y. Wu, W.-S. Bao, S. Cao, *et al.*, Strong quantum computational advantage using a superconducting quantum processor, Physical review letters **127**, 180501 (2021).

[42] F. Arute, K. Arya, R. Babbush, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[43] G. J. Mooney, C. D. Hill, and L. C. Hollenberg, Entanglement in a 20-qubit superconducting quantum computer, Scientific reports **9**, 13465 (2019).

[44] G. J. Mooney, G. A. White, C. D. Hill, and L. C. Hollenberg, Generation and verification of 27-qubit Greenberger-Horne-Zeilinger states in a superconducting quantum computer, Journal of Physics Communications **5**, 095004 (2021).

[45] G. J. Mooney, G. A. White, C. D. Hill, and L. C. Hollenberg, Whole-device entanglement in a 65-qubit superconducting quantum computer, Advanced Quantum Technologies **4**, 2100061 (2021).

[46] P. C. Lotshaw, T. Nguyen, A. Santana, *et al.*, Scaling quantum approximate optimization on near-term hardware, Scientific Reports **12**, 12388 (2022).

[47] S. Hadfield, Z. Wang, B. O'gorman, *et al.*, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

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

[49] L. Zhu, H. L. Tang, G. S. Barron, *et al.*, Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer, Physical Review Research **4**, 033029 (2022).

[50] R. Tate, M. Farhadi, C. Herold, *et al.*, Bridging classical and quantum with SDP initialized warm-starts for QAOA, ACM Transactions on Quantum Computing **4**, 1 (2023).

[51] J. Cook, S. Eidenbenz, and A. Bärtschi, The quantum alternating operator ansatz on maximum k-vertex cover, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2020) pp. 83–92.

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

[53] R. LaRose, E. Rieffel, and D. Venturelli, Mixer-phaser ansätze for quantum optimization with hard constraints, Quantum Machine Intelligence **4**, 17 (2022).

[54] R. Herrman, P. C. Lotshaw, J. Ostrowski, *et al.*, Multiangle quantum approximate optimization algorithm, Scientific Reports **12**, 6781 (2022).

[55] F. Neukart, G. Compostella, C. Seidel, *et al.*, Traffic flow optimization using a quantum annealer, Frontiers in ICT **4**, 29 (2017).

[56] F. G. Brandao, M. Broughton, E. Farhi, *et al.*, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances, arXiv preprint arXiv:1812.04170 (2018).

[57] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, Parameter concentrations in quantum approximate optimization, Physical Review A **104**, L010401 (2021).

[58] Gurobi Optimization, LLC, Gurobi optimizer reference manual (2023).

[59] G. Boeing, Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks, Computers, environment and urban systems **65**, 126 (2017).

[60] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, *et al.*, Quantum computing with Qiskit, arXiv preprint arXiv:2405.08810 (2024).

[61] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, Scalable mitigation of measurement errors on quantum computers, PRX Quantum **2**, 040326 (2021).