Applying the Quantum Approximate Optimization Algorithm to the Tail-Assignment Problem

Pontus Vikstål[1,*] Mattias Grönkvist,[2] Marika Svensson,[1,2] Martin Andersson[2] Göran Johansson,[1] and Giulia Ferrini[1]

[1]*Wallenberg Centre for Quantum Technology, Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden*
[2]*Jeppesen Systems AB, 411 03 Gothenburg, Sweden*

Airlines today are faced with a number of large-scale scheduling problems. One such problem is the tail-assignment problem, which is the task of assigning individual aircraft to a given set of flights, minimizing the overall cost. Each aircraft is identified by the registration number on its tail fin. In this paper, we simulate the quantum approximate optimization algorithm (QAOA) applied to instances of this problem derived from real-world data. The QAOA is a variational hybrid quantum-classical algorithm recently introduced and likely to run on near-term quantum devices. The instances are reduced to fit on quantum devices with 8, 15, and 25 qubits. The reduction procedure leaves only one feasible solution per instance, which allows us to map the tail-assignment problem onto the exact-cover problem. We find that repeated runs of the QAOA identify the feasible solution with close to unit probability for all instances. Furthermore, we observe patterns in the variational parameters such that an interpolation strategy can be employed, which significantly simplifies the classical optimization part of the QAOA. Finally, we empirically find a relation between the connectivity of the problem graph and the single-shot success probability of the algorithm.

I. INTRODUCTION

Real-world planning and scheduling problems typically require heuristic algorithms, which is also the case for the tail-assignment problem. The problem is to assign a set of flights to a set of aircraft in order to create a feasible flight schedule for an airline, while minimizing the overall cost [1].

Recently, quantum computing hardware has reached the regime where it is possible to run quantum algorithms, which are hard to simulate on classical hardware, even considering the world's largest supercomputer [2]. This motivates the search for a heuristic quantum algorithm for solving the tail-assignment problem. A promising approach for this is the quantum approximate optimization algorithm (QAOA) [3], which is a heuristic hybrid quantum-classical algorithm designed for solving combinatorial optimization problems. Since the algorithm was first proposed by Farhi *et al.* [3] it has been an active area of research interest [4–10], mainly because of its promising possibility to run on a near-term noisy intermediate-scale quantum (NISQ) device. An important open question is whether a quantum computer, in general, can provide advantages with regards to such classically hard combinatorial optimization problems. Recent studies have indicated that QAOA can have a quadratic Grover-type speed up for state transfer and unstructured search problems [11,12]. Although these results are promising, the performance is largely unknown for QAOA with respect to real-world optimization problems.

Here we present results for QAOA when applied to a real-world aircraft-assignment problem. We perform numerical simulations of an ideal quantum computer to investigate the performance of QAOA for solving the simplified case of the tail-assignment problem where all costs are equal to *zero*. This simplified case can be mapped onto the exact-cover problem [13]. In this context, we note that the solution of random instances of the exact cover and of its restricted version exact cover by three sets on a quantum annealer has been considered in Refs. [14–19]. QAOA for exact cover has recently been executed on a two-qubit quantum computer in a proof-of-principle experiment by some of the authors of the present paper, and collaborators [20].

The paper is organized as follows. In Sec. II, we introduce the tail-assignment problem, and we explain how we extract the exact-cover instances that we analyze in this work. In Sec. III, we review the QAOA and explain how it

_____

*vikstal@chalmers.se

can be utilized to solve the exact-cover problem. Then, in Sec. IV we present numerical results of the performance of QAOA with respect to the tail-assignment-problem extracted instances of exact cover for three different problem sizes. Specifically, we look at the dependence of the success probability as a function of the algorithm iteration level p and of the problem size. Finally, in Sec. V we present what implications these results might have for solving the tail-assignment problem.

II. THE TAIL-ASSIGNMENT PROBLEM

Airlines are confronted daily with several complicated large-scale planning problems involving many different resource types such as passengers, crew, aircraft, maintenance, and ground staff. The typical airline planning process [21] is a sequential process, which starts with the construction of a timetable, followed by a number of aircraft and crew-planning steps. These steps are all large-scale optimization problems and have different objectives, but the overall goal is to maximize profit, safety, and crew satisfaction while minimizing the potential for disruptions. At the same time a large number of complex regulatory, operational, and quality constraints must be satisfied.

The tail-assignment problem [1] is one of the fleet-planning problems where the goal is to decide which individual aircraft (or tail, from the aircraft-tail-identification number) should operate each flight. A set of flights operated in sequence by the same aircraft is called a *route*. In order for a route to be considered legal to operate, it needs to satisfy a number of constraints. For example, the buffer time between the arrival of a flight and the departure of the next flight in the route (the *turn time*) must be above a certain threshold, called the minimum turn time. The minimum turn time can depend on the type of flights involved (domestic and international), the airport, the time of day, and possibly even the individual aircraft characteristics. Another type of constraint is a destination restriction, which prohibits specific aircraft from visiting certain airports, for example due to limited engine thrust combined with short runways. Curfew restrictions are timed restrictions, typically limiting noisy aircraft from operating during night hours at centrally placed airports. Finally, routes must satisfy a number of long- and short-term maintenance constraints. This typically means that the aircraft must regularly visit some airports with a maintenance facility for long enough to perform maintenance.

Now, let F denote the set of flights, T the set of tails, and R the set of all legal routes. Denote by c_r the cost of route $r \in R$ and by C_f the cost of leaving flight f unassigned. The route cost can, for example, indicate how robust the route is with respect to disruptions, what the fuel cost is for the route, or a combination of several different criteria. Let a_{fr} be 1 if flight f is covered by route r and 0 otherwise,

and let b_{tr} be 1 if route r uses tail t and 0 otherwise. The decision variable x_r is 1 if route r should be used in the solution, and 0 otherwise. The variables u_f and v_t are 1 if flight f is left unassigned or tail t is unused, respectively, and 0 otherwise. The tail-assignment problem can now be formulated as

$$\text{minimize} \quad \sum_{r \in R} c_r x_r + \sum_{f \in F} C_f u_f, \quad (1)$$

$$\text{subject to} \quad \sum_{r \in R} a_{fr} x_r + u_f = 1, \quad \forall f \in F, \quad (2)$$

$$\sum_{r \in R} b_{tr} x_r + v_t = 1, \quad \forall t \in T, \quad (3)$$

$$x_r, u_f, v_t \in \{0, 1\}. \quad (4)$$

The objective (1) is to minimize the total cost of the selected routes, subject to constraints (2) ensuring that each flight is assigned to exactly one route and constraints (3) ensuring that each tail is used at most once. Flights can be left unassigned at a cost C_f, but that cost is typically very high compared to the route costs. Not using an aircraft does not come with any penalty cost. The model is an example of a set partitioning problem, which is nondeterministic polynomial-time (NP) hard [22].

A. Solving the tail-assignment problem

Clearly, the number of legal routes for a tail-assignment instance increases exponentially with the number of flights. Since the model presented above requires all the legal routes to be enumerated, it only works for small instances. The solution method traditionally used for these types of models is *column generation* [1]. Column generation starts from some initial solution and uses information from the linear programming dual problem to dynamically generate new variables (columns in the constraint matrix), which are known to potentially improve the current solution. In the tail-assignment case, the problem of generating improving variables turns out to be a resource-constrained shortest-path problem. Given mild conditions on the variable-generation step, the column-generation process can be shown to guarantee optimality for the linear programming (LP) relaxation of the problem, i.e., without the integrality conditions Eq. (4). To find an optimal solution for the full problem including the integrality conditions, column generation must be combined with tree search. The combination of tree search and column generation is often called *branch and price* [23].

B. Instances' extraction

For the purposes of this paper, given the current capability of quantum computers, we focus on tail-assignment instances where we artificially limit the number of routes. The instances are originally solved using

a branch-and-price heuristic, and we randomly select a number of routes from the set of all generated routes to create instances of specific sizes. The solution we find by the branch-and-price heuristic is always included, so we know that all instances have a solution with all flights assigned. This means that we can skip the u_f variables in the model. We also uniquely assign start flights for each aircraft, which means that constraints Eq. (3) can be omitted. Finally, in the remainder of this paper we focus on the decision version of the tail-assignment problem where the goal is to find any solution satisfying all the constraints, disregarding the costs c_r. This decision version of the set partitioning problem is called the exact-cover problem, it is known to be NP complete [24], and can be expressed as the following optimization problem:

$$\text{minimize} \quad 0 \tag{5}$$

$$\text{subject to} \quad \sum_{r \in R} a_{fr} x_r = 1, \quad \forall f \in F, \tag{6}$$

$$x_r \in \{0, 1\}, \tag{7}$$

where the minimization on 0 is left to recall that this formulation stems from the tail-assignment problem, where we neglect the costs in Eq. (1). Despite the simplification introduced, the exact-cover problem is still very relevant for the study of tail-assignment as many airlines, including, for example, Air France, consider the tail-assignment problem to be a pure feasibility problem [25].

III. QAOA APPLIED TO THE TAIL-ASSIGNMENT PROBLEM

A large class of NP-complete optimization problems including the exact cover (and even many NP-hard problems) can naturally be expressed as the problem of finding the ground state, or minimum energy configuration, of a quantum Ising Hamiltonian [26]

$$\hat{H}_C = \sum_{i < j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z + \sum_{i=1}^{n} h_i \hat{\sigma}_i^z. \tag{8}$$

We refer to this quantum Ising Hamiltonian as a cost Hamiltonian. In this section, we derive explicitly the cost Hamiltonian corresponding to the exact-cover problem expressed by Eqs. (6) and (7). Later, we recall the QAOA algorithm, and in particular how it makes use of the cost Hamiltonian for finding its minimum energy configuration.

A. Ising formulation of the exact-cover problem

Consider the formulation of the exact-cover problem presented in Eqs. (6) and (7). By subtracting 1 from both sides of Eq. (6) and squaring the expression an energy

function formulation is obtained:

$$\mathcal{E}(s_1, \ldots, s_{|R|}) = \sum_{f=1}^{|F|} \left(\sum_{r=1}^{|R|} a_{fr} x_r - 1 \right)^2. \tag{9}$$

Here $|R|$ and $|F|$ denote the cardinality of R and F, respectively. We see that all constraints are satisfied if the energy, Eq. (9), is equal to *zero*.

By replacing the binary variables $x_r \in \{0, 1\}$ with spin variables $s_r \in \{-1, 1\}$ as

$$x_r = \frac{s_r + 1}{2}, \tag{10}$$

and expanding the square of Eq. (9) we obtain the *Ising energy function* for the exact-cover problem

$$\mathcal{E}(s_1, \ldots, s_{|R|}) = \sum_{f=1}^{|F|} \left(\sum_{r=1}^{|R|} a_{fr} \frac{s_r + 1}{2} - 1 \right)^2$$

$$= +\frac{1}{4} \sum_{f=1}^{|F|} \sum_{r=1}^{|R|} \sum_{r'=1}^{|R|} a_{fr} a_{fr'} s_r s_{r'}$$

$$+ \frac{1}{2} \sum_{f=1}^{|F|} \sum_{r=1}^{|R|} a_{fr} s_r \left(\sum_{r'=1}^{|R|} a_{fr'} - 2 \right)$$

$$+ \frac{1}{4} \sum_{f=1}^{|F|} \left(\sum_{r=1}^{|R|} a_{fr} - 2 \right)^2. \tag{11}$$

By defining $J_{rr'}$ as

$$J_{rr'} \equiv \frac{1}{2} \sum_{f=1}^{|F|} a_{fr} a_{fr'}, \tag{12}$$

and h_r as

$$h_r \equiv \frac{1}{2} \sum_{f=1}^{|F|} a_{fr} \left(\sum_{r'=1}^{|R|} a_{fr'} - 2 \right), \tag{13}$$

the Ising energy function becomes

$$\frac{1}{2} \sum_{r=1}^{|R|} \sum_{r'=1}^{|R|} J_{rr'} s_r s_{r'} + \sum_{r=1}^{|R|} h_r s_r + \text{const}, \tag{14}$$

where the constant is equal to $\frac{1}{4} \sum_{f=1}^{|F|} \left(\sum_{r=1}^{|R|} a_{fr} - 2 \right)^2$. The sum of all the diagonal terms ($i = j$) in the first sum is equal to $\text{Tr}(J)$ since $s_i^2 = 1$; because J_{ij} is symmetric,

i.e., $J_{ij} = J_{ji}$, we can further simplify the expression and write the Ising energy function as

$$\mathcal{E}(s_1, \ldots, s_{|R|}) = \sum_{r<r'} J_{rr'} s_r s_{r'} + \sum_{r=1}^{|R|} h_r s_r + \text{const}, \quad (15)$$

where we absorb $\frac{1}{2}\text{Tr}(J)$ into the constant. Finally, by promoting the spin variables to Pauli spin matrices $s_i \to \hat{\sigma}_i^z$, a cost Hamiltonian in the form of Eq. (8) is obtained.

B. The quantum approximate optimization algorithm

The QAOA starts from an initial quantum state, which is taken as a superposition of all possible computational basis states $|+\rangle^{\otimes n}$. The second step of QAOA is to apply in an alternating sequence two parametrized noncommuting quantum gates, $\hat{U}(\gamma)$ and $\hat{V}(\beta)$, that are defined as

$$\hat{U}(\gamma) \equiv e^{-i\gamma \hat{H}_C}, \quad \hat{V}(\beta) \equiv e^{-i\beta \hat{H}_M}, \quad (16)$$

where $\hat{H}_C$ is the cost Hamiltonian given by Eq. (8), and $\hat{H}_M \equiv \sum_{i=1}^{n} \hat{\sigma}_i^x$ is a so-called mixing Hamiltonian. The alternating sequence continues for a total of p times with different variational parameters $\vec{\gamma} = (\gamma_1, \ldots, \gamma_p)$ with $\gamma_i \in [0, 2\pi]$ if $\hat{H}_C$ has integer-valued eigenvalues, and $\vec{\beta} = (\beta_1, \ldots, \beta_p)$ with $\beta_i \in [0, \pi]$, such that the final variational state obtained is

$$|\psi_p(\vec{\gamma}, \vec{\beta})\rangle \equiv \hat{V}(\beta_p)\hat{U}(\gamma_p) \ldots \hat{V}(\beta_1)\hat{U}(\gamma_1)|+\rangle^{\otimes n}. \quad (17)$$

The parametrized quantum gates are then optimized in a closed loop using a classical optimizer, see Fig. 1. The objective of the classical optimizer is to find the optimal variational parameters that minimize the expectation value of the cost Hamiltonian

$$(\vec{\gamma}^*, \vec{\beta}^*) = \arg\min_{\vec{\gamma}, \vec{\beta}} E_p(\vec{\gamma}, \vec{\beta}), \quad (18)$$

where

$$E_p(\vec{\gamma}, \vec{\beta}) \equiv \langle \psi_p(\vec{\gamma}, \vec{\beta})|\hat{H}_C|\psi_p(\vec{\gamma}, \vec{\beta})\rangle. \quad (19)$$
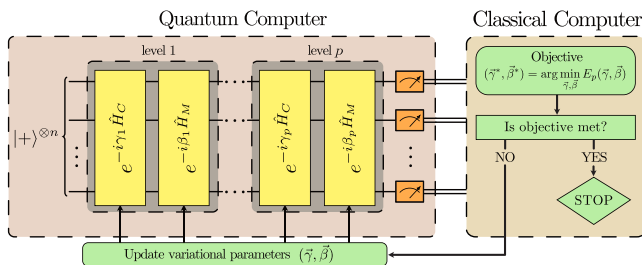

FIG. 1.   Schematic representation of the QAOA. The quantum processor prepares the variational state, depending on variational parameters. The variational parameters $(\vec{\gamma}, \vec{\beta})$ are optimized in a closed loop using a classical optimizer.

Note that this requires, in principle, multiple state preparations and measurements. Once the best possible variational parameters are found, they are used to create the state $|\psi_p(\vec{\gamma}^*, \vec{\beta}^*)\rangle$, using the quantum processor for the state preparation. Then, one samples from this state by measuring in the computational basis, and the cost of the configuration obtained in the measurement, given by Eq. (8), is evaluated. The latter step is classically efficient.

The success probability is defined as the probability of finding the qubits in their ground-state configuration $|x_{\text{sol}}\rangle$ when performing a single-shot measurement of the $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ state, i.e.,

$$F_p(\vec{\gamma}, \vec{\beta}) \equiv |\langle x_{\text{sol}}|\psi_p(\vec{\gamma}, \vec{\beta})\rangle|^2, \quad (20)$$

where $x_{\text{sol}} = x_1 x_2 \ldots x_n$ is the bit string corresponding to the solution. Given this success probability we can ask the following: what is the probability of having observed the solution at least once after m repeated measurements? The answer is given by

$$1 - [1 - F_p(\vec{\gamma}, \vec{\beta})]^m. \quad (21)$$

Thus to have the probability $(1 - \varepsilon)$ of observing the solution, m has to be

$$m > \frac{\log \varepsilon}{\log[1 - F_p(\vec{\gamma}, \vec{\beta})]}. \quad (22)$$

To fix the ideas, consider a fair coin. In order to have a probability higher than 99.9% of observing head at least once, one has to flip and "measure" the coin ten times.

In what follows, we apply this paradigm to solve the exact-cover problem, by using the corresponding cost Hamiltonian, expressed by Eq. (8) with J_{ij} and h_i given by Eqs. (12) and (13), respectively.

IV. RESULTS

We examine instances for three different problem sizes of the tail-assignment problem given in Table I, corresponding to 8, 15, and 25 routes. As clear from Eq. (8), this requires quantum processors with 8, 15, and 25 qubits, respectively.

A. Energy landscape

Firstly, we can reduce the search space by noting that the eigenvalues of both Hamiltonians $\hat{H}_C$ and $\hat{H}_M$ are integer

TABLE I.   Information about the problem instances.

Routes	Flights	No. of instances	No. of sol. per instance
8	77	10	1
15	77	9	1
25	278	10	1

valued. As a consequence, the expectation value Eq. (19) has even symmetry, i.e., $E_p(\vec{\gamma}, \vec{\beta}) = E_p(-\vec{\gamma}, -\vec{\beta})$. This symmetry allows us to restrict the domain of each γ_i to $\gamma_i \in [0, \pi]$.

To highlight the difficulty of finding the best variational parameters we can visualize the landscape of the expectation value $E_1(\gamma, \beta)$, as well as the corresponding success probability $F_1(\gamma, \beta)$, as a function of γ and β, for $p = 1$, by evaluating them on a fine grid $[0, \pi] \times [0, \pi]$. Figure 2 shows the simulation result for one of the 25 route instances. The variational parameters resulting in the lowest expectation value, $(\gamma_{\exp}, \beta_{\exp})$, and those resulting in the highest success probability, $(\gamma_{\text{succ}}, \beta_{\text{succ}})$, are approximately the same. In fact $|\gamma_{\exp} - \gamma_{\text{succ}}| \simeq 0$ and $|\beta_{\exp} - \beta_{\text{succ}}| \simeq 0.047$. Note that this is not obvious, since QAOA only minimizes the expectation value, and does not explicitly maximize the success probability; a low expectation value does not necessarily translates onto a high success probability. For example, consider a variational state that is a linear combination of low-energy excited eigenstates of the cost Hamiltonian. This state could potentially have a low expectation value while the success probability is zero. Similarly, a variational state that is a linear combination of the ground state with high-energy eigenstates could have a high success probability, while the cost Hamiltonian expectation value is large. However, in the limit $p \to \infty$, 100% success probability is always achieved [3]. For our problem, it is clear from Eq. (9) that the minimum energy of the first excited state is at least 1, so if we find an average cost, which is lower than 1 for our variational state, we know that the ground state is a part of this state. The
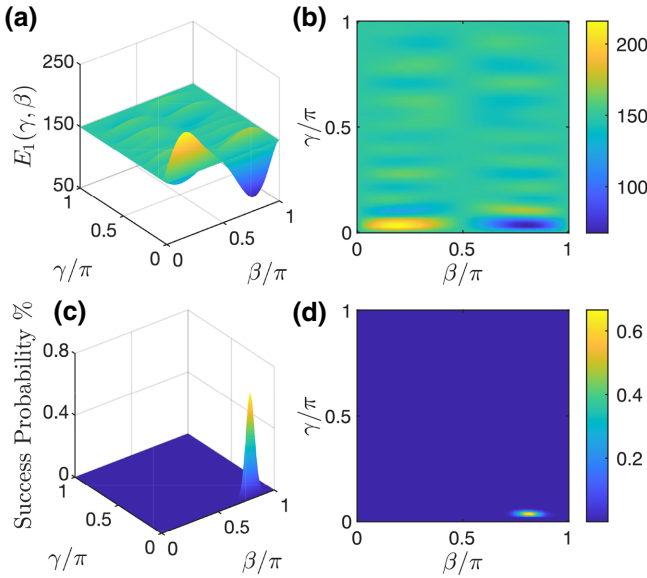
corresponding plots for one of the 8 and 15 route instances are shown in Appendix A, Fig. 5 and Fig. 6. We note that all figures have qualitatively similar shape and that the optimal variational parameters for $p = 1$ are located in the same region.

B. Low iteration levels: patterns in optimal variational parameters

Before we look at the performance of QAOA, we search for patterns in the optimal variational parameters for low iteration levels of the QAOA algorithm, namely up to $p = 5$. Patterns in the optimal variational parameters have been observed before in the context of MaxCut in Ref. [27], where it was shown that if a pattern exists it is possible to use different heuristics that can drastically speed up the classical optimization part of QAOA. This can potentially help us simulate the solution of our instances for intermediate p level beyond $p = 5$, namely for $5 < p \leq 20$.

In order to find the optimal variational parameters, one possible approach consists of a grid-search method. However, evaluation of the cost-Hamiltonian expectation value on a fine grid for higher dimensions quickly becomes computational expensive due to the large search space $[0, \pi]^p \times [0, \pi]^p$. Therefore, we discard the grid-search method and resort to another optimization routine for finding good variational parameters for $1 \leq p \leq 5$. This optimization routine is still exhaustive but more computational efficient. It distributes several random start points in the variational parameter landscape, and runs the gradient-based Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [28] for every start point from which it records the global optimum. We provide relevant details in Appendix B. In Fig. 3 we present the

FIG. 2. Simulation results for one of the 25 route instances as a function of γ and β for $p = 1$. (a),(b) Expectation value $E_1(\gamma, \beta)$. (c),(d) Success probability $F_1(\gamma, \beta)$.
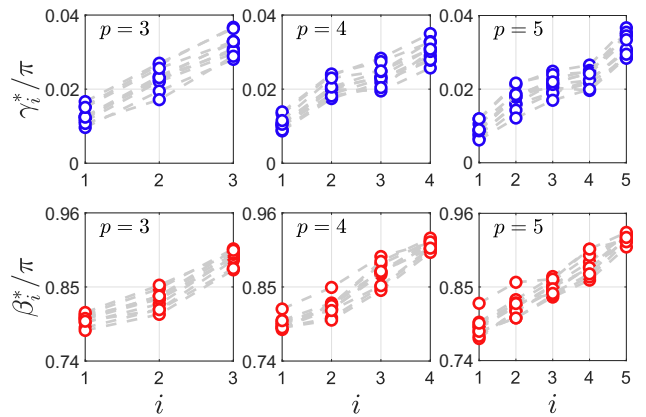

FIG. 3. Optimal QAOA variational parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ for the 8 route instances, for $3 \leq p \leq 5$. The pattern is visualized by plotting the optimal variational parameters where each gray dashed line connects the variational parameters for one 8 route instance.

optimal variational parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ from $p = 3$ up to $p = 5$ for the 8 route instances. We observe that a persistent pattern shows up, and that both γ_i and β_i tend to increase slowly with $i = 1, 2, \ldots, p$. An analogous analysis for the 15 route instances, shown in the Appendix in Fig. 7, yields a qualitatively similar result. For the 25 route instances, it is not possible to perform this analysis, because for $p > 1$ performing an exhaustive search becomes too computationally expensive.

C. Intermediate iteration levels: analysis of success probability

Based on the patterns found in the previous section, we now use an interpolation-based strategy, introduced in Ref. [27], in order to study the performance of intermediate p-level QAOA. This strategy consists in predicting a good starting point for the variational parameters search at level $p + 1$ for each individual instance based on the best variational parameters found at level p for the same instance. From the produced starting point we run the gradient-free Nelder-Mead method [29,30], which is reported in Ref. [27] to work equally well as the BFGS method, for this heuristic strategy. The Nelder-Mead algorithm is implemented in MATLAB version R2019b using the `fminsearch` function. Furthermore, in order to force the Nelder-Mead algorithm to terminate after sufficiently many iterations, we set the two stopping

criteria − maximum number of function evaluations and iterations − both to $60p$. We furthermore make the assumption that a pattern in the variational parameters also exists in each of the 25 route instances, and we therefore use the interpolation strategy mentioned above for each of these instances as well, as an educated guess. We base this assumption on the qualitatively similar shape of the expectation value landscape that the three different problems sizes investigated had for $p = 1$.

We use the aforementioned interpolation strategy for finding good local optimal variational parameters up to $p = 10$ for all the instances. The success probability as a function of iteration level p averaged over all the instances for the three different problem sizes is plotted in Fig. 4(a). Moreover, we select one instance from each problem size, for which we perform simulations up to $p = 20$. In Fig. 4(b) we plot the success probability for these three instances. The corresponding variational parameters $\vec{\gamma}^*$ and $\vec{\beta}^*$ are provided in Appendix A, Fig. 8. It is observed that the success probability increases with the parameter p in both the averaged and the single-instance cases, reaching almost 100% for the instances where we use high iteration level $p = 20$.

From the results in Figs. 4(a) and 4(b) we also note that the 25 route instances are easier to solve than the 15 route instances, in the sense that the success probability is higher for the former instances at any given iteration level p of the algorithm. This fact can seem
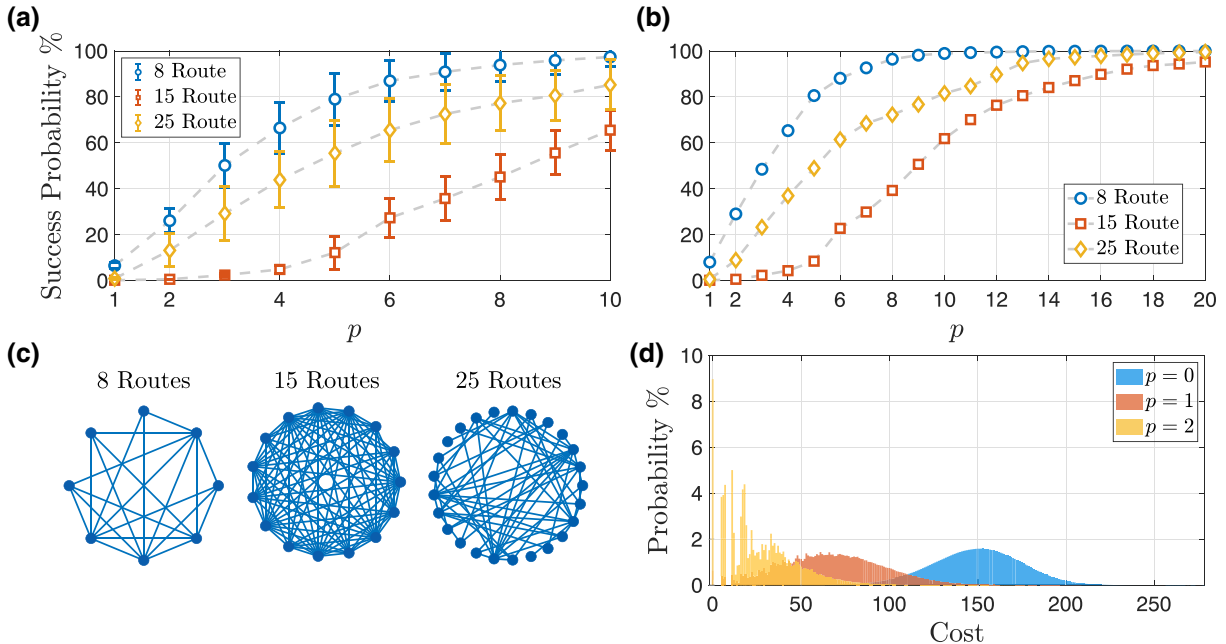

FIG. 4. (a) Average success probability as a function of the iteration level p using the best found variational parameters for the three different problem sizes. Error bars in the figure represent the standard deviation of the average success probability. (b) Success probability $F_p(\vec{\gamma}^*, \vec{\beta}^*)$ as a function of p for one selected instance from each problem size. (c) Graph representation of the three instances shown in (b). (d) Probability that a measurement of the state $|\psi_p(\vec{\gamma}^*, \vec{\beta}^*)\rangle$ will yield a certain cost (or equivalently, eigenvalue of the cost Hamiltonian) for the iteration levels $p = 0, 1, 2$, where $p = 0$ is the initial or "random" state $|+\rangle^{\otimes n}$.

counterintuitive, as one could naively think that larger instances correspond to harder problems. We perform further analysis in order to explain this apparent contradiction.

We start by representing each instance as a graph, by identifying J_{ij} in Eq. (12) with an adjacency matrix. In this way, each vertex in the graph represents a route and two vertices are connected by an edge if they share a flight. The valency of a vertex, i.e., the number of incident edges to the vertex, indicates how many "clauses" the vertex is contained in, or in other words how many other vertices it has to compete with. In Table II we list the average valency of each vertex for the three problem sizes. We note that the 15 instances have more than twice the average valency compared to the 25 route instances. This is also visualized in Fig. 4(c), where the graph connectivity for one instance of each problem size is represented. It is clear that the connectivity for the 15 instances is the most dense. Establishing a general connection between the hardness of the instances and their valency is beyond the scope of our paper. However, such a connection is known to exist in some specific contexts, e.g., for exact cover by three sets [31,32]. This hints to the fact that such a connection might exist also for our instances, despite the fact that they are not in the form of exact cover by three sets. To elucidate further why denser graphs are more difficult to solve with the QAOA we recall, following Refs. [3] and [9], that the expectation value Eq. (19) can be expressed as a sum of expectation values involving all possible subgraphs. Subgraphs are obtained by starting from an edge $\langle ij \rangle$ of a graph, e.g., the type of graph given in Fig. 4(c), and "walking" along the graph at most p steps away from that edge, for a given iteration level p. Indicating with $f_g(\vec{\gamma}, \vec{\beta})$ the contribution to the expectation value from subgraph g, and with w_g the corresponding subgraph occurrence, it is possible to rewrite the expectation value as $E_p(\vec{\gamma}, \vec{\beta}) = \sum_g w_g f_g(\vec{\gamma}, \vec{\beta})$. Since the contribution to the expectation value is different for each subgraph, the higher the number of important subgraphs (with a significant w_g) is, the harder it is to make the cost close to zero for a given iteration level p, since the QAOA needs to make each individual term in the sum small. Since the average valency of a graph contributes to the number of subgraphs, this results in a lower success probability for the 15 route instances, as [as we show in Fig. 4(c) and Table II] those possess higher average valency.

TABLE II. Valency of the graphs. The first column in the table is the number of routes. The second column is the average valency of a vertex taken as an average over all the instances. The corresponding standard deviation is given in the third column.

Routes	Mean	Standard deviation
8	5.15	0.24
15	12.62	0.42
25	5.54	0.77

Finally, in Fig. 4(d) we visualize how the probability of measuring a certain cost, or equivalently an eigenvalue of the cost Hamiltonian, given the state $|\psi_p(\vec{\gamma}^*, \vec{\beta}^*)\rangle$, changes for each iteration $p = 0, 1, 2$ of QAOA using the best found variational parameters for one of the 25 route instances. It is clear that the effect of iterating QAOA is that the probability of configurations with lower cost increases. This validates the effectiveness of QAOA in producing output configurations corresponding to low-energy states of the cost Hamiltonian, when the iteration level p is increased. In particular, for $p = 2$ a peak at the zero-cost configuration appears clearly, corresponding to a success probability of 8.97%. This results in only 74 measurements needed, in order to have a probability greater than 99.9% of measuring the solution at least once.

In order to benchmark the effectiveness of QAOA in solving this problem against other quantum algorithms, in Appendix C we compare the time to solution of QAOA with that of quantum annealing, and find that QAOA outperforms quantum annealing for all the 8 and 15 route instances.

Finally, noise and imperfection in practical experimental implementations on a quantum computer will induce departures from the obtained success probabilities, and it is an open question whether realistic hardware will still be able to produce the good solution, with satisfactory success probability. Although a complete study of the effect of noise is beyond the scope of the present paper, in Appendix D we characterize the effect of a simple depolarizing noise model, to study how noise affects the performance of QAOA. As expected, we find that with noise an optimal value of p exists. Beyond that value of p, the success probability starts to decrease, due to the larger effect of decoherence when the gate sequence becomes longer. However, for the optimal p, the success probability is only halved, still pointing to relevance of the use of QAOA for solving this problem even in realistic experimental conditions.

V. CONCLUSIONS

In conclusions, we simulate the solution of instances of the exact-cover problem that stem as a reduction of the tail-assignment problem to the case where the goal is to find any solution satisfying all the constraints, using the QAOA.

Our results indicate that these instances can be solved satisfactorily by means of QAOA, yielding relative high success probabilities even for low iteration level of the algorithm. For instance, for the 25-qubit case, we obtain a success probability of 8.97% for $p = 2$ in the single-measurement scenario. This corresponds to a success probability of 99.9% for 74 repeated measurements. This low iteration level translates into a low circuit depth needed

for the implementation of this algorithm, corroborating feasibility on a near-term quantum device.

Moreover, we observe patterns for the variational parameters $(\vec{\gamma}, \vec{\beta})$, which allow for a substantial simplification of the classical optimization problem of finding the best variational parameters, despite the fact that the problem instances are extracted from a real-world problem.

Our analysis reveals nontrivial properties in the connectivity of the instances considered, i.e., the 15-qubit instances are more connected than the 25-qubit ones. A thorough study of the connectivity and graph type that are relevant for the tail-assignment problem in the context of complex quantum networks [33,34] is beyond the scope of the present paper, but stems as an interesting perspective. Another interesting question is whether the implementation of the QAOA algorithm on hardware with restricted connectivity would still yield nontrivial success probabilities, as shown in Ref. [35] for MaxCut on three regular graphs.

Our successful solution with QAOA of small-size instances of exact cover extracted from tail assignment motivates further studies, such as the use of QAOA for solving instances with multiple feasible solutions, where costs are reintroduced, and where the number of considered routes is larger, towards tackling real-world instances.

It remains an open question how the performance of QAOA compares with existing classical algorithms for solving large instances of the exact-cover problem extracted from the tail-assignment problem. However, we expect that current known methods as branch and bound, cutting planes, or branch and cut [36] will perform well on these small instances. Further investigations are needed in order to compare the scaling in terms of time complexity of QAOA fixing a target success probability (i.e., the required iteration level p) and standard classical methods, when the size of the problem increases.

While finalizing this work, we became aware of an alternative method for the optimization of the variational parameters, that makes use of the *Gibbs objective function*, defined as $-\log\langle e^{-\eta \hat{H}_C}\rangle$, where $\eta > 0$, instead of the expectation value Eq. (8) [37]. This approach is expected to be superior because the Gibbs objective function rewards lower-energy states, which increases the success probability. We leave the use of this approach for optimization of the variational parameters in our problem to further study.
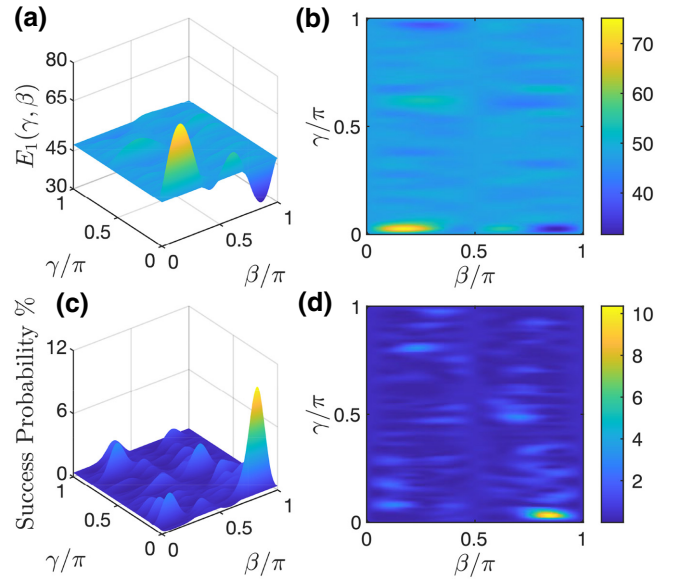
FIG. 5. Simulation results for one of the 8 route instances as a function of γ and β for $p = 1$. (a),(b) Expectation value $E_1(\gamma, \beta)$. (c),(d) Success probability $F_1(\gamma, \beta)$.

APPENDIX A: ADDITIONAL FIGURES

This Appendix contains additional figures that are mentioned in the main text. Figures 5 and 6 shows the expectation value $E_1(\gamma, \beta)$ and the corresponding success probability $F_1(\gamma, \beta)$, as a function of γ and β for one of the 8 and 15 route instances.

Figure 7 illustrates the optimal QAOA parameters for the 15 route instances.

Figure 8 shows the variational parameters $\vec{\gamma}^*$ and $\vec{\beta}^*$ that are used in Fig. 4(b).
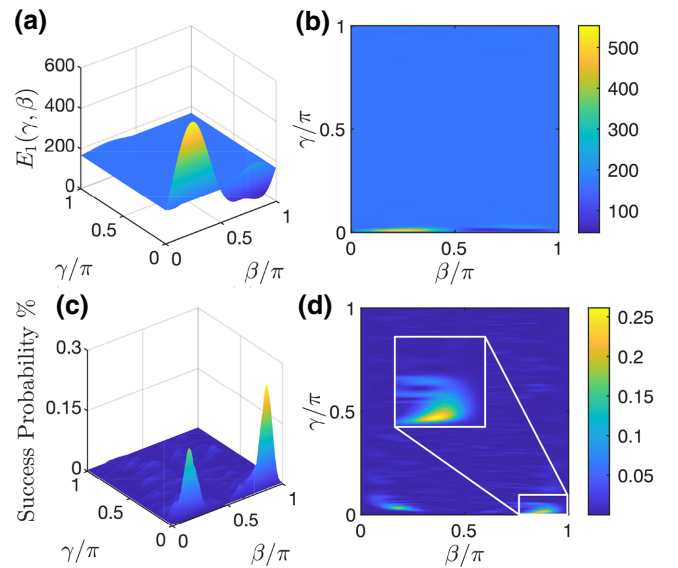

FIG. 6. Simulation results for one of the 15 route instances as a function of γ and β for $p = 1$. (a),(b) Expectation value $E_1(\gamma, \beta)$. (c),(d) Success probability $F_1(\gamma, \beta)$.
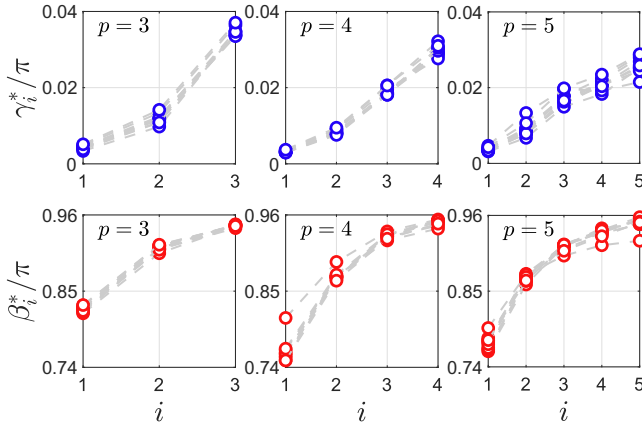
FIG. 7. Optimal QAOA variational parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ for the 15 route instances, for $3 \leq p \leq 5$. The pattern is visualized by plotting the optimal parameters where each gray dashed line connects the optimal variational parameters of one particular instance.

APPENDIX B: NUMERICAL SIMULATIONS

The numerical simulations for the exhaustive search method is done in MATLAB version R2019b where the MultiStart function is used to search thoroughly for the optimal variational parameters. MultiStart attempts to find multiple local minimums to the objective function by starting from various points in the variational parameter landscape. When run, it distributes start points to multiple processors (cpus) that run in parallel. From a start point it runs a *local solver* and when the solver reaches

FIG. 8. The best found $\vec{\gamma}^*$ and $\vec{\beta}^*$ for the three instances shown in Fig. 4(b).

a stopping criterion it terminates and the obtained minima from the solver is stored in an array. When MultiStart runs out of start points it stops, and the array with minimums from the solver is sorted by the objective function value in ascending order. The parameters where the objective function is the lowest is then returned as output. As *local solver* we use the BFGS algorithm [28] which is implemented as fmincon in MATLAB. The number of random start points is chosen to be 4×10^3. This number is empirically determined by running the simulations a few times for this value and observing that the minimum of the objective function always converges to the same value and gives the same parameters. As mentioned, the solver stops when the solver's stopping criteria is met. Two examples of such criterion's are the function tolerance and the step tolerance. The first one, the function tolerance, is a lower bound on the change in the value of the objective function during a step, that is if $|F_p(\vec{\gamma}, \vec{\beta}) - F_p(\vec{\gamma}', \vec{\beta}')| <$ FunctionTolerance, the iteration ends. The second one, the step tolerance, is such that if the solver attempts to take a step that is smaller than $|\vec{\gamma} - \vec{\gamma}'|^2 + |\vec{\beta} - \vec{\beta}'|^2 <$ StepTolerance, the iteration ends. Both StepTolerance and FunctionTolerance are set to their default values, which is 10^{-6}.

APPENDIX C: COMPARISON: TIME TO SOLUTION OF QUANTUM ANNEALING VERSUS QAOA

In this Appendix, we compare the time to solution of the quantum-annealing (QA) algorithm with that of the QAOA. In quantum annealing we start from the same initial state as the QAOA, which is in fact the ground state of the mixing Hamiltonian that we use in QAOA, but with a minus sign in front, $\hat{H}_M^{QA} \equiv -\hat{H}_M = -\sum \hat{\sigma}_i^x$. By adiabatically changing from the mixing Hamiltonian to the cost Hamiltonian the system remains in its instantaneous ground state throughout the evolution, and end up in the ground state of the cost Hamiltonian. For a linear time dependence, the quantum-annealing Hamiltonian is given by

$$\hat{H}(t) = \frac{t}{T}\hat{H}_C + \left(1 - \frac{t}{T}\right)\hat{H}_M^{QA}, \quad 0 \leq t \leq T, \quad (C1)$$

where $\hat{H}_C$ is the cost Hamiltonian, $\hat{H}_M^{QA}$ is the quantum-annealing starting Hamiltonian, and T is the total annealing time. It is known that rather than running the algorithm adiabatically, it can be advantageous to run the algorithm for a shorter time (nonfully adiabatically). On the one hand, this yields to a finite probability to excite higher-energy states and decreases the success probability on a single run; on the other hand, since the annealing time T is shorter, one can then increase the number of repetitions, yielding an increase of the total success probability, on several runs.
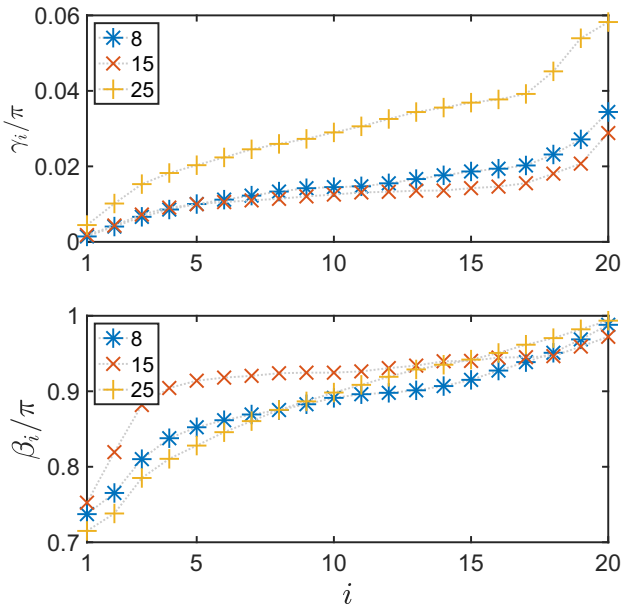
Therefore, one can define the time to solution (TTS), which is a measure of how quickly the algorithm can find the optimal solution. The time to solution for QA is defined by [38]

$$\text{TTS}_{\text{QA}}(T) = T \frac{\log(1 - p_d)}{\log[1 - F_{\text{GS}}(T)]},$$

where p_d is the target success probability that we fix to 99%, and $F_{\text{GS}}(T)$ is the ground state population after running the algorithm for a time T. The optimal $\text{TTS}_{\text{QA}}(T)$ is thus given by the time T that minimize

$$\text{TTS}_{\text{QA}}^{\text{opt}} = \min_{T>0} \text{TTS}_{\text{QA}}(T).$$

Following the spirit of Ref. [27], it is possible to interpret the sum of the optimal variational parameters of the QAOA as the total "annealing" time that is used, in order to sequentially evolve the system under the action of each of the two Hamiltonians, $T_p = \sum_{i=1}^{p}(|\gamma_i^*| + |\beta_i^*|)$. Thus, the time to solution for QAOA is

$$\text{TTS}_{\text{QAOA}}(p) = T_p \frac{\log(1 - p_d)}{\log[1 - F_p(\vec{\gamma}^*, \vec{\beta}^*)]},$$

where $F_p(\vec{\gamma}^*, \vec{\beta}^*)$ is given by Eq. (20). Analogously as for QA, the optimal $\text{TTS}_{\text{QAOA}}(p)$ is given by

$$\text{TTS}_{\text{QAOA}}^{\text{opt}} = \min_{p>0} \text{TTS}_{\text{QAOA}}(p).$$

We would of course like T_p to be as small as possible, therefore we subtract all the optimal β^* values by π. We can do this since $\psi_p(\vec{\gamma}, \vec{\beta})$ is π periodic in β up to a global phase. This π-shifted value of β is the value that one would obtain, if one would choose to use the quantum-annealing

mixer Hamiltonian (i.e., the one with a minus in front of the summation), instead of the mixer commonly used for the QAOA.

We run the QA algorithm for all the 8 and 15 route instances for different total annealing times T and record the optimal TTS that we find. In Fig. 9 we plot the TTS^{opt} for both algorithms, and find that the $\text{TTS}_{\text{QAOA}}^{\text{opt}}$ is smaller than $\text{TTS}_{\text{QA}}^{\text{opt}}$ for all the instances. For the 15 route instances, QAOA is one order of magnitude faster in achieving 99% success probability. It should be noted that the linear annealing schedule chosen here is a restricted form of quantum annealing and that some of the performance gap might be addressed by optimizing the annealing schedule within a practical family of curves (similar to the optimization of the QAOA).

APPENDIX D: DEPOLARIZING NOISE

In this Appendix we perform a simple study of how depolarizing noise affects the performance of QAOA. We model the depolarizing noise as random uncorrelated Pauli X, Y, or Z operations using the error gate

$$\mathcal{E} = (1 - \eta)I + \frac{\eta}{3}(X + Y + Z), \qquad \text{(D1)}$$

where η is the probability that an error occurs, that we fix to 1%. This error gate acts on each individual qubit between the applications of the cost and mixing Hamiltonian, see Fig. 10(a). We then repeat the circuit sufficiently many times to get a statistical average over the noise. In Fig. 10(b) we plot the success probability with noise for the same 8 and 15 route instances as shown in Fig. 4(b). A trade-off appears between the level of iteration of the algorithm p, and the success probability. In particular, we observe that for $p > 6$ the success probability starts to decrease for the 8 route instance, while for the 15 route instance it levels off, indicating that the gain of increasing one level p equals the decrease due to the noise. This is
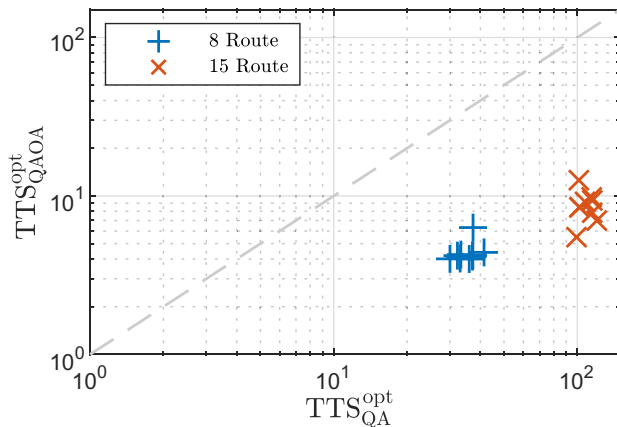

FIG. 9.   The optimal time to solution for QAOA and QA. The fact that the markers are below the dashed diagonal line means that QAOA outperforms QA in the time required to achieve a 99% success probability.
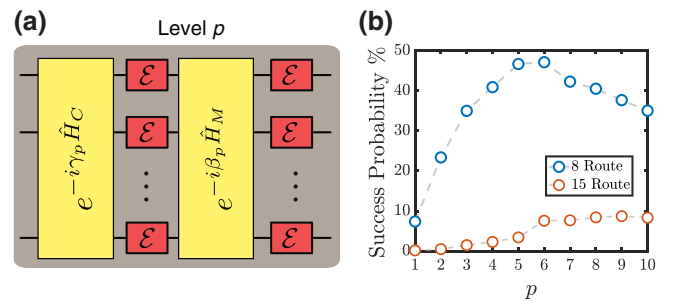

FIG. 10.   (a) After each application of the cost and mixing Hamiltonian of the QAOA an error gate $\mathcal{E}$ given by Eq. (D1) is independently applied to every qubit. (b) Success probability with noise for one of the 8 and 15 route instances.

expected, as faulty gates decrease the fidelity of the prepared state with the best theoretically found variational state. However, the resulting success probabilities at $p = 6$ are roughly halved with respect to the noiseless case.

———————

[1] M. Grönkvist, Ph.D. thesis, Chalmers University of Technology and Göteborg University, 2005.

[2] F. Arute *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

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

[4] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv:1811.08419v1 (2018).

[5] M. Willsch, D. Willsch, F. Jin, H. D. Raedt, and K. Michielsen, Benchmarking the quantum approximate optimization algorithm, arXiv:1907.02359 (2019).

[6] R. Shaydulin and Y. Alexeev, Evaluating quantum approximate optimization algorithm: A case study, arXiv:1910.04881 (2019).

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

[8] C. Xue, Z.-Y. Chen, Y.-C. Wu, and G.-P. Guo, Effects of quantum noise on quantum approximate optimization algorithm, arXiv:1909.02196 (2019).

[9] 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, arXiv:1812.04170 (2018).

[10] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size, arXiv:1910.08187 (2019).

[11] M. Y. Niu, S. Lu, and I. L. Chuang, Optimizing QAOA: Success probability and runtime dependence on circuit depth, arXiv:1905.12134 (2019).

[12] Z. Jiang, E. G. Rieffel, and Z. Wang, Near-optimal quantum circuit for Grover's unstructured search using a transverse field, Phys. Rev. A **95**, 062317 (2017).

[13] A. Parmentier and F. Meunier, Aircraft routing and crew pairing: Updated algorithms at air france, Omega **93**, 102073 (2020).

[14] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem, Science **292**, 472 (2001).

[15] A. P. Young, S. Knysh, and V. N. Smelyanskiy, First-Order Phase Transition in the Quantum Adiabatic Algorithm, Phys. Rev. Lett. **104**, 020502 (2010).

[16] B. Altshuler, H. Krovi, and J. Roland, Anderson localization makes adiabatic quantum optimization fail, Proc. Natl. Acad. Sci. **107**, 12446 (2010).

[17] V. Choi, Different adiabatic quantum optimization algorithms for the NP-complete exact cover and 3SAT problems, arXiv:1010.1221 (2010).

[18] H. Wang and L.-A. Wu, Ultrafast adiabatic quantum algorithm for the NP-complete exact cover problem, Sci. Rep. **6**, 22307 (2016).

[19] T. Graß, Quantum Annealing with Longitudinal Bias Fields, Phys. Rev. Lett. **123**, 120501 (2019).

[20] A. Bengtsson, P. Vikstål, C. Warren, M. Svensson, X. Gu, A. Frisk Kockum, P. Krantz, C. Križan, D. Shiri, I.-M. Svensson, G. Tancredi, G. Johansson, P. Delsing, G. Ferrini, and J. Bylander, Quantum approximate optimization of the exact-cover problem on a superconducting quantum processor, arXiv:1912.10495 (2019).

[21] T. L. Jacobs, L. A. Garrow, M. Lohatepanont, Frank S. Koppelman, G. M. Coldren, and H. W. Purnomo, in *Quantitative Problem Solving Methods in the Airline Industry: A Modeling Methodology Handbook* (Springer US, Boston, MA, 2012), p. 35.

[22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Fransisco, 1979).

[23] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, Branch-and-price: Column generation for solving huge integer programs, Oper. Res. **46**, 316 (1996).

[24] R. M. Karp, *Reducibility Among Combinatorial Problems* (Springer US, Boston, MA, 1972), p. 85.

[25] A. Parmentier and F. Meunier, Aircraft routing and crew pairing: Updated algorithms at air france, Omega **93**, 102073 (2020).

[26] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 5 (2014).

[27] 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, arXiv:1812.01041 (2018).

[28] C. G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, IMA J. Appl. Math. **6**, 76 (1970); R. Fletcher, A new approach to variable metric algorithms, Comput. J. **13**, 317 (1970); D. Goldfarb, A family of variable-metric methods derived by variational means, Math. Comput. **24**, 23 (1970); D. F. Shanno, Conditioning of quasi-newton methods for function minimization, Math. Comput. **24**, 647 (1970).

[29] J. A. Nelder and R. Mead, A simplex method for function minimization, Comput. J. **7**, 308 (1965).

[30] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions, SIAM J. Optim. **9**, 112 (1998).

[31] V. Kalapala and C. Moore, The phase transition in exact cover, arXiv:cs/0508037 (2005).

[32] J. Raymond, A. Sportiello, and L. Zdeborová, Phase diagram of the 1-in-3 satisfiability problem, Phys. Rev. E **76**, 011101 (2007).

[33] F. Sansavini and V. Parigi, Continuous variables graph states shaped as complex networks: optimization and manipulation, arXiv:1912.03265 (2019).

[34] R. Albert and A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. **74**, 47 (2002).

[35] E. Farhi, J. Goldstone, S. Gutmann, and H. Neven, Quantum algorithms for fixed qubit architectures, arXiv:1703.06199 (2017).

[36] M. Conforti, G. Cornuejols, and G. Zambelli, *Integer Programming* (Springer International Publishing, 2014), p. 5.

[37] L. Li, M. Fan, M. Coram, P. Riley, and S. Leichenauer, Quantum optimization with a novel Gibbs objective function and Ansatz architecture search, arXiv:1909.07621v1 (2019).

[38] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).