THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Applying quantum approximate optimization to the heterogeneous vehicle routing problem

DAVID FITZEK

Microtechnology and Nanoscience - MC2
Applied Quantum Physics Laboratory
Chalmers University of Technology
Göteborg, Sweden, 2022

Applying quantum approximate optimization to
the heterogeneous vehicle routing problem
DAVID FITZEK

Technical Report MC2-450
ISSN 1652-0769

Applied Quantum Physics Laboratory
Microtechnology and Nanoscience - MC2
Chalmers University of Technology
SE–412 96 Göteborg, Sweden
Phone +46 (0)31-772 1000

Cover: Schematic representation of a computational process consisting of mapping a combinatorial optimization problem to an Ising model and searching for approximate solutions through a hybrid quantum-classical optimization algorithm.

Chalmers Digitaltryck
Göteborg, Sweden, 2022

Applying quantum approximate optimization to
the heterogeneous vehicle routing problem
DAVID FITZEK
Microtechnology and Nanoscience - MC2
Applied Quantum Physics Laboratory
Chalmers University of Technology

Abstract

Quantum computing offers new heuristics for combinatorial problems. With small- and intermediate-scale quantum devices becoming available, it is possible to implement and test these heuristics on small-size problems. A candidate for such combinatorial problems is the heterogeneous vehicle routing problem (HVRP): the problem of finding the optimal set of routes, given a heterogeneous fleet of vehicles with varying loading capacities, to deliver goods to a given set of customers. This licentiate thesis is an extended introduction to the accompanying paper, which consists of a study of a new formulation of the HVRP applicable to both quantum annealers and programmable noisy intermediate-scale quantum (NISQ) devices.

Keywords: Quantum computing, variational quantum algorithm, quantum approximate optimization algorithm, vehicle routing, combinatorial optimization

Publications

A

Applying quantum approximate optimization to the heterogeneous vehicle routing problem
David Fitzek, Toheed Ghandriz, Leo Laine, Mats Granath, Anton Frisk Kockum
ArXiv:2110.06799 (2021)

Other publications

B **Deep Q-learning decoder for depolarizing noise on the toric code**
David Fitzek, Mattias Eliasson, Anton Frisk Kockum, Mats Granath
Physical Review Research **2**, 023230 (2020)

Acknowledgments

To begin with, I would like to express my deepest thanks to my supervisors, Anton Frisk Kockum and Mats Granath, for all their support and guidance. Göran Johansson was of great help and a valuable source of discussion for me. In addition, I am thankful for the collaboration with colleagues at Volvo Group: Leo Laine, Toheed Ghandriz, Henrik Kaijser, Petter Wirfält, Fredrik Sandblom, Jenny Erneman, Thierry Hours and Nicolas Andersson. Furthermore, I would like to thank all the great co-workers and friends at the AQP and QT divisions for making MC2 a very enjoyable workplace. Last but not least, I could not have accomplished this without my friends, Piera's, and my family's generous support.

David Fitzek, Göteborg, April 2022

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AQO	Adiabatic quantum optimization
HVRP	Heterogeneous vehicle routing problem
NISQ	Noisy intermediate-scale quantum
NN	Neural network
NP	Non-deterministic polynomial-time
NP-complete	Non-deterministic polynomial-time complete
NP-hard	Non-deterministic polynomial-time hard
QAOA	Quantum approximate optimization algorithm
SAT	Boolean satisfiability problem
VQA	Variational quantum algorithm

Symbols defined in Chapter 1

θ	Variational parameter
C	Cost function
H_0	Initial Hamiltonian
H_C	Cost Hamiltonian
N	Number of objects
T	Time

Symbols defined in Chapter 2

β, γ	Variational parameter
$\boldsymbol{s}$	Collection of spins
$\mathcal{C}$	Total value of knapsack
$\mathcal{N}$	Set of objects
$\neg$	Logical NOT operation
σ^x	Pauli X matrix
$\vee$	Logical OR operation
$\wedge$	Logical AND operation
A, B	Positive constants
c_i	Cost of object

$G(V,E)$	Graph with nodes and vertices
H	Ising Hamiltonian
h_i	Strength of applied field at an atom with index i
H_M	Mixing Hamiltonian
$J_{i,j}$	Interaction strength between spins i and j
s_i	Spin with index i
W	Capacity of knapsack
w_i	Weight of object
W_{ij}	Edge-weight encoding the distance from city i to city j
x, y, z	Binary decision variables

Symbols defined in Chapter 3

$\#q$	Number of qubits
$\hat{A}, \hat{B}$	Two non-commuting operators
$\|+\rangle$	Equal superposition state
$\|0\rangle$	Computational zero state, analogous to classical zero bit
$\|1\rangle$	Computational one state, analogous to classical one bit
τ	Time segment defined as T/p
$\tilde{H}$	Hadamard gate
p	Integer value defining the number of parameterized layers in the quantum approximate optimization algorithm
r	Some positive integer
$U(\beta)$	Operator applying the mixing Hamiltonian
$U(\boldsymbol{\beta}, \boldsymbol{\gamma})$	Unitary transformation of variational quantum circuit
$U(\gamma)$	Operator applying the cost Hamiltonian
$U(T)$	Time evolution operator

Symbols defined in Chapter 4

$\mathcal{E}$	Set of edges
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	A fully connected graph
$\mathcal{N}$	Set of nodes or vertices
$\mathcal{N}_0$	Set of customers to visit
$\mathcal{V}$	Set of vehicle types
c_{ij}^v	Cost of traveling on edge (i, j) with the vehicle of type v
f_{ij}^v	The amount of goods leaving node i to go to node j using truck of type v
m_v	Number of available vehicles of type v
N_0	Number of customers to visit
Q^v	Vehicle capacity for vehicle type v
q_i	Positive customer demand
t^v	Fixed vehicle cost
V	Number of trucks
v	Vehicle of type v

Chapter 1

Introduction

The theory of quantum computing emerged in the eighties when physicists began to discuss paradigms of computation that integrated quantum mechanics. Since Benioff [1] and Deutsch [2] published their pioneering work on the subject of quantum Turing machines and universal quantum computation, the field has also developed to encompass the theoretical application of simulation of quantum systems [3–8]. Along the way, other algorithms were discovered, including a quantum algorithm provided by Peter Shor [9] for finding prime factors of composite integers in 1994. Arguably it is what set the quantum computing revolution in motion. Shor's algorithm rendered most classical cryptographic protocols insecure, where once they were widely believed to be unbreakable [10, 11]. Quantum algorithms and heuristics [12–14] have since developed into a sophisticated subfield of quantum computing, with applications including machine learning [15–21], simulation of quantum systems [22–24], natural language processing [25–28], cryptography [9], and search and optimization [29–32], where particularly optimization problems are common in industry.

Large-scale optimization problems underlie many critical issues in industry, so the automotive industry is not exempt from the challenge. The optimization problems range from logistics to resource allocation and planning [33–35]. Finding solutions to these optimization problems is driven by powerful classical algorithms [36–39]. With the advent of quantum computing, we need to assess if a quantum computer can improve the current state of the art or help classical heuristics find better solutions. In this thesis, we pursue the path of solving optimization problems with the help of hybrid quantum-classical algorithms. In particular, we focus on logistics problems.

In the following part of the introduction, we motivate why the pursuit of quantum computing is essential. In this context, we introduce the notion of noisy intermediate-scale quantum (NISQ) devices. Then, we present the adiabatic quantum optimization algorithm and continue with a more general class of quantum algorithms known as variational quantum algorithms (VQAs).

1.1 Why quantum computing matters

Several problems are considered to be classically hard and “quantumly” easy. Consider the example involving a large number, such as 184,568,767. This large number might prompt the question: is this number prime — is it divisible only by one and itself? Computer scientists can answer this with fast classical algorithms [40], for which the computational resources scale polynomially with the size of the problem. Our number, 184,568,767, is not a prime number. So we might ask: What are its prime factors? But no such fast algorithm can be applied here unless we use a quantum computer.

We can look for more such problems. Simulating a many-particle quantum system would be a natural place to start for physicists. The difficulty of these systems lies within the simulation of many highly entangled particles since they cannot be separated as individual particles but must be considered as a whole. Considering the problem of simulating many-particle quantum systems, Pines and Laughlin pointed out, ”we can write down the equations precisely — they are the equations that describe how atomic nuclei and electrons interact electromagnetically. But we can’t solve those equations.” [41] Laughlin dramatized how pointless the task was: ”No computer existing, or that will ever exist, can break this barrier” of solving the equations describing many entangled particles [41].

However, Richard Feynman had proposed a groundbreaking idea years before Laughlin and Pines had written these words: using computers to simulate quantum systems [3]. In Feynman’s words: “Nature isn’t classical dammit, and if you want to make a simulation of Nature, you better make it quantum mechanical, and by golly, it’s a wonderful problem because it doesn’t look so easy.” [42] Feynman’s vision was to use a quantum computer to solve the quantum physics problems that scientists could not solve with digital computers. While Laughlin and Pines knew Feynman had made this proposal years ago, they dismissed his idea as impractical. Feynman’s proposal was made some 40 years ago; today, we are just beginning to reach the stage where quantum computers can address challenging quantum problems in valuable ways.

There are good reasons suggesting that quantum computers will surpass classical computers for some tasks. First, some problems are thought to be hard for classical computers, but quantum algorithms have been discovered to solve them efficiently. Shor’s algorithm is the best-known example. It efficiently solves the problem of finding the prime factors of a large composite integer. Despite decades of trying, no efficient classical factoring algorithm has been discovered. However, that does not prove that there exist no efficient algorithms, and finding one would be exciting. Scientists have provided evidence that quantum states easily prepared with a quantum computer can be classically hard to generate based on complexity theory. By sampling from a quantum device in such a state, we are sampling from a correlated probability distribution that cannot be sampled efficiently by classical methods [43–46]. Furthermore, we cannot efficiently simulate a quantum computer using a digital computer, which is perhaps the strongest argument for why quantum computing is powerful. Scientists have striven to find better ways to simulate quantum systems using digital computers for decades, but to no avail [47].

1.1.1 Noisy intermediate-scale quantum devices (NISQ)

Even with fault-tolerant quantum computers still in the distant future, quantum technology is entering a new era. To describe this new era, Preskill [47] coined the term noisy intermediate-scale quantum (NISQ). In this context, intermediate-scale refers to the size of quantum computers with 50 to a few hundred qubits. "Noisy" implies that we cannot protect the system against losses and errors, which will lead to severe limitations in near-term quantum devices.

Nevertheless, a quantum device with 50 controllable qubits is a significant milestone because that is likely beyond what can be simulated by brute force using the most powerful existing digital supercomputers. A breakthrough experimental realization [46] has demonstrated the first evidence that quantum computing can outperform classical computation.

These NISQ devices give physicists new tools to explore the physics of many entangled particles, and there might even be commercial applications [6, 48–52]. Therefore, the ongoing development of such devices [45, 46, 53, 54] provides an opportunity to test algorithms on small problem instances, leading to new solutions to many different problems ranging from optimization and chemistry to Hamiltonian simulation.

1.2 Adiabatic quantum optimization

Adiabatic quantum optimization (AQO) was among the first algorithms proposed for quantum computing [55] and is shown to be universal for time, $T \to \infty$, and equivalent to digital quantum computing [56].

Suppose we have two non-commuting Hamiltonians: H_C, whose ground state encodes the solution to a problem of interest, and H_0, whose ground state is easy to prepare. Further, let us prepare the quantum system in the ground state of H_0 and adiabatically change the Hamiltonian for a time T according to

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_C. \tag{1.1}$$

Then if we choose T large enough, the quantum system will remain in the ground state for all times, by the adiabatic theorem of quantum mechanics [57]. There is a fundamental limitation to this class of algorithms, giving rise to whether they would run faster than classical algorithms. The limitation is that one finds typically, for a problem size N,

$$T = \mathrm{O}\left[\exp\left(\alpha N^{\beta}\right)\right] \, , \tag{1.2}$$

for the system to remain in the ground state. This exponential scaling in the worst case indicates that it is unlikely that intractable problems can be solved in polynomial time by AQO. Nevertheless, the coefficients α, β may be smaller than for known classical algorithms, so there is still a possibility that AQO may be more efficient than classical algorithms on some classes of problems.

1.3 Quantum algorithms

Several fault-tolerant quantum algorithms have been demonstrated to outperform the best current classical algorithms in the past few decades [9, 58]. Still, it is currently unknown whether NISQ devices can provide such an advantage [30] for some real-world problems.

As discussed in Section 1.1.1, current devices are limited in size and prone to noise. We, therefore, focus on algorithms that are supposed to be noise-resilient and can be employed on NISQ devices and exclude algorithms that require fault-tolerant quantum computers. The class of algorithms tailored for NISQ devices is called variational quantum algorithms (VQAs).

1.3.1 Variational quantum algorithms

Variational quantum algorithms (VQAs) are a class of hybrid quantum-classical algorithms that have emerged to run on the current generation of NISQ devices [12, 47, 59]. Many experimental proposals for NISQ devices involve training a closed-loop optimization between a quantum device and a classical computer. Such hybrid quantum-classical algorithms are popular for chemistry [51, 60, 61], optimization [23, 29–32], and machine learning [15, 18, 19, 21, 62] applications. The first step is to define a cost (or loss) function C, which encodes the solution to the problem. Next, one proposes an Ansatz, i.e., a quantum operation depending on a set of continuous or discrete parameters $\boldsymbol{\theta}$ that can be optimized. This Ansatz is then optimized in a hybrid quantum-classical loop (see Fig. 1.1) to solve the optimization task,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, C(\boldsymbol{\theta})\,. \tag{1.3}$$

Further, this class of algorithms takes advantage of the toolbox of classical optimization since they run only the parametrized quantum circuit on the quantum computer and outsource the parameter optimization to the classical optimizer. Some VQAs have the advantage of keeping the quantum circuit depth shallow and thus working well with the imposed limitations by NISQ devices in contrast to quantum algorithms developed for the fault-tolerant era.

There are several shortcomings known to this class of quantum algorithms, including challenges in trainability and accuracy. A pressing challenge is the efficient optimization of variational parameters used in the quantum circuit. Researchers have shown that this optimization problem belongs to the complexity class NP-hard [63]. Nevertheless, we shall not be discouraged by this

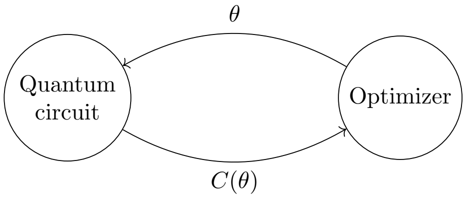

Figure 1.1: A schematic description of the VQA. The goal is to minimize the energy/cost $C(\theta)$ of the variational quantum circuit by adapting the variational parameters θ.

result since neural networks (NNs) faced the same challenge before the discovery of back-propagation[64] and the availability of potent hardware.

The trainability of VQAs has been rigorously studied, and researchers have chosen, due to its simplicity and hardware efficiency, random circuits as initial guesses for exploring the space of quantum states. It has been shown that a large family of random quantum circuits do not serve as a good ansatz for VQAs, because they have gradients that vanish almost everywhere and therefore making this choice unsuitable for hybrid quantum-classical algorithms run on more than a few qubits. This phenomenon is also known as barren plateaus [59, 65–68].

The subset of VQAs working with combinatorial problems is called the quantum approximate optimization algorithm (QAOA). Here, the classical optimization problem is encoded in the ground state of a system of interacting spins.

1.4 Thesis outline

This thesis introduces the appended paper, where we develop a mapping for the heterogeneous vehicle routing problem (HVRP) to an Ising Hamiltonian and numerically simulate the QAOA for this combinatorial optimization problem.

The thesis is organized as follows. In Chapter 2, we introduce the Ising model and discuss that the task of finding its ground state energy is a member of the complexity class NP-hard. Further, we give explicit mappings from classical combinatorial optimization problems to an Ising model. Next, Chapter 3 is dedicated to the hybrid quantum-classical algorithm QAOA. We discuss its connection to the Ising model and briefly describe its relevant components. Finally, in Chapter 4, we introduce the heterogeneous vehicle routing problem (HVRP) and develop the mapping to the corresponding Ising model to combine this problem with the QAOA. We give an overview of the appended paper in Chapter 5 and conclude in Chapter 6 by summarizing our work and looking to the future.

Chapter 2

The Ising model

To address a combinatorial optimization problem with a quantum algorithm, we must capture a specific problem that we seek to solve in an understandable way for the quantum device. This may be accomplished by expressing the optimization problem in terms of quantum variables.

This chapter explores how a combinatorial optimization problem can be framed as a cost Hamiltonian in the Ising form. First, we describe the Ising model and how it was initially proposed to model magnetism. Further, we explain that the decision version of the Ising model belongs to the complexity class NP-complete and therefore is closely connected to all other NP-complete problems. We conclude this chapter by providing explicit mappings for the Knapsack problem and traveling salesperson problem (TSP) to an Ising model. These NP-complete problems are the key ingredients needed for the HVRP.

2.1 A model to describe magnetism

Models are of fundamental importance in several scientific contexts and help us understand the world around us. They have described numerous phenomena, including the inflation of the universe, general circulation models of global climate, the double-helix model of DNA, evolutionary models in biology, agent-based models in social science, and equilibrium models of markets. Models of nature that can encompass a wide range of completely different systems are the most fruitful—understanding how these models work leads to understanding all the physical systems the model can represent.

One of the most extensively used models in physics is the Ising model. It was initially proposed in the mid-1920s by Ernst Ising and Wilhelm Lenz to explain the inner workings of magnetic materials. The idea is to model a magnetic material as a collection of atoms. Each of these atoms has a spin, either aligning or anti-aligning with an applied magnetic field. Moreover, all of the spins interact with one another [69, 70]. Let $s_i \in \{-1, +1\}$ serve as the spin of the ith atom. The energy of a collection of spins $\boldsymbol{s}$ is

$$E(\boldsymbol{s}) = \sum_{i,j'} J_{i,j} s_i s_j + \sum_i h_i s_i \equiv \langle \boldsymbol{s}, \boldsymbol{J}\boldsymbol{s} \rangle + \langle \boldsymbol{h}, \boldsymbol{s} \rangle \,, \tag{2.1}$$

where h_i is the strength of the applied field at atom i and $J_{i,j}$ acts as an interaction strength between spins i and j. At low temperatures, where the system is strongly biased towards low-energy states, a positive $J_{i,j}$ favors anti-aligned neighbouring spins ($s_i s_j = -1$). If $J_{i,j}$ is negative the opposite occurs: s_i and s_j tend to align ($s_i s_j = 1$). The simple model Ising devised provides excellent experimental agreement with the properties of many magnetic materials, although he was unaware of it at the time [71].

Ising's model eventually proved suitable for modeling a wide range of different physical systems. This framework can describe any system made of a set of independent elements that interact with each other pairwise. In our work, we are using a generalized type of Ising model, which is not on a lattice and where interactions may be long-ranged and irregular. Over 12,000 papers have been published using the Ising model since 1969, describing systems in fields ranging from artificial intelligence [72, 73], zoology [74] to quantum computing [29–32, 71, 75].

2.2 The Ising model and NP-hardness

Now that we have introduced the Ising model we can ask if finding a configuration with a particular energy is "easy" or "hard". We know that not all computational problems are equal. Some problems are so "easy" or simple that even large instances can be solved efficiently (solutions can be obtained in polynomial time). Several problems may not scale well with problem size and become intractable as they become more complex. There is essentially no hope of solving them exactly. Computer scientists have formalized this insight and have divided problems into different complexity classes.

We consider two kinds of problems: decision problems having a yes or no answer and optimization problems that aim to minimize a cost or energy measure. When given a candidate's solution to a decision problem, its correctness can be easily verified. We can easily verify which solution is correct for these problems once we have a candidate solution. This class of problems, where given a "yes" answer (candidate solution), there is a short proof that establishes the answer is correct, are called non-deterministic polynomial (NP) [71].

In 1971, Stephen Cook [76] and Leonid Levin [77] independently showed that there is a class of NP problems having a unifying feature. If we can solve any problem in this class, we can solve all problems in NP with only polynomial overhead. This subclass of NP problems is called NP-complete. In 1971, Stephen Cook identified the boolean satisfiability (SAT) problem as the first NP-complete problem [76]. The goal of this problem is to determine whether the variables of a given Boolean formula can be consistently replaced with TRUE or FALSE in such a way that the formula always evaluates to TRUE. Since Cook and Levin's influential work, many other problems have been shown to be NP-complete. In 1982, Francisco Baharona showed that finding the global ground state of the Ising spin glass model is NP-hard [78]. NP-hard problems are at least as hard as any problem in NP. The decision version of the Ising model is NP-complete. The author did so by constructing Ising equivalents of the logical operators $\neg$ (not), $\wedge$ (and), and $\vee$ (or) of SAT. This serves as a reduction and thus establishes the fact that the Ising model is a member of the complexity class NP-complete.

2.3 Ising model for some NP-complete problems

We have just discussed that the computational complexity of the decision problem formulated as an Ising model is NP-complete. Therefore, there exists a polynomial-time mapping to any other NP-complete problem. We will now explain the mapping to some NP-complete problems that will be useful for mapping the HVRP.

2.3.1 Knapsack problem

The knapsack problem can be found in many different areas of optimization, ranging from logistics to finance [50, 79]. The knapsack problem with integer weights is the following. We have a set of $\mathcal{N}$ objects, labeled by i, with the weight of each object given by w_i and its value by c_i. The knapsack has limited capacity, corresponding to the maximum total weight of W. The binary decision variable $x_i \in \{0, 1\}$ denotes whether an item is contained (1) in the knapsack or not (0). The total weight of the knapsack is

$$\mathcal{W} = \sum_{i=1}^{N} w_i x_i \tag{2.2}$$

with a total value of

$$\mathcal{C} = \sum_{i=1}^{N} c_i x_i \,. \tag{2.3}$$

The NP-hard [57, 80] knapsack problem is to maximize $\mathcal{C}$ while satisfying the inequality constraint $\mathcal{W} \leq W$.

We can write an Ising formulation of the knapsack problem as follows. Let z_n for $1 \leq n \leq W$ be a binary variable which is (1) if the final weight of the knapsack is n and (0) otherwise [57]. The Hamiltonian whose energy we seek to minimize is then

$$H = H_A + H_B\,, \tag{2.4}$$

$$H_A = A\left(1 - \sum_{n=1}^{W} z_n\right)^2 + A\left(\sum_{n=1}^{W} n z_n - \sum_{i=1}^{N} w_i x_i\right)^2 , \tag{2.5}$$

$$H_B = -B\sum_{i=1}^{N} c_i x_i\,. \tag{2.6}$$

To make sure that the hard constraint is not violated, we require $0 < \max(|H_B|) < A$.

Reducing the number of auxiliary qubits

It is possible to reduce the number of variables required for the auxiliary variable z_n. We want to encode a variable which can take the values from 0 to W. Let $M \equiv \lfloor \log_2 W \rfloor$. We then require $M+1$ binary variables instead of W binary variables:

$$\sum_{n=1}^{W} n z_n \to \sum_{n=0}^{M-1} 2^n z_n + \left(W + 1 - 2^M\right) z_M\,. \tag{2.7}$$

Note that if $W \neq 2^{M+1} - 1$, degeneracies are possible [57]. Within this “log” formulation, several of the auxiliary variables can be 1, so the first part of Eq. (2.5) should not be included as this constraint enforces a one-hot encoding (exactly one element of the bitstring is one, and the rest are zero) of the bitstrings. The decision variable z_n switches from a one-hot encoding to a binary representation by using the log trick.

2.3.2 Traveling salesperson problem

The travelling salesperson problem (TSP) asks to find the shortest path between a series of cities. It consists of a graph $G = (V, E)$ with nodes representing cities and edges denoting routes between them. Edges are assigned a weight W_{ij}, which refers to the distance between two cities. The objective is to find the Hamiltonian cycle such that the sum of the weights of all edges in the cycle is minimized [57]. The Ising formulation for the TSP is given in Ref. [57] with the Hamiltonian H encoding the total cost:

$$H = H_A + H_B \,, \tag{2.8}$$

$$H_A = A \sum_{i=1}^{N} \left(1 - \sum_{\alpha=1}^{N} y_{i\alpha}\right)^2 + A \sum_{\alpha=1}^{N} \left(1 - \sum_{i=1}^{N} y_{i\alpha}\right)^2 \tag{2.9}$$

$$+ A \sum_{(i,j)\notin E} \sum_{\alpha=1}^{N} y_{i\alpha} y_{j\alpha+1} \,, \tag{2.10}$$

$$H_B = B \sum_{(i,j)\in E} W_{ij} \sum_{\alpha=1}^{N} y_{i\alpha} y_{j\alpha+1} \,, \tag{2.11}$$

$$\tag{2.12}$$

where $N = |V|$ is the number of nodes, A and B are positive constants, satisfying $0 < \max(|H_B|) < A$, and W encodes the distances between the nodes. The index i represents the nodes and α the order in a prospective cycle. The binary variables $y_{i\alpha}$ can be referred to as “routing variables” indicating in which order of the cycle node i is visited. There are N^2 variables, with $y_{i,N+1} \equiv y_{i,1}$ for all i, such that the route ends where it starts.

To use the Ising model in the context of quantum computing, we need to understand how hybrid classical-quantum algorithms for optimization problems work. We will discuss this in the next chapter.

Chapter 3

Quantum approximate optimization algorithm

This chapter is dedicated to the quantum approximate optimization algorithm (QAOA) [81]. We have discussed in Section 1.3.1 that this class of algorithms is especially well suited for the NISQ era, and thus a rigorous investigation is needed to assess their future impact.

We start by showing the similarities between two optimization algorithms, namely adiabatic quantum optimization (AQO), which we introduced as one of the first proposals for quantum computing in Section 1.2, and the QAOA, a promising candidate to achieve a quantum advantage. Further, we give an in-depth description of the algorithm and examine the interplay between a classical computer and a quantum device. Moreover, we discuss several different classical optimization algorithms. Finally, we end this chapter by extending the QAOA to the quantum alternating operator ansatz (QAOA), broadening the range of possible mixers and initial states used for the hybrid quantum-classical algorithm.

3.1 Relation to quantum adiabatic optimization

The AQO algorithm, first proposed by Farhi et al. [82], is the inspiration for the QAOA. To understand why, let us recap the primary idea underlying AQO. The AQO is based on the adiabatic theorem [83] and uses the adiabatic evolution to transition from the lowest energy eigenstate of an easy-to-prepare starting Hamiltonian H_0 to the lowest energy state of a cost Hamiltonian H_C. Two non-commuting Hamiltonians are required. This is because the adiabatic theorem holds only if there is always a gap between the eigenstates of the two Hamiltonians. If the Hamiltonians commute, they have the same set of eigenstates. Thus, $[H_0, H_1] \neq 0$ is a necessary condition for keeping the gap open. We refer to Section 1.2 for more details. Note that the cost Hamiltonian is classical and thus diagonal. The Hamiltonian is written as a sum of two non-commuting

Hamiltonians,

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_C \,, \tag{3.1}$$

The exact time evolution of the Hamiltonian is governed by Eq. (3.2). Further, we recognize that approximating the time evolution in Eq. (3.3) is the simplest way to simulate the ground state of the Hamiltonian [22],

$$U(T) \equiv \mathcal{T} \exp\left[-\mathrm{i} \int_0^T H(t) \mathrm{d}t\right] \,, \tag{3.2}$$

$$U(T) \approx \prod_{k=1}^{p} \exp[-\mathrm{i} H(k\tau)\tau] \,, \tag{3.3}$$

where $U(T)$ is the evolution operator form 0 to T, $\mathcal{T}$ is the time-ordering operator, and p is a large integer such that $\tau = T/p$ is a small time segment.

Further, for two non-commuting operators $\hat{A}$ and $\hat{B}$ and sufficiently small δ, the Trotter-Suzuki formula implies that [4]

$$e^{\delta(\hat{A}+\hat{B})} \approx e^{\delta \hat{A}} e^{\delta \hat{B}} + O\left(\delta^2\right) \,. \tag{3.4}$$

Applying Eq. (3.4) to the discretized time evolution operator Eq. (3.3) produces

$$U(T) \approx \prod_{k=1}^{p} \exp\left[-\mathrm{i}(1 - \frac{t}{T} k\tau) H_0 \tau\right] \exp\left[-\mathrm{i} \frac{t}{T} k\tau H_C \tau\right] \,. \tag{3.5}$$

As a result, by applying H_0 and H_C in an alternating sequence, we can approximate the time evolution of Eq. (3.3). Farhi, Goldstone, and Gutman [81] came up with the insight of truncating this product to an arbitrary positive integer and redefining the time dependency in each exponent with variational parameters, $\beta \equiv (1 - \frac{t}{T})\tau$ and $\gamma \equiv \frac{t}{T}\tau$,

$$U(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{k=1}^{p} \mathrm{e}^{-\mathrm{i}\beta_k H_0} \mathrm{e}^{-\mathrm{i}\gamma_k H_C} \,, \quad p \in \mathbb{N}^+ \,. \tag{3.6}$$

Further, we define H_0 to be

$$H_0 = \sum_i \sigma_i^x \,, \tag{3.7}$$

where σ_i^x is the Pauli-X matrix of the ith qubit [84]. Let the operator $U(\boldsymbol{\beta}, \boldsymbol{\gamma})$ act on the superposition state $|+\rangle$ of all possible states, and we arrived at the QAOA proposed in [81]. We see that QAOA is the trotterized version of AQO [85].

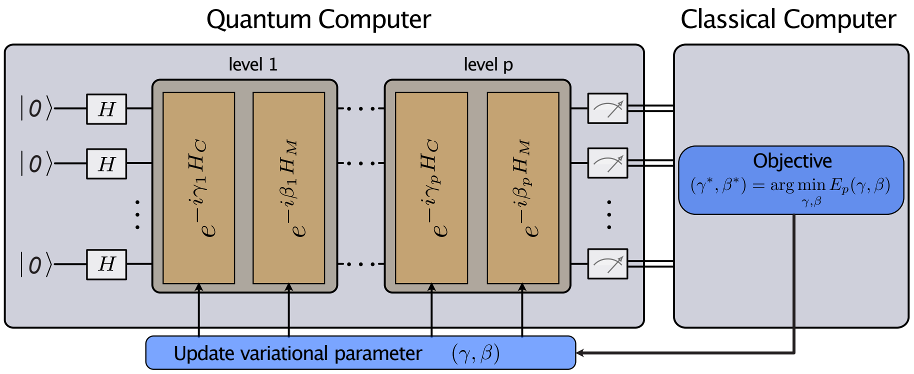

Figure 3.1: A schematic illustration of the QAOA, visualizing the interplay between the quantum device and the classical computer. The quantum computer implements a variational state formed by applying p parameterized layers of operations. Each layer has operations involving the cost Hamiltonian H_C and a mixing Hamiltonian H_M, weighted by the angles γ and β, respectively. Measurements of the variational state and calculations of its resulting energy are used to guide the classical optimizer, which minimizes the energy in a closed-loop optimization.

3.2 The interplay between a classical computer and a quantum device

We just saw how to transition from the adiabatic time evolution of two non-commuting Hamiltonians in Eq. (3.1) to an approximated expansion based on the sequential application of two Hamiltonians. We now explain how we can use this evolution operator to optimize an arbitrary optimization problem described as an Ising model that we have introduced in Section 2.

The QAOA belongs to the class of hybrid quantum-classical algorithms, combining quantum and classical processing. We illustrate the different components of the QAOA and show the building blocks that realize the quantum circuit. The closed-loop optimization of the classical and quantum devices is visualized in Fig. 3.1. It fills out the details to go from a schematic VQA introduced in Fig. 1.1 to the QAOA, a subset of VQAs able to solve combinatorial optimization problems. The quantum subroutine, operating on n qubits, consists of a consecutive application of two non-commuting operators defined as

$$U(\gamma) \equiv e^{-i\gamma H_C} \quad \gamma \in [0, 2\pi]\,, \tag{3.8}$$

$$U(\beta) \equiv e^{-i\beta H_M} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x} \quad \beta \in [0, \pi]\,. \tag{3.9}$$

The σ_j^x operation is analogous to the classical NOT gate. It changes the $|0\rangle$

state to the $|1\rangle$ state, and vice versa. The operator $U(\gamma)$ gives a phase rotation to each bitstring depending on the cost of the string, while the mixing term $U(\beta)$ scrambles the bitstrings. We call $U(\gamma)$ the phase separation operator with H_C the cost Hamiltonian and $U(\beta)$ the mixing operator with H_M the mixing Hamiltonian. The bounds for γ and β are valid if H_C has integer eigenvalues [81].

The initial state for the algorithm is a superposition of all possible computational basis states. This superposition can be obtained by first preparing the system in the initial state $|0\rangle^{\otimes n} = |00\ldots0\rangle$ for all qubits and then applying the Hadamard gate on each qubit:

$$\left(\tilde{H}\,|0\rangle\right)^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} \equiv |+\rangle^{\otimes n}\,, \tag{3.10}$$

where $\otimes$ denotes the tensor product and $\tilde{H}$ the Hadamard gate.

For any integer $p \geq 1$ and $2p$ angles $\gamma_1 \ldots \gamma_p \equiv \boldsymbol{\gamma}$ and $\beta_1 \ldots \beta_p \equiv \boldsymbol{\beta}$, we define the angle-dependent quantum state

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(\beta_p)U(\gamma_p)\ldots U(\beta_1)U(\gamma_1)\,|+\rangle^{\otimes n}\,. \tag{3.11}$$

The quantum circuit parameterized by $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ is then optimized in a closed loop using a classical optimizer. The objective is to minimize the expectation value of the cost Hamiltonian H_C [81], i.e.,

$$(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) = \underset{\boldsymbol{\gamma},\boldsymbol{\beta}}{\operatorname{argmin}}\; E(\boldsymbol{\gamma}, \boldsymbol{\beta})\,, \tag{3.12}$$

$$E(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta}|\, H_C\, |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle\;. \tag{3.13}$$

The problem of calculating the energy of $2^{\#q}$ ($\#q$ denotes the number of qubits) possible bitstrings (solutions) is thus reduced to a variational optimization over $2p$ parameters. For a detailed description, see the algorithm 1.

Algorithm 1 Quantum approximate optimization algorithm

```
Input cost Hamiltonian, $H_C$, mixer Hamiltonian, $H_M$ and layers, $p$.
Construct the circuits $U(\gamma) = e^{-i\gamma H_C}$ and $U(\beta) = e^{-i\beta H_M}$.
Build the circuit

    $U(\boldsymbol{\gamma}, \boldsymbol{\beta}) = U(\beta_p)U(\gamma_p)\ldots U(\beta_1)U(\gamma_1)\,|+\rangle^{\otimes n}$.

while Optimization (e.g. Nelder-Mead) of $\langle \boldsymbol{\gamma}, \boldsymbol{\beta}|\, H_C\, |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ do
    for number of shots do
        Sample variational quantum state, $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$
        Record measurement outcome
    end for
    Calculate $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$
    Use optimization routine to propose new $\boldsymbol{\gamma}, \boldsymbol{\beta}$ with information $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$
end while
return the final parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ and energy of the cost function $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$
```

3.3 Optimizing the variational parameters

The classical optimization method is a vital component in all hybrid quantum-classical algorithms (see Section 3.2). Moreover, the limits of the NISQ hardware place even greater dependence on robust and reliable classical optimization routines. These procedures must identify solutions in the presence of noise and scale favorably with the number of variational parameters, i.e., the running time must not increase exponentially as the number of parameters increases. In addition, the classical optimizer should be able to locate good parameter settings while limiting the number of queries it sends to the QPU. It is not surprising that classical optimizers have attracted much interest with all of these requirements. Researchers have proposed several novel optimization algorithms ranging from classical noise-resilient algorithms to machine-learning approaches. We review some of the notable developments and some general trends observed while analyzing the optimization landscape of QAOA.

A number of studies observed and examined parameter concentration, an effect in which the optimal parameters for a fixed-depth ansatz circuit continue to be optimal regardless of the problem size [55, 85–89]. Works based on machine learning (ML) detect a similar behavior showing that it is possible to extrapolate the parameters from small to large problem instances [90, 91].

Gradient-free methods: Gradient-free methods are exceptionally well suited for VQAs on NISQ devices. In Ref. [92], a set of gradient-free classical optimizers were introduced since estimating gradients from a noisy estimator is difficult. It is possible to extend the optimization problem to include optimization of the variational parameters of the quantum circuit and optimizing the prefactors of the Ising Hamiltonian [93]. For example, consider the Ising Hamiltonian of the TSP in Eq. (2.8). Here the prefactors A and B can be chosen freely as long as they satisfy the constraint $0 < \max(|H_B|) < A$. Researchers also tested well-established gradient-free optimizers, such as Nelder-Mead, Basinhopping, Cobyla, or Differential evolution [94–97]. These optimizers differ in many regards; some use global search mechanisms, e.g., Basinhopping and Differential evolution, consisting of multiple random initial guesses, while others use only a single random initial guess as a starting point for the optimization e.g., Nelder-Mead and Powell. Further, differential evolution belongs to the class of biologically inspired stochastic optimization algorithms [98]. Researchers also envision more ML-based approaches to optimize VQAs.

These ML methods are increasingly being applied to several domains, from computer games [99, 100], computer vision [101–103], and natural language processing [104, 105] to optimizing variational quantum circuits [75, 90, 91, 106–110]. As part of [90], the authors use a supervised-learning approach and propose to use a recurrent neural network to select the parameters for the optimization. An advantage of this approach is that a network can be trained on smaller, classically simulatable instances to suggest parameters for intractable cases. Moreover, several numerical studies focused on reinforcement learning (RL) methods for estimating variational parameters [91, 106–110]. Interestingly, some methods can cope with the noise constraints imposed by the hardware [62]. Gradient-free methods can be used for optimizing variational circuits to meet the noise constraints of the NISQ hardware.

Gradient-based methods: There have been proposals to apply gradient-based optimization protocols to variational quantum circuits. Optimizing an objective function via its gradient, i.e., the change of the function with respect to a shift of its P parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_P)$ is common practice. The gradient represents the direction in which the objective function shows the most significant change. This local optimization strategy starts from a given parameter value $\boldsymbol{\theta}^{(0)}$ and iteratively updates $\boldsymbol{\theta}^{(t)}$ it over several discrete steps. A common update rule for $\boldsymbol{\theta}^{(0)}$ is

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{\nabla} f(\boldsymbol{\theta})\,. \tag{3.14}$$

where η is the learning rate and

$$\partial_i \equiv \frac{\partial}{\partial \theta_i}, \quad \boldsymbol{\nabla} = (\partial_1, \dots, \partial_P)\ , \tag{3.15}$$

is the partial derivative with respect to the parameter θ_i and the gradient vector, respectively [12].

There are various ways of estimating the gradient on a quantum computer [12, 111]. One proposal is the parameter-shift rule that has been introduced by [112] and extended by [113]. McArdle [61] proposed a variational imaginary time-evolution method [110, 114–116] instead of standard gradient descent for determining the evolution of parameters. They consider a parameterized quantum circuit encoding the state $|\psi(\tau)\rangle$ as a parameterized trial state $|\psi(\boldsymbol{\theta(\tau)})\rangle$. In this way, a differential equation can be solved to obtain the evolution with respect to all the parameters. According to Stokes et al. [117], this method is analogous to gradient descent via the quantum natural gradient when minimal step size is considered. Note that the noise of the NISQ devices negatively impacts the performance of these optimization techniques [12, 111, 118, 119]. If the gradient can be estimated accurately, VQAs can benefit from gradient-based methods.

3.4 Quantum alternating operator ansatz

In Section 3.2, we introduced the QAOA, a heuristic quantum algorithm which alternates between applying two distinct unitary operators, a cost function unitary $U(\gamma) = e^{-i\gamma H_C}$ and a mixing unitary $U(\beta) = e^{-i\beta\sigma_j^x}$. An extension to these unitary matrices was introduced in [30] to include different mixers and initial states. This framework is called the quantum alternating operator ansatz (QAOA). This extension is especially interesting for optimization problems where only a small subspace of the Hilbert space contains physical solutions. We will review some of the notable extensions of QAOA and lay out properties required for the initial states and mixing Hamiltonians.

First, it is best to implement the initial state trivially. As an example, the initial state of all possible bitstrings ($|+\rangle$) can be generated in O(1) steps by applying on each qubit a Hadamard gate. We can relax this constraint and require a circuit of constant depth $[O(n)]$ to prepare the initial quantum state. Many optimization problems have specific properties, such as constant Hamming weight, i.e., the number of symbols different from the zero-symbol. Consider the example of the TSP, introduced in Section 2.3.2, where physical solutions visit each city exactly once. Thus, we must have for N cities exactly N non-zero binary variables $y_{i\alpha}$. This defines the Hamming weight as equal to the number of cities and renders much of the Hilbert space irrelevant. Using this property, we can construct an initial state consisting of a superposition of states with equal Hamming weights. It has been suggested that these states (Dicke states) can be efficiently generated using a programmable quantum computer [120].

The phase separation (see Section 3.2) unitary is left unchanged. The algorithm requires this unitary to be diagonal in the computational basis.

The mixing Hamiltonian offers many opportunities to extend beyond the x-mixer suggested in [81]. Ideally, a mixer satisfies specific properties to function best in the context of VQAs. Primarily, preserving the subspace of feasible solutions. In Section 2.3.2, we introduce a TSP example. Finding the shortest route between cities requires the bitstring solution to have a Hamming weight of N, where N is the number of cities. Therefore, it is sufficient for the mixer to only transition between states with the same Hamming distance, N. Second, the mixer provides transitions between all pairs of states corresponding to feasible points. For any pair of feasible computational-basis states $|a\rangle, |b\rangle$ there is some parameter value β and some positive integer r such that the corresponding mixer connects those two states: $|\langle \mathbf{a} | U(\beta)^r | \mathbf{b}\rangle| > 0$.

A mixer satisfying all these properties has been introduced in [29]. The authors propose an XY-mixer that restricts the space that the algorithm can act in. In a subsequent study, the authors analyze the probability of staying in such symmetry-preserved subspace under noise, providing an exact formula for local depolarizing noise [87]. The numerical results indicate that the algorithms fail to stay in the allowed subspace with noise. Further, most advanced mixers are more complex to implement [29, 30, 121] and require more gates than the x-mixer initially introduced in [81]. In conclusion, neither QAOA [81] nor the specialized XY-mixer are robust to noise for some challenging optimization problems [120].

In this chapter, we have introduced the QAOA and discussed many of its components and the connection to AQO. The next chapter presents a combinatorial optimization problem known as the HVRP. We show how to derive its Ising form that we can then use in the framework of hybrid quantum-classical algorithms.

Chapter 4

The heterogeneous vehicle routing problem and its formulation for quantum computing

As one of the most studied combinatorial optimization problems, the vehicle routing problem (VRP) deals with finding the optimal route design for a fleet of vehicles serving several customers. Since Dantzig and Ramser first formalized this problem in 1959 [122], hundreds of papers have been written discussing its many variants' exact and approximate solutions. One variant is the HVRP, in which a fleet of vehicles characterized by different capacities and costs is available for distribution activities. The HVRP was first addressed by Golden et al. [123].

This chapter is organized as follows and introduces the main contributions of the appended paper. First, we present the notation used throughout the chapter and describe the mathematical formulation of the HVRP. Then, we derive an Ising model representing the HVRP and discuss the resources required to execute problem instances on an actual programmable quantum device.

4.1 The mathematical formulation of the heterogeneous vehicle routing problem

The HVRP can be formulated as follows [124]. A fleet of vehicles is available at a depot, which becomes node 0 of a complete graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ (we do not consider multiple depots). Here, $\mathcal{N} = \{0, \ldots, n\}$ is the set of nodes or vertices, such that the n customers that the fleet of vehicles should deliver goods to constitute the customer set $\mathcal{N}_0 = \mathcal{N} \setminus \{0\}$, and $\mathcal{E} = \{(i, j) : 0 \leq i, j \leq n, i \neq j\}$

is the set of edges or arcs. Each customer i has a positive demand q_i. The set of available vehicle types is $\mathcal{V} = \{1, ..., k\}$, with m_v vehicles of type $v \in \mathcal{V}$.

When using these vehicles to deliver goods to meet customer demands, several costs and constraints need to be taken into account. First, there is the fixed vehicle cost t^v, i.e., the cost independent of the distance traveled by the vehicle of type v. This cost depends on the powertrain, trailer, engine type, and other criteria. Further, there is the vehicle capacity Q^v. Note that different vehicle types can have the same capacities, but differ in, e.g., the type of powertrain used [39]. Finally, there is the distance-dependent cost c^v_{ij} of traveling on edge (i, j) with the vehicle of type v, composed of the fuel cost and powertrain. The binary variables x^v_{ij} are equal to 1 if and only if a vehicle of type v travels on edge (i, j). Furthermore, we denote by f^v_{ij} the amount of goods that are leaving node i to go to node j using truck of type v, while the amount of goods entering the node is denoted f^v_{ji}.

Using this notation, the HVRP is to minimize the cost

$$C_{\text{tot}} = \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}_0} t^v x^v_{0j} + \sum_{v\in\mathcal{V}} \sum_{(i,j)\in\mathcal{E}} c^v_{ij} x^v_{ij} \,, \tag{4.1}$$

subject to the constraints

$$\sum_{j\in\mathcal{N}_0} x^v_{0j} \leq m_v \quad v \in \mathcal{V} \,, \tag{4.2}$$

$$\sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}} x^v_{ij} = 1 \quad i \in \mathcal{N}_0 \,, \tag{4.3}$$

$$\sum_{v\in\mathcal{V}} \sum_{i\in\mathcal{N}} x^v_{ij} = 1 \quad j \in \mathcal{N}_0 \,, \tag{4.4}$$

$$\sum_{j\in\mathcal{N}_0} x^v_{j0} = \sum_{j\in\mathcal{N}_0} x^v_{0j} \quad v \in \mathcal{V} \,, \tag{4.5}$$

$$\sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}} f^v_{ji} - \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}} f^v_{ij} = q_i \quad i \in \mathcal{N}_0 \,, \tag{4.6}$$

$$q_j x^v_{ij} \leq f^v_{ij} \leq (Q_v - q_i) x^v_{ij} \quad (i, j) \in \mathcal{E}, v \in \mathcal{V} \,, \tag{4.7}$$

$$x^v_{ij} \in \{0, 1\} \quad (i, j) \in \mathcal{E}, v \in \mathcal{V} \,, \tag{4.8}$$

$$f^v_{ij} \geq 0 \quad (i, j) \in \mathcal{E}, v \in \mathcal{V} \,. \tag{4.9}$$

The objective function in Eq. (4.1) is the sum of the fixed vehicle cost for the vehicles used to deliver goods and the total (variable) travel cost for those vehicles. The constraint in Eq. (4.2) ensures that the maximum number of available vehicles for a specific vehicle type is not exceeded. The constraints in Eqs. (4.3) and (4.4) make sure that each customer is visited exactly once, and the constraint in Eq. (4.5) establishes that all vehicles leaving the depot return to it after delivering their goods. The constraints in Eqs. (4.6) and (4.7) ensure a correct commodity flow that meets all customer demands. Finally, the constraints in Eqs. (4.8) and (4.9) enforce the variables' binary form and non-negativity restrictions.

4.2 Ising formulation of the HVRP

We can combine the Ising models introduced in Section 2.3.1 and Section 2.3.2 with our mathematical description of the HVRP to generate the Ising model of the HVRP. Formulations similar to this have been described in [125, 126]. We first transform the TSP into the VRP then add a capacity constraint inspired by the Knapsack Ising formulation.

In the original Ising formulation of the TSP, the binary variables $y_{i\alpha}$ indicate in which order of the cycle city i is visited. On the other hand, the decision variable x_{ij} used for the mathematical formulation of the HVRP is equal to 1 if and only if a vehicle travels on edge ij. To combine this formulation with the mathematical formulation of the HVRP given in Eqs. (4.1)–(4.9), we need a map from the decision variable y to x. The map we use is

$$x_{ij}^v = \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v y_{j\alpha+1}^v \,, \tag{4.10}$$

$$x_{0i}^v = y_{i1}^v + \sum_{\alpha=2}^{N_0} \left(1 - \sum_{\substack{j=1 \\ j\neq i}}^{N_0} y_{j\alpha-1}^v \right) y_{i\alpha}^v \,, \tag{4.11}$$

$$x_{i0}^v = y_{iN_0}^v + \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v \left(1 - \sum_{\substack{j=1 \\ j\neq i}}^{N_0} y_{j\alpha+1}^v \right) . \tag{4.12}$$

The summation in Eq. (4.10) is not equal to 0 if and only if i and j are subsequent stops on the same route. Equations (4.11) and (4.12) ensure that the first and last stops are automatically connected to the depot (assuming a single depot). Remember that index 0 denotes the depot, and index 1 is the first city (node) in the list of cities (nodes).

We can now write the Ising formulation for the routing problem. Let $V = |\mathcal{V}|$ be the number of trucks, where $\mathcal{V}$ now is the set of vehicles chosen for the optimization (instead of the set of vehicle *types*, as in Section 4.1), and denote by $N_0 = |\mathcal{N}_0|$ the number of customers to visit. The indices v now represent a specific truck of a specific type (instead of just a specific type, as in Section 4.1).

The Ising Hamiltonian we arrive at is then

$$H = H_A + H_B + H_C + H_D\,, \tag{4.13}$$

$$H_A = A\sum_{v=1}^{V}\sum_{i=1}^{N_0}\sum_{j=1}^{N_0} c_{ij}^v \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v y_{j\alpha+1}^v \tag{4.14}$$

$$+ A\sum_{v=1}^{V}\sum_{i=1}^{N_0} c_{0i}^v \left[y_{i1}^v + \sum_{\alpha=2}^{N_0}\left(1-\sum_{\substack{j=1\\ j\neq i}}^{N_0} y_{j\alpha-1}^v\right) y_{i\alpha}^v \right] \tag{4.15}$$

$$+ A\sum_{v=1}^{V}\sum_{i=1}^{N_0} c_{i0}^v \left[y_{iN_0}^v + \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v \left(1-\sum_{\substack{j=1\\ j\neq i}}^{N_0} y_{j\alpha+1}^v\right)\right]\,, \tag{4.16}$$

$$H_B = B\sum_{j=1}^{N_0}\sum_{v=1}^{V} t^v \sum_{\alpha=2}^{N_0}\left(1-\sum_{i=1}^{N_0} y_{i\alpha-1}^v\right) y_{j\alpha}^v\,, \tag{4.17}$$

$$H_C = C\sum_{i=1}^{N_0}\left(1-\sum_{\alpha=1}^{N_0}\sum_{v=1}^{V} y_{i\alpha}^v\right)^2\,, \tag{4.18}$$

$$H_D = D\sum_{\alpha=1}^{N_0}\left(1-\sum_{i=1}^{N_0}\sum_{v=1}^{V} y_{i\alpha}^v\right)^2\,. \tag{4.19}$$

The Hamiltonian H in Eq. (4.13) is composed of different parts. Here, H_A in Eq. (4.16) captures the first part of the original mathematical formulation, i.e., the minimization of the variable cost. The first term estimates the variable cost for traveling between the different customers/cities, while the second and third terms measure the cost of leaving and arriving at the depot. For this particular mapping, it is necessary to define the set of vehicles used for the optimization beforehand. Therefore, we can neglect the inequality constraint defined in Eq. (4.2) from the original formulation, which ensures that the number of vehicles of a specific type does not exceed the number of available vehicles. Similarly, H_B in Eq. (4.17) estimates the fixed costs of each vehicle leaving the depot [see Eq. (4.1)]. Note that the prefactors A and B must be equal in order not to rescale the relative fixed versus variable costs. The constraint given by H_C in Eq. (4.18) ensures that each city is visited exactly once. Furthermore, H_D in Eq. (4.19) guarantees that each city has a unique position in the cycle and that not more than one city can be traveled to at the same time. We require $0 < \max(H_A + H_B) < C, D$, to satisfy the constraints.

The decision variables y_{ij}^v are positioned as shown in Fig. 4.1. It helps us understand what the different constraints enforce. Consider Eq. (4.18) and Eq. (4.19), summing over all indices v and i, ensuring that each column and row contains a single non-zero element. This is important since a valid solution

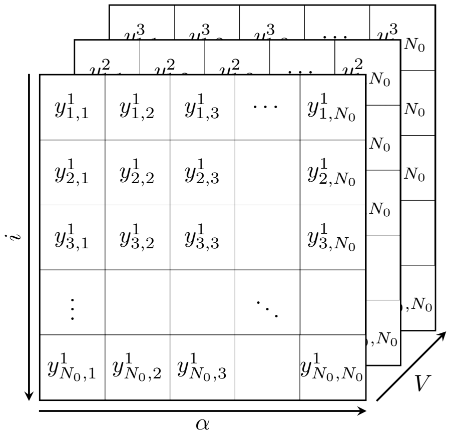

Figure 4.1: Visualization of the decision variables y^v_{ij} in the Ising formulation of the routing problem.

must visit each city exactly once. To clarify this even further, consider

$$y = \left[\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right], \tag{4.20}$$

where we show a valid configuration of binary variables encoding a solution to a simple example with two trucks over three cities. The first truck visits first the second and then the third customer ($v_1 : c_2 \to c_3$), while the second truck goes from the depot to the first customer and then back to the depot ($v_2 : c_1$). Now summing over the index v results in a matrix of size (N_0, N_0), where each column and row contains exactly a single non-zero entry, illustrating that this solution serves each customer as requested.

Next, we focus on the capacity constraints described by the HVRP. The capacity constraint is of a similar nature as the constraints for the knapsack problem — both are described by an inequality constraint, which for the knapsack problem is not to add too many items to the knapsack and for the capacity problem not to overload the vehicles. Therefore, we can use the formulation given in Ref. [57] to model the inequality constraint introduced by the capacities.

We can make use of the inequality constraint given in the knapsack formulation [see Eq. (2.5)] to encode the capacity constraints for the HVRP. Therefore, we can neglect H_B [see Eq. (2.6)] and only consider H_A [see Eq. (2.5)]. Let Q^v be the maximum capacity of vehicle v. The Hamiltonian then becomes

$$H_E = E \sum_v \left(1 - \sum_{k=0}^{Q^v} z^v_k \right)^2 + E \sum_v \left(\sum_{k=0}^{Q^v} k \cdot z^v_k - \sum_{\alpha,i} q_i y^v_{i\alpha} \right)^2 , \tag{4.21}$$

or equivalently using the log formulation,

$$H_E = E \sum_v \left(\sum_{k=0}^{M^v-1} 2^k z_k^v + (Q^v + 1 - 2^{M^v}) z_{M^v}^v - \sum_{\alpha,i} q_i y_{i\alpha}^v \right)^2 . \tag{4.22}$$

Note that the decision variable z_k^v switches from a one-hot encoding to a binary representation by using the log trick.

We can now combine the Ising model for the routing and capacities into one unifying Hamiltonian:

$$H_{\mathcal{C}} = H_A + H_B + H_C + H_D + H_E . \tag{4.23}$$

For the terms H_A to H_E, see Eqs. (4.16)–(4.19) and (4.22).

4.3 Resources

We consider three types of resources that are especially important for NISQ devices: the number of qubits, the number of two-qubit interactions, and the circuit depth.

Determining the number of qubits for solving the HVRP on a quantum computer using QAOA can be separated into two parts. The first part is given by $N_0^2 \cdot V$ and encodes the connections between customers. Secondly, auxiliary qubits are required for the constraining term H_E. The total number of qubits, $\#q$, required is

$$\#q = N_0^2 \cdot V + \sum_{v=1}^{V} \lfloor \log_2 Q^v \rfloor + 1 . \tag{4.24}$$

Further, we evaluate explicitly the resources required for three example instances visualized in Fig. 4.2. We summarize the key parameters in Table 4.1 and include the number of qubits, the number of two-qubit interactions as well as a lower bound for the circuit depth.

As a comparison, we note that modern high-performance optimizers (classical computers) for the HVRP can solve problem instances with more than 1,000 customers [127, 128]. For a quantum computer to solve problem instances of this size, it would need at least millions of controllable qubits with millions of multi-qubit interactions.

The Ising formulation for the HVRP concludes our overview of the theoretical methods and tools used in the appended paper. With the full theoretical toolbox from Chapters 2-4 in hand, we are now ready to take a closer look at the appended paper in the following chapter.

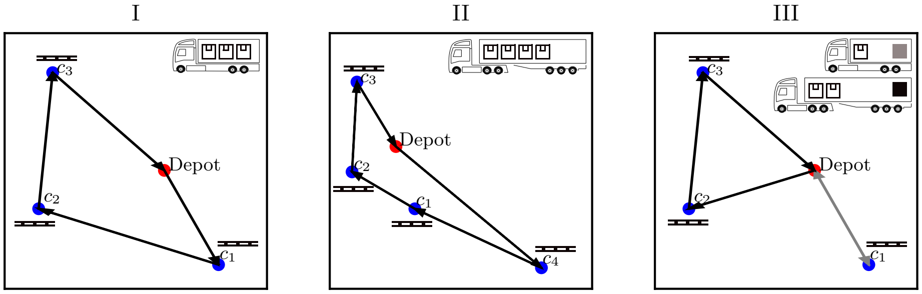

Figure 4.2: A graphical representation of three problem instances. The optimal solution is shown. Each truck carries a predefined amount of goods and brings them to the respective customers. Each pallet indicates that one item has to be moved to the customer. The crates show how many goods are carried by each truck. A color-coding indicates the route assignment for problem instance III. Arrows indicate the optimal order in which the customers are visited (the reverse order is also optimal).

Table 4.1: Information about three different problem instances. We show the values for three types of resources that are especially important for NISQ devices. These resources are the number of qubits, the number of two-qubit interactions and the circuit depth.

Problem instance	I	II	III
# cities	3	4	3
# trucks	1	1	2
# qubits for routing	9	16	18
# qubits for capacities	2	3	3
# qubits	11	19	21
# two-qubit inter.	126	288	252
Circuit depth (lower bound)	22	29	22

Chapter 5

Paper overview

In this penultimate chapter, we overview the appended paper upon which the thesis is based. The overview focuses on explaining the paper's main ideas and showing how the theoretical methods of the previous chapters are applied in practice.

5.1 Paper A - Applying quantum approximate optimization to the heterogeneous vehicle routing problem

Classical heuristics dominate the field of optimization. Now, with the emergence of quantum computing, it is imperative to investigate the impact quantum devices may have on searching for heuristic solutions. This paper focuses on the application of hybrid quantum-classical algorithms to optimization problems. In Chapter 2 and Chapter 3, we introduced the Ising model and the QAOA, respectively, the key components for our work. We focus on a specific logistics problem known as the HVRP, introduced in Chapter 4.

We develop a mapping for the HVRP to an Ising model as shown in Chapter 4, allowing us to apply the framework of hybrid quantum-classical computation to find approximate solutions. Further, we simulate the QAOA and determine its performance, i.e. its ability to find the global minimum of the optimization problem. We investigate problem instances consisting of 11, 19, and 21 qubits and use well-established optimization techniques to optimize the quantum circuit. We focus on the Nelder-Mead, basinhopping, differential evolution, and Powell optimizers.

We analyze the optimization landscape for $p = 1$ and find distinct minima for each problem instance. We also simulate the QAOA with higher p and see that with increasing circuit depth, the performance increases. However, there is a trade-off that with a growing number of variational parameters, the optimization of these variational parameters becomes increasingly difficult. This

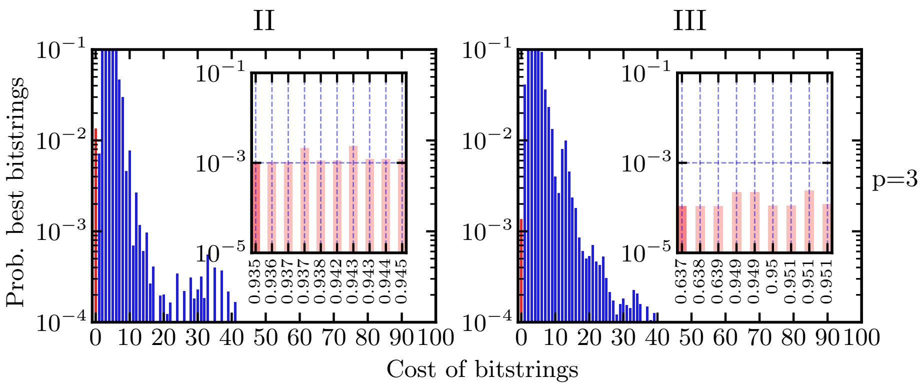

Figure 5.1: The probability distribution of the variational state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ for the 19- (left), and 21-qubit (right) problem instances introduced in Fig. 4.2 as a function of the circuit depth p for finding the best tour. The circuit depth is set to $p = 3$. The probability of sampling the best bitstrings is marked with red. The inset shows the probability for sampling any of the two best bitstrings (dark red, leftmost bin) and the probabilities for sampling any of the other feasible bitstrings (light red). The simulations were conducted with the classical optimizer Basinhopping.

can even lead to a shallower circuit reaching a lower energy state than a highly parameterized quantum circuit. We analyze the running time of the four considered optimizers and can observe an exponential scaling. This is in accordance with [63] considering that the optimization of the energy landscape is NP-hard. Both differential evolution and basinhopping give the best performance, i.e., they find the optimal solution with the highest probability.

We investigate whether the variational states generated via the hybrid quantum-classical routine generate states with a high probability of sampling the optimal bitstring. This is desired since we are looking for the best solution to the problem. In Fig. 5.1, we see the probability distribution of an optimized variational state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ for a 19- and 21-qubit instance with $p = 3$. The probability of sampling the best bitstrings is marked with red. The inset shows the probability for sampling any of the two best bitstrings (dark red, leftmost bin) and the probabilities for sampling any of the other feasible bitstrings (light red). Figure 5.1 shows that the optimization fails to distinguish between different feasible solutions. All feasible bitstrings have a similar probability of being sampled. Thus, the algorithm does not improve on sampling randomly a valid solution. Further, the resources needed for the problem instances used in the appended paper are summarized in Table 4.1. Considering the size of realistic problem instances discussed in Section 4.3, we see that this goes beyond what current NISQ devices can provide with 50 to 100 qubits.

We find that constrained optimization problems are particularly challenging for the QAOA since much work is spent on staying in the subspace of physically viable solutions. The quantum alternating operator ansatz suggests some novel mixers and initial states, but as shown in [29, 31], also these methods are sensitive to the inherent noise of NISQ devices.

Chapter 6

Conclusion

In this thesis, we study a hybrid quantum-classical algorithm, the QAOA, for application to the HVRP. We discuss the working principles of the QAOA as well as how an Ising model is used in the framework of hybrid quantum-classical algorithms to solve many different combinatorial optimization problems. Chapter 2 introduces the Ising model and its connection to many NP-complete problems. We review the different components of the QAOA and its relation to the AQO algorithm in Chapter 3. As a candidate problem to test the QAOA, we choose the HVRP. Based on its mathematical description, we construct the Ising model of the HVRP in Chapter 4.

The focus of the appended paper is to introduce a novel Ising mapping for the HVRP. Further, we simulate this mapping for toy instances and see that obtaining the optimal solution with high probability is challenging. One problem is that the energetics of the Hamiltonian is dominated by the constraints, which makes the algorithm optimize primarily to satisfy these. Resolving the finer energy scale corresponding to the actual cost thus becomes challenging. The algorithm uses most of its computational capacity to stay in the subspace of feasible solutions and thus fails to optimize for the actual problem, which is to find approximate solutions in the subspace of physical solutions.

In the context of the TSP, the subspace of physical solutions entails all the solutions that visit each city exactly once, and for the HVRP, all solutions that satisfy the constraints of visiting each customer as well as not exceeding the maximum allowed capacity of each truck. This opens up the question of whether the expectation value or energy of the variational state is a good signal for the classical optimizer to perform its work. This might be the case for unconstrained problems, but, especially in the case of constrained optimization, the system's total energy can mislead the optimization to solely focus on eliminating high-energy unphysical solutions without focusing on the actual optimization task at hand [129, 130].

Looking to the future, It would be interesting to see if a transformation exists for constrained optimization problems where the constraints are encoded directly into the Hamiltonian and not via a high energy penalty. Further, envisioning an utterly different encoding making use of existing symmetries of the HVRP could enhance its performance. For example, if we consider again the TSP problem introduced in Section 2.3.2 and focus on the form of a valid solution shown in Eq. (4.20), we clearly see that for each block of N cities, we have a single non-zero entry. Thus the parity of this block must be odd. Substituting $y_{i\alpha}$ with $y_{i\alpha}y_{i-1\alpha}$ in Eq. (2.8) transforms the Hamiltonian where every Nth qubit will be constant when measured and can thus be removed from the circuit. Therefore, we can reduce the number of qubits but with the drawback of more complicated multi-qubit interactions. Possibly there are better ways of representing the HVRP. Much exciting work aims to investigate and manipulate the energy landscape of VQAs [5, 131, 132]. This simplifies the optimization procedure and can significantly boost the algorithm's performance.

We have used state-of-the-art hybrid quantum-classical algorithms to improve approximations of classical optimization problems. Still, we are only beginning to realize the potential of this research field.

Bibliography

[1] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines", Journal of Statistical Physics **22**, 563–591 (1980).

[2] D. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer", Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **400**, 97–117 (1985).

[3] R. P. Feynman, "Simulating physics with computers", International Journal of Theoretical Physics **21**, 467–488 (1982).

[4] S. Lloyd, "Universal Quantum Simulators", Science **273**, 1073–1078 (1996).

[5] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, "Exploring Entanglement and Optimization within the Hamiltonian Variational Ansatz", PRX Quantum **1**, 020319 (2020).

[6] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, "The Variational Quantum Eigensolver: A review of methods and best practices", arXiv:2111.05176 (2021).

[7] C. Cade, L. Mineh, A. Montanaro, and S. Stanisic, "Strategies for solving the Fermi-Hubbard model on near-term quantum computers", Physical Review B **102**, 235122 (2020).

[8] C.-Y. Park, "Efficient ground state preparation in variational quantum eigensolver with symmetry breaking layers", arXiv:2106.02509 (2021).

[9] P. Shor, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (IEEE Comput. Soc. Press, 1994) pp. 124–134.

[10] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The Impact of Quantum Computing on Present Cryptography", International Journal of Advanced Computer Science and Applications **9**, 405–414 (2018).

[11] M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia, in *Advances in Cryptology – CRYPTO 2016*, edited by M. Robshaw and J. Katz (Springer Berlin Heidelberg, Berlin, Heidelberg, 2016) pp. 207–237.

[12] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, "Noisy intermediate-scale quantum algorithms", Reviews of Modern Physics **94**, 015004 (2022).

[13] A. Montanaro, "Quantum algorithms: An overview", npj Quantum Information **2**, 15023 (2016).

[14] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum Algorithm for Linear Systems of Equations", Physical Review Letters **103**, 150502 (2009).

[15] M. Schuld and F. Petruccione, "Quantum ensembles of quantum classifiers", Scientific Reports **8**, 2772 (2018).

[16] V. Saggio, B. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. Harris, M. Hochberg, D. Englund, S. Wölk, H. Briegel, and P. Walther, in *Quantum Nanophotonic Materials, Devices, and Systems 2021*, edited by M. Agio, C. Soci, and M. T. Sheldon (SPIE, San Diego, United States, 2021) p. 18.

[17] C. Blank, D. K. Park, J. K. K. Rhee, and F. Petruccione, "Quantum classifier with tailored quantum kernel", npj Quantum Information **6**, 41 (2020).

[18] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces", Nature **567**, 209–212 (2019).

[19] M. Schuld and N. Killoran, "Quantum Machine Learning in Feature Hilbert Spaces", Physical Review Letters **122**, 040504 (2019).

[20] V. Dunjko, J. M. Taylor, and H. J. Briegel, in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE, Banff, AB, 2017) pp. 282–287.

[21] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers", Physical Review A **101**, 032308 (2020).

[22] Y. Sun, J.-Y. Zhang, M. S. Byrd, and L.-A. Wu, "Adiabatic Quantum Simulation Using Trotterization", arXiv:1805.11568 (2018).

[23] D. Amaro, M. Rosenkranz, N. Fitzpatrick, K. Hirano, and M. Fiorentini, "A case study of variational quantum algorithms for a job shop scheduling problem", arXiv:2109.03745 (2021).

[24] Q. Gao, J. M. Garcia, N. Yamamoto, H. Nakamura, T. P. Gujarati, G. O. Jones, J. E. Rice, S. P. Wood, M. Pistoia, and N. Yamamoto, "Computational investigations of the lithium superoxide dimer rearrangement on noisy quantum devices", Journal of Physical Chemistry A **125**, 1827–1836 (2021).

[25] K. Meichanetzidis, S. Gogioso, G. de Felice, N. Chiappori, A. Toumi, and B. Coecke, "Quantum Natural Language Processing on Near-Term Quantum Computers", Electronic Proceedings in Theoretical Computer Science **340**, 213–229 (2021).

[26] E. R. Miranda, R. Yeung, A. Pearson, K. Meichanetzidis, and B. Coecke, "A Quantum Natural Language Processing Approach to Musical Intelligence", arXiv:2111.06741 (2021).

[27] B. Coecke, G. de Felice, K. Meichanetzidis, and A. Toumi, "Foundations for Near-Term Quantum Natural Language Processing", arXiv:2012.03755 (2020).

[28] D. Kartsaklis, I. Fan, R. Yeung, A. Pearson, R. Lorenz, A. Toumi, G. de Felice, K. Meichanetzidis, S. Clark, and B. Coecke, "Lambeq: An Efficient High-Level Python Library for Quantum NLP", arXiv:2110.04236 (2021).

[29] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, "XY mixers: Analytical and numerical results for the quantum alternating operator ansatz", Physical Review A **101**, 012320 (2020).

[30] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz", Algorithms **12**, 34 (2019).

[31] M. Streif, M. Leib, F. Wudarski, E. Rieffel, and Z. Wang, "Quantum algorithms with local particle-number conservation: Noise effects and error correction", Physical Review A **103**, 042412 (2021).

[32] S. Yarkoni, A. Alekseyenko, M. Streif, D. Von Dollen, F. Neukart, and T. Bäck, "Multi-car paint shop optimization with quantum annealing", arXiv:2109.07876 (2021).

[33] G. Laporte, S. Ropke, and T. Vidal, in *Vehicle Routing*, September (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014) pp. 87–116.

[34] B. Golden, S. Raghavan, and E. Wasil, "The vehicle routing problem: Latest advances and new challenges", Operations Research/ Computer Science Interfaces Series **43**, 3–27 (2008).

[35] M. Lombardi and M. Milano, "Optimal Methods for Resource Allocation and Scheduling: A Cross-Disciplinary Survey" (2010).

[36] A. Van Breedam, “Improvement heuristics for the Vehicle Routing Problem based on simulated annealing”, European Journal of Operational Research **86**, 480–490 (1995).

[37] C.-P. Hwang, B. Alidaee, and J. D. Johnson, “A tour construction heuristic for the travelling salesman problem”, Journal of the Operational Research Society **50**, 797–809 (1999).

[38] Y. Huang, B. Chen, W. Lu, Z. X. Jin, and R. Zheng, in *Advances in Intelligent Systems and Computing*, Vol. 991 (2020) pp. 1067–1077.

[39] T. Ghandriz, B. Jacobson, M. Islam, J. Hellgren, and L. Laine, “Transportation-Mission-Based Optimization of Heterogeneous Heavy-Vehicle Fleet Including Electrified Propulsion”, Energies **14**, 3221 (2021).

[40] M. Agrawal, N. Kayal, and N. Saxena, “PRIMES is in P”, Annals of Mathematics **160**, 781–793 (2004).

[41] R. B. Laughlin and D. Pines, “The Theory of Everything”, Proceedings of the National Academy of Sciences **97**, 28–31 (2000).

[42] A. Trabesinger, “Quantum simulation”, Nature Physics **8**, 263 (2012).

[43] A. W. Harrow and A. Montanaro, “Quantum computational supremacy”, Nature **549**, 203–209 (2017).

[44] A. P. Lund, M. J. Bremner, and T. C. Ralph, “Quantum sampling problems, BosonSampling and quantum supremacy”, npj Quantum Information **3**, 15 (2017).

[45] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, “Quantum computational advantage using photons”, Science **370**, 1460–1463 (2020).

[46] F. Arute *et al.*, “Quantum supremacy using a programmable superconducting processor.”, Nature **574**, 505–510 (2019).

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

[48] Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning”, Nature Physics **17**, 1013–1017 (2021).

[49] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, “Option Pricing using Quantum Computers”, Quantum **4**, 291 (2020).

[50] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum Computing for Finance: State of the Art and Future Prospects", IEEE Transactions on Quantum Engineering **1**, 3101724 (2020).

[51] S. Mensa, E. Sahin, F. Tacchino, P. K. Barkoutsos, and I. Tavernelli, "Quantum Machine Learning Framework for Virtual Screening in Drug Discovery: A Prospective Quantum Advantage", arXiv:2204.04017 (2022).

[52] M. G. de Andoin, E. Osaba, I. Oregi, E. Villar-Rodriguez, and M. Sanz, "Hybrid Quantum-Classical Heuristic for the Bin Packing Problem", arXiv:2204.05637 (2022).

[53] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges", Applied Physics Reviews **6**, 021314 (2019).

[54] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting Qubits: Current State of Play", Annual Review of Condensed Matter Physics **11**, 369–395 (2020).

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

[56] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation", SIAM Review **50**, 755–787 (2008).

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

[58] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC '96* (ACM Press, Philadelphia, Pennsylvania, United States, 1996) pp. 212–219.

[59] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms", Nature Reviews Physics **3**, 625–644 (2021).

[60] V. E. Elfving, B. W. Broer, M. Webber, J. Gavartin, M. D. Halls, K. P. Lorton, and A. Bochevarov, "How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?" (2020).

[61] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, "Variational ansatz-based quantum simulation of imaginary time evolution", npj Quantum Information **5**, 75 (2019).

[62] J. Yao, M. Bukov, and L. Lin, "Policy Gradient based Quantum Approximate Optimization Algorithm", Proceedings of Machine Learning Research **107**, 1–30 (2020).

[63] L. Bittel and M. Kliesch, “Training Variational Quantum Algorithms Is NP-Hard”, Physical Review Letters **127**, 120502 (2021).

[64] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors”, Nature **323**, 533–536 (1986).

[65] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Barren plateaus in quantum neural network training landscapes”, Nature Communications **9**, 4812 (2018).

[66] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, “Cost function dependent barren plateaus in shallow parametrized quantum circuits”, Nature Communications **12**, 1791 (2021).

[67] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, “Barren plateaus preclude learning scramblers”, Physical Review Letters **126**, 190501 (2021).

[68] S. Thanasilp, S. Wang, N. A. Nghiem, P. J. Coles, and M. Cerezo, “Subtleties in the trainability of quantum machine learning models”, arXiv:2110.14753 (2021).

[69] S. G. Brush, “History of the Lenz-Ising Model”, Reviews of Modern Physics **39**, 883–893 (1967).

[70] V. Bapst, L. Foini, F. Krzakala, G. Semerjian, and F. Zamponi, “The Quantum Adiabatic Algorithm applied to random optimization problems: The quantum spin glass perspective”, Physics Reports **523**, 127–205 (2013).

[71] Z. Bian, F. Chudak, W. Macready, and G. Rose, in *D-Wave Systems* (2010) pp. 1–32.

[72] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation.*, Introduction to the Theory of Neural Computation. (Addison-Wesley/Addison Wesley Longman, Reading, MA, US, 1991) pp. xxii, 327.

[73] P. Mehta and D. J. Schwab, “An exact mapping between the Variational Renormalization Group and Deep Learning”, arXiv:1410.3831 (2014).

[74] H. Matsuda, “The Ising Model for Population Biology”, Progress of Theoretical Physics **66**, 1078–1080 (1981).

[75] M. Alam, A. Ash-Saki, and S. Ghosh, in *Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020* (2020) pp. 686–689.

[76] S. A. Cook, in *Proceedings of the Third Annual ACM Symposium on Theory of Computing - STOC ’71* (ACM Press, Shaker Heights, Ohio, United States, 1971) pp. 151–158.

[77] L. A. Levin, “Universal sequential search problems”, Problemy peredachi informatsii **9**, 115–116 (1973).

[78] F. Barahona, “On the computational complexity of Ising spin glass models”, Journal of Physics A: Mathematical and General **15**, 3241–3253 (1982).

[79] H. Al-Khazraji, S. Khlil, and Z. Alabacy, “Industrial Picking and Packing Problem: Logistic Management for Products Expedition”, Journal of Mechanical Engineering Research and Developments **43**, 74–80 (2020).

[80] R. Andonov and S. Rajopadhye, “Unbounded Knapsack Problem: Dynamic Programming Revisited” (1997).

[81] E. Farhi, J. Goldstone, and S. Gutmann, “A Quantum Approximate Optimization Algorithm” (2014).

[82] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, “Quantum Computation by Adiabatic Evolution”, arXiv:quant-ph/0001106 (2000).

[83] M. Born and V. Fock, “Beweis des Adiabatensatzes”, Zeitschrift für Physik **51**, 165–180 (1928).

[84] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Vol. 66 (Cambridge University Press, Cambridge, 2010).

[85] P. Vikstål, *Application of the Quantum Approximate Optimization Algorithm to Combinatorial Optimization Problems*, Licenciate Thesis, Chalmers University of Technology (2020).

[86] 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).

[87] 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).

[88] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, “Parameter Concentration in Quantum Approximate Optimization”, (2021), arXiv:2103.11976 .

[89] 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 (2021).

[90] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, “Learning to learn with quantum neural networks via classical neural networks”, arXiv:1907.05415 (2019).

[91] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Learning to Optimize Variational Quantum Circuits to Solve Combinatorial Problems", Proceedings of the AAAI Conference on Artificial Intelligence **34**, 2367–2375 (2020).

[92] W. Lavrijsen, A. Tudor, J. Muller, C. Iancu, and W. De Jong, "Classical Optimizers for Noisy Intermediate-Scale Quantum Devices", Proceedings - IEEE International Conference on Quantum Computing and Engineering, QCE 2020 , 267–277 (2020).

[93] C. Roch, A. Impertro, T. Phan, T. Gabor, S. Feld, and C. Linnhoff-Popien, "Cross entropy hyperparameter optimization for constrained problem hamiltonians applied to QaoA", Proceedings - 2020 International Conference on Rebooting Computing, ICRC 2020 , 50–57 (2020).

[94] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization", The Computer Journal **7**, 308–313 (1965).

[95] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives", The Computer Journal **7**, 155–162 (1964).

[96] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization **11**, 341–359 (1996).

[97] D. Wales and J. Doye, "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms", The Journal of Physical Chemistry A **101**, 5111–5116 (1998).

[98] M. Wahde, "Biologically inspired optimization methods: An introduction", Choice Reviews Online **46**, 46-3899-46-3899 (2009).

[99] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning", Nature **572**, 350–354 (2019).

[100] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver, "Mastering Atari, Go, chess and shogi by planning with a learned model", Nature **588**, 604–609 (2020).

[101] L. Ding, J. Terwilliger, R. Sherony, B. Reimer, and L. Fridman, "Value of Temporal Dynamics Information in Driving Scene Segmentation", arXiv:1904.00758 (2019).

[102] A. Ghimire, N. Werghi, S. Javed, and J. Dias, "Real-Time Face Recognition System", arXiv:2204.08978 (2022).

[103] J. Terwilliger, M. Glazer, H. Schmidt, J. Domeyer, H. Toyoda, B. Mehler, B. Reimer, and L. Fridman, "Dynamics of Pedestrian Crossing Decisions Based on Vehicle Trajectories in Large-Scale Simulated and Real-World Data", arXiv:1904.04202 (2019).

[104] T. Brown *et al.*, in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 1877–1901.

[105] M. Chen *et al.*, "Evaluating Large Language Models Trained on Code", (2021), arXiv:2107.03374 .

[106] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro, "Reinforcement-learning-assisted quantum optimization", Physical Review Research **2**, 033446 (2020).

[107] A. Garcia-Saez and J. Riu, "Quantum Observables for continuous control of the Quantum Approximate Optimization Algorithm via Reinforcement Learning", arXiv:1911.09682 (2019).

[108] D. Beloborodov, A. E. Ulanov, J. N. Foerster, S. Whiteson, and A. I. Lvovsky, "Reinforcement learning enhanced quantum-inspired algorithm for combinatorial optimization", Machine Learning: Science and Technology **2**, 025009 (2021).

[109] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, "Universal quantum control through deep reinforcement learning", npj Quantum Information **5**, 33 (2019).

[110] C. Cao, Z. An, S.-Y. Hou, D. L. Zhou, and B. Zeng, "Quantum imaginary time evolution steered by reinforcement learning", Communications Physics **5**, 57 (2022).

[111] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, "General parameter-shift rules for quantum gradients", Quantum **6**, 677 (2022).

[112] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning", Physical Review A **98**, 032309 (2018).

[113] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware", Physical Review A **99**, 032331 (2019).

[114] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. Benjamin, "Theory of variational quantum simulation", Quantum **3**, 191 (2019).

[115] H. Nishi, T. Kosugi, and Y.-i. Matsushita, "Implementation of quantum imaginary-time evolution method on NISQ devices by introducing nonlocal approximation", npj Quantum Information **7**, 85 (2021).

[116] S.-H. Lin, R. Dilip, A. G. Green, A. Smith, and F. Pollmann, "Real- and imaginary-time evolution with compressed quantum circuits", PRX Quantum **2**, 010342 (2021).

[117] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, "Quantum Natural Gradient", Quantum **4**, 269 (2020).

[118] G. E. Crooks, "Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition", arXiv:1905.13311 (2019).

[119] G. Boyd and B. Koczor, "Training variational quantum circuits with CoVaR: Covariance root finding with classical shadows", arXiv:2204.08494 (2022).

[120] A. Bärtschi and S. Eidenbenz, in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11651 LNCS (2019) pp. 126–139.

[121] A. Bartschi and S. Eidenbenz, "Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation", Proceedings - IEEE International Conference on Quantum Computing and Engineering, QCE 2020 , 72–82 (2020), arXiv:2006.00354 .

[122] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem", Management Science **6**, 80–91 (1959).

[123] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem", Computers & Operations Research **11**, 49–66 (1984).

[124] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "Thirty years of heterogeneous vehicle routing", European Journal of Operational Research **249**, 1–21 (2016).

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

[126] R. Harikrishnakumar, S. Nannapaneni, N. H. Nguyen, J. E. Steck, and E. C. Behrman, "A Quantum Annealing Approach for Dynamic Multi-Depot Capacitated Vehicle Routing Problem", arXiv:2005.12478 (2020).

[127] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian, "New benchmark instances for the Capacitated Vehicle Routing Problem", European Journal of Operational Research **257**, 845–858 (2017).

[128] S. E. Barman, P. Lindroth, and A.-B. Strömberg, in *Optimization, Control, and Applications in the Information Age*, edited by A. Migdalas and A. Karakitsiou (Springer International Publishing, Cham, 2015) pp. 113–138.

[129] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, "Benchmarking the quantum approximate optimization algorithm", Quantum Information Processing **19**, 197 (2020).

[130] M. P. Harrigan *et al.*, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor", Nature Physics **17**, 332–336 (2021).

[131] J. Lee, A. B. Magann, H. A. Rabitz, and C. Arenz, "Progress toward favorable landscapes in quantum combinatorial optimization", Physical Review A **104**, 032401 (2021).

[132] T. Fujii, K. Komuro, Y. Okudaira, R. Narita, and M. Sawada, "Energy landscape transformation of Ising problem with invariant eigenvalues for quantum annealing", arXiv:2202.05927 (2022).

Paper A

Applying quantum approximate optimization to the heterogeneous vehicle routing problem
David Fitzek, Toheed Ghandriz, Leo Laine, Mats Granath, Anton Frisk Kockum
ArXiv:2110.06799 (2021)

Applying quantum approximate optimization to the heterogeneous vehicle routing problem

David Fitzek,[1,2,*] Toheed Ghandriz,[2,3] Leo Laine,[2,3] Mats Granath,[4,†] and Anton Frisk Kockum[1,‡]

[1] *Department of Microtechnology and Nanoscience,*
Chalmers University of Technology, 412 96 Gothenburg, Sweden
[2] *Volvo Group Trucks Technology, 405 08 Gothenburg, Sweden*
[3] *Department of Mechanics and Maritime Sciences,*
Chalmers University of Technology, 412 96 Gothenburg, Sweden
[4] *Department of Physics, University of Gothenburg, 412 96 Gothenburg, Sweden*

Quantum computing offers new heuristics for combinatorial problems. With small- and intermediate-scale quantum devices becoming available, it is possible to implement and test these heuristics on small-size problems. A candidate for such combinatorial problems is the heterogeneous vehicle routing problem (HVRP): the problem of finding the optimal set of routes, given a heterogeneous fleet of vehicles with varying loading capacities, to deliver goods to a given set of customers. In this work, we investigate the potential use of a quantum computer to find approximate solutions to the HVRP using the quantum approximate optimization algorithm (QAOA). For this purpose we formulate a mapping of the HVRP to an Ising Hamiltonian and simulate the algorithm on problem instances of up to 21 qubits. We find that the number of qubits needed for this mapping scales quadratically with the number of customers. We compare the performance of different classical optimizers in the QAOA for varying problem size of the HVRP, finding a trade-off between optimizer performance and runtime.

I. INTRODUCTION

Devices utilizing quantum-mechanical effects provide a new computational paradigm that enables novel algorithms and heuristics [1–5]. The ongoing development of such devices [6–9] provides an opportunity to test these algorithms on small problem instances, which could lead to new solutions to hard optimization problems. In this work, we show how a quantum approximate optimization algorithm (QAOA) [10] can be employed to find approximate solutions for the heterogeneous vehicle routing problem (HVRP) [11]. Our approach can be utilized on both noisy intermediate-scale quantum (NISQ) [12] computers and quantum annealers [13]. It also paves the way for implementing challenging instances of the HVRP on larger quantum computers in the future.

The HVRP belongs to the well known and extensively studied class of optimization problems known as the vehicle routing problem (VRP) [14] in the field of logistics. The VRP captures the problem of finding the optimal set of routes for a fleet of vehicles to travel in order to deliver goods to a given set of customers. This problem is also found in supply-chain management and scheduling [15]. Variants of the VRP include the capacitated vehicle routing problem (CVRP), in which the vehicles have a limited carrying capacity [14], and the HVRP studied here, in which the fleet composition is unknown and capacity constraints are given [11, 16]. All these VRPs are very challenging since they belong to the complexity class NP-hard [17].

Due to its industrial relevance, there has been tremendous effort devoted to finding good approximate solutions to the VRP and its variants through various heuristics [18, 19], e.g., construction heuristics [20], improvement heuristics [21], and metaheuristic top-level strategies [22]. In construction heuristics, e.g., the Clarke and Wright saving algorithm [23], one starts from an empty solution and iteratively extends it until a complete solution is obtained. In improvement heuristics, one instead starts from a complete solution (often generated by a construction heuristic) and then try to improve further through local moves. There are several software libraries and tools that implement ready-to-use solvers for all these methods [24–26]. Moreover, exact methods for solving the VRP and its variants have also been investigated [27].

In this article, we instead investigate a heuristic method for solving the HVRP on a quantum computer. Such devices, including both programmable quantum processors [4, 8, 28] and quantum annealers [13], are gradually becoming available due to the recent advances in controlling quantum systems. The current quantum computers are known as NISQ devices [12], since they are largely limited by their intermediate number (several tens [6, 29–33]) of controllable qubits, limited connectivity, imperfect qubit control, short coherence times, and minimal error correction. Only a subset of known quantum algorithms can run on these near-term devices [5, 34]; other algorithms require more advanced hardware.

The heuristic method we apply to the HVRP here is an example of a variational quantum algorithm (VQA) [35], which is a promising class of quantum algorithms that are compatible with NISQ devices. These algorithms generally need access to a description of the problem, and also

[*] davidfi@chalmers.se
[†] mats.granath@physics.gu.se
[‡] anton.frisk.kockum@chalmers.se

possibly to a set of training data. With this in hand, the first step is to define a cost (or loss) function C, which encodes the solution to the problem. Next, one proposes an ansatz, i.e., a quantum operation depending on a set of continuous or discrete parameters $\boldsymbol{\theta}$ that can be optimized. This ansatz is then optimized in a hybrid quantum-classical loop to solve the optimization task

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ C(\boldsymbol{\theta}). \tag{1}$$

Such algorithms have emerged as a leading contender for obtaining quantum advantage [35] within the constraints of NISQ devices. By now, variational quantum algorithms (VQAs) have been proposed for numerous applications that researchers have envisioned for quantum computers, e.g., in chemistry, logistics, and finance [36–40].

The type of VQA we employ here is the QAOA [10], which is a heuristic that can approximate the solution to many combinatorial problems, including VRPs. Current research in this area ranges from applications on large-scale VRP instances with a quantum annealer [41] to more specific variants of the VRP, such as the CVRP [42] or the multi-depot capacitated VRP [43]. These approximation algorithms have been tested on quantum annealers [42] and NISQ devices [44]. There have also been several experimental realizations of the QAOA applied to other optimization problems [45–49].

However, a problem description suited for the QAOA, an Ising Hamiltonian [50–52] (describing the energy of interacting two-level systems), seems to be lacking for the case of the HVRP. In this work, we provide such a mapping for the HVRP, which can be utilized on both NISQ computers and quantum annealers. We show that, in this formulation, the number of qubits scales quadratically with the number of customers. To explore the performance of the QAOA applied to the HVRP, we simulate problem instances with up to 21 qubits. We check how this performance depends both on the choice of classical optimizer and on the depth of the quantum circuit. This work lays the foundation for finding approximate solutions to large problem instances of the HVRP when sufficiently advanced quantum-computing hardware becomes available.

The paper is organized as follows. In Sec. II, we introduce the HVRP and its mathematical formulation. Then, in Sec. III, we develop the Ising formulation of the HVRP. In Sec. IV, we review the QAOA and describe how it can be used to find approximate solutions to the HVRP. In Sec. V, we present numerical results from applying the QAOA to a few HVRPs of different sizes. Finally, we conclude the paper and give an outlook for future work in Sec. VI.

II. THE HETEROGENEOUS VEHICLE ROUTING PROBLEM

The HVRP can be formulated as follows [11]. A fleet of vehicles is available at a depot, which becomes node 0 of a complete graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ (we do not consider multiple depots). Here, $\mathcal{N} = \{0, ..., n\}$ is the set of nodes or vertices, such that the n customers that the fleet of vehicles should deliver goods to constitute the customer set $\mathcal{N}_0 = \mathcal{N} \setminus \{0\}$, and $\mathcal{E} = \{(i, j) : 0 \leq i, j \leq n, i \neq j\}$ denotes the set of edges or arcs. Each customer i has a positive demand q_i.

The set of available vehicle types is $\mathcal{V} = \{1, ..., k\}$, with m_v vehicles of type $v \in \mathcal{V}$. When using these vehicles to deliver goods to meet the customer demand, there are several costs and constraints that need to be taken into account. First, there is the fixed vehicle cost t^v, i.e., the cost that is independent of the distance travelled by the vehicle of type v. Then, there is the vehicle capacity Q^v. Note that different vehicle types can have the same capacities, but differ in, e.g., the type of powertrain used [16]. Finally, there is the cost c_{ij}^v of travelling on edge (i, j) with the vehicle of type v. To describe all constraints, it is also useful to introduce the binary variables x_{ij}^v, which are equal to 1 if and only if a vehicle of type v travels on edge (i, j). Furthermore, we denote by f_{ij}^v the amount of goods that are leaving node i to go to node j using truck v, while the amount of goods entering the node is denoted f_{ji}^v.

Using this notation, the HVRP is to minimize the cost

$$C_{\text{tot}} = \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}_0} t^v x_{0j}^v + \sum_{v\in\mathcal{V}} \sum_{(i,j)\in\mathcal{E}} c_{ij}^v x_{ij}^v\,, \tag{2}$$

subject to the constraints

$$\sum_{j\in\mathcal{N}_0} x_{0j}^v \leq m_v \quad v \in \mathcal{V}\,, \tag{3}$$

$$\sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}} x_{ij}^v = 1 \quad i \in \mathcal{N}_0\,, \tag{4}$$

$$\sum_{v\in\mathcal{V}} \sum_{i\in\mathcal{N}} x_{ij}^v = 1 \quad j \in \mathcal{N}_0\,, \tag{5}$$

$$\sum_{j\in\mathcal{N}_0} x_{j0}^v = \sum_{j\in\mathcal{N}_0} x_{0j}^v \quad v \in \mathcal{V}\,, \tag{6}$$

$$\sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}} f_{ji}^v - \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{N}} f_{ij}^v = q_i \quad i \in \mathcal{N}_0\,, \tag{7}$$

$$q_j x_{ij}^v \leq f_{ij}^v \leq (Q_v - q_i) x_{ij}^v \quad (i, j) \in \mathcal{E}, v \in \mathcal{V}\,, \tag{8}$$

$$x_{ij}^v \in \{0, 1\} \quad (i, j) \in \mathcal{E}, v \in \mathcal{V}\,, \tag{9}$$

$$f_{ij}^v \geq 0 \quad (i, j) \in \mathcal{E}, v \in \mathcal{V}\,. \tag{10}$$

The objective function in Eq. (2) is the sum of the fixed vehicle cost for the vehicles used to deliver goods and the total (variable) travel cost for those vehicles. The constraint in Eq. (3) ensures that the maximum number of available vehicles for a specific vehicle type is not exceeded. The constraints in Eqs. (4) and (5) make sure that each customer is visited exactly once, and the constraint in Eq. (6) sees to that all vehicles leaving the depot return to it after delivering their goods. The constraints in Eqs. (7) and (8) ensure a correct commodity flow that meets all customer demands. Finally, the constraints in Eqs. (9) and (10) enforce the binary form and non-negativity restrictions on the variables.

III. ISING FORMULATION FOR THE HVRP

All optimization problems in the complexity class NP can be reformulated as the problem of finding the ground state (lowest-energy configuration) of a quantum Hamiltonian [52]. This is also the method we use for the HVRP in this work. Since the HVRP combines two distinct problems, a routing problem and a capacity problem, we have to derive an Ising Hamiltonian that captures both these problems simultaneously.

A. Routing problem

For the routing problem, we start from the travelling salesperson problem (TSP) formulation given in Ref. [52] with the Hamiltonian H encoding the total cost:

$$H = H_A + H_B \,, \tag{11}$$

$$H_A = A\sum_{i=1}^{N}\left(1-\sum_{\alpha=1}^{N} y_{i\alpha}\right)^2 + A\sum_{\alpha=1}^{N}\left(1-\sum_{i=1}^{N} y_{i\alpha}\right)^2 + A\sum_{(i,j)\notin\mathcal{E}}\sum_{\alpha=1}^{N} y_{i\alpha}y_{j\alpha+1}\,, \tag{12}$$

$$H_B = B\sum_{(i,j)\in\mathcal{E}} W_{ij}\sum_{\alpha=1}^{N} y_{i\alpha}y_{j\alpha+1}\,, \tag{13}$$

where $N = |\mathcal{N}|$ is the number of nodes including the depot, A and B are positive constants, and W encodes the distances between the nodes. The index i represents the nodes and α the order in a prospective cycle. The binary variables $y_{i\alpha}$ can be referred to as 'routing variables' indicating in which order of the cycle node i is visited. There are N^2 variables, with $y_{i,N+1} \equiv y_{i,1}$ for all i, such that the route ends where it starts. The last term in Eq. (12), which ensures that non-existent edges are not used, can be neglected for the problem we investigate because we assume a fully connected graph.

To combine this formulation with the mathematical formulation of the HVRP given in Eqs. (2)–(10), we need a map from the decision variable y to x. The map we use is

$$x_{ij}^v = \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v y_{j\alpha+1}^v\,, \tag{14}$$

$$x_{0i}^v = y_{i1}^v + \sum_{\alpha=2}^{N_0}\left(1-\sum_{\substack{j=1\\ j\neq i}}^{N_0} y_{j\alpha-1}^v\right) y_{i\alpha}^v\,, \tag{15}$$

$$x_{i0}^v = y_{iN_0}^v + \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v\left(1-\sum_{\substack{j=1\\ j\neq i}}^{N_0} y_{j\alpha+1}^v\right). \tag{16}$$

The summation in Eq. (14) is not equal 0 if and only if i and j are subsequent stops on the same route. Equations (15) and (16) ensure that the first and last stops are automatically connected to the depot (assuming a single depot). Remember that the index 0 denotes the depot and the index 1 the first city (node) in the list of cities (nodes).

We can now write the Ising formulation for the routing problem. Here we extend the formulation compared to previous works to capture different types of trucks, not just multiple trucks of the same type (having the same capacity) [52, 53]. Let $V = |\mathcal{V}|$ be the number of trucks, where $\mathcal{V}$ now is the set of vehicles chosen for the optimization (instead of the set of vehicle *types*, as in Section II), and denote by $N_0 = |\mathcal{N}_0|$ the number of customers to visit. The indices v now represent a specific truck of a specific type (instead of just a specific type, as in Section II). The Ising Hamiltonian we derive is then

$$H = H_A + H_B + H_C + H_D\,, \tag{17}$$

$$\begin{aligned} H_A = {} & A\sum_{v=1}^{V}\sum_{i=1}^{N_0}\sum_{j=1}^{N_0} c_{ij}^v \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v y_{j\alpha+1}^v \\ & + A\sum_{v=1}^{V}\sum_{i=1}^{N_0} c_{0i}^v\left[y_{i1}^v + \sum_{\alpha=2}^{N_0}\left(1-\sum_{\substack{j=1\\ j\neq i}}^{N_0} y_{j\alpha-1}^v\right) y_{i\alpha}^v\right] \\ & + A\sum_{v=1}^{V}\sum_{i=1}^{N_0} c_{i0}^v\left[y_{iN_0}^v + \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v\left(1-\sum_{\substack{j=1\\ j\neq i}}^{N_0} y_{j\alpha+1}^v\right)\right], \end{aligned} \tag{18}$$

$$H_B = B\sum_{j=1}^{N_0}\sum_{v=1}^{V} t^v \sum_{\alpha=2}^{N_0}\left(1-\sum_{i=1}^{N_0} y_{i\alpha-1}^v\right) y_{j\alpha}^v\,, \tag{19}$$

$$H_C = C\sum_{i=1}^{N_0}\left(1-\sum_{\alpha=1}^{N_0}\sum_{v=1}^{V} y_{i\alpha}^v\right)^2, \tag{20}$$

$$H_D = D\sum_{\alpha=1}^{N_0}\left(1-\sum_{i=1}^{N_0}\sum_{v=1}^{V} y_{i\alpha}^v\right)^2. \tag{21}$$

The Hamiltonian H in Eq. (17) is composed of different parts. Here, H_A in Eq. (18) captures the first part

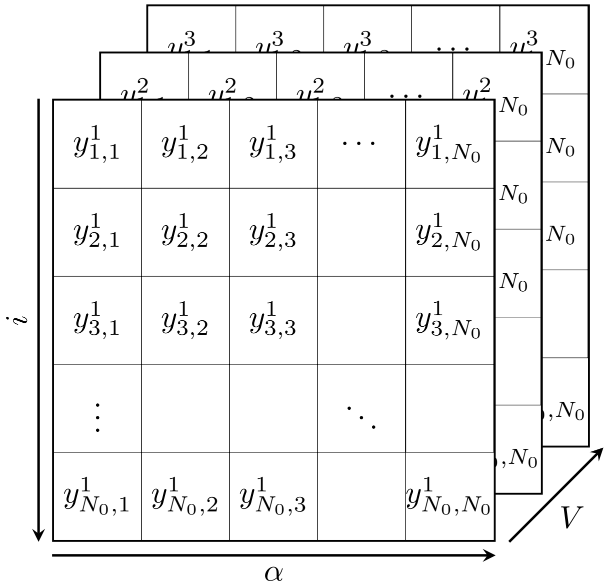

FIG. 1. Visualization of the decision variables y^v_{ij} in the Ising formulation of the routing problem.

of the original mathematical formulation, i.e., the minimization of the variable cost. The first term estimates the variable cost for traveling between the different customers/cities, while the second and third terms measure the cost of leaving and arriving at the depot. For this particular mapping it is necessary to define the set of vehicles that are used for the optimization beforehand. Therefore, we can neglect the inequality constraint defined in Eq. (3) from the original formulation, which ensures that the number of vehicles of a specific type does not exceed the number of available vehicles. Similarly, H_B in Eq. (19) estimates the fixed costs of each vehicle leaving the depot [see Eq. (2)]. Note that the prefactors A and B must be equal, in order not to rescale the relative fixed versus variable costs. The constraint given by H_C in Eq. (20) ensures that each city is visited exactly once. Furthermore, H_D in Eq. (21) ensures that each city has a unique position in the cycle and that not more than one city can be travelled to at the same time. To make sure that the constraints are not violated, we require $0 < \max(H_A + H_B) < C, D$.

The decision variables y^v_{ij} are positioned as shown in Fig. 1. This picture allows us to see the operations that are taking place when summing over specific indices. As an example, consider Eq. (20). First summing over the indices v and α corresponds to a summation over these two axes. After the summation it is easy to see that if the goal is to visit each customer/city once, then each element of the length N_0 array must be one.

One notable technicality about the formulation is that certain solutions that may be considered valid are excluded by the constraint in Eq. (21). However, the excluded solutions are physically equivalent to some allowed solution, as illustrated by the following simple example

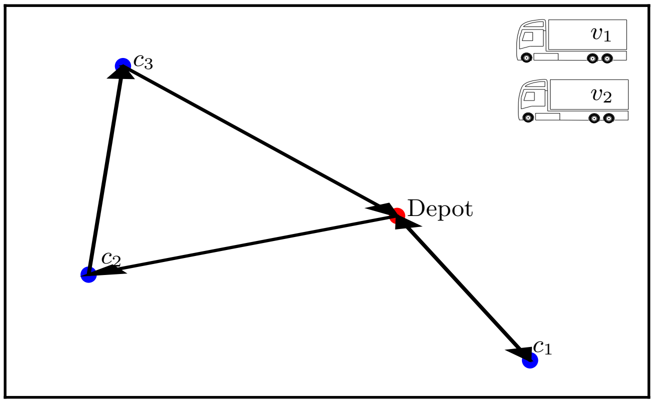

FIG. 2. Visualization of a problem instance with a suggested solution. The first truck v_1 visits city c_2 and the city c_3 before returning to the depot. The second truck v_2 only visits city c_1.

(see Fig. 2) with two trucks over three cities

$$y = \left[\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right], \tag{22}$$

and

$$y = \left[\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right]. \tag{23}$$

In both cases, the two different matrices describe the routes of the two different vehicles and the order in which they visit the different customers c_i. They both represent a physically valid solution where the first truck visits first the second and then the third customer ($v_1 : c_2 \to c_3$), while the second truck goes from the depot to the first customer and then back to the depot ($v_2 : c_1$). The constraint in Eq. (21), however, rules out the latter representation as it has two non-zero entries for $\alpha = 2$. If we want to allow this larger set of viable representations, physically equivalent to solutions already allowed by Eq. (21), we can replace that constraint by a reformulated one,

$$H'_D = D \sum_{v=1}^{V} \sum_{\alpha=1}^{N_0} \left(u^v_\alpha - \sum_{i=1}^{N_0} y^v_{i\alpha} \right)^2, \tag{24}$$

where we have introduced N_0^2 additional auxiliary qubits u^v_α. Especially in the NISQ era, where quantum resources are scarce, it is important to encode the problem with as few qubits as possible. Thus we do not consider Eq. (24) a viable route for implementations, but use Eq. (21) for the simulations in Section V.

B. Capacity problem

The capacity constraint is of a similar nature as the constraints for the knapsack problem — both are described by an inequality constraint, which for the knapsack problem is to not add too many items to the knapsack and for the trucks to not overload the vehicles. Therefore, we can use the formulation given in Ref. [52] to model the inequality constraint introduced by the capacities.

The knapsack problem with integer weights is the following. We have a list of N objects, labeled by i, with the weight of each object given by w_i and its value by c_i. The knapsack has a limited capacity of W. The binary decision variable x_i denotes whether an item is contained (1) in the knapsack or not (0). The total weight of the knapsack is

$$\mathcal{W} = \sum_{i=1}^{N} w_i x_i \tag{25}$$

with a total value of

$$\mathcal{C} = \sum_{i=1}^{N} c_i x_i \,. \tag{26}$$

The NP-hard knapsack problem is to maximize $\mathcal{C}$ while satisfying the inequality constraint $\mathcal{W} \leq W$.

We can write an Ising formulation of the knapsack problem as follows. Let z_n for $1 \leq n \leq W$ be a binary variable which is 1 if the final weight of the knapsack is n and 0 otherwise [52]. The Hamiltonian whose energy we seek to minimize is then

$$H = H_A + H_B \,, \tag{27}$$

$$H_A = A\left(1 - \sum_{n=1}^{W} z_n\right)^2 + A\left(\sum_{n=1}^{W} n z_n - \sum_{i=1}^{N} w_i x_i\right)^2 , \tag{28}$$

$$H_B = -B \sum_{i=1}^{N} c_i x_i \,. \tag{29}$$

To make sure that the hard constraint is not violated, we require $0 < \max(|H_B|) < A$.

1. *Reducing the number of auxiliary qubits*

It is possible to reduce the number of variables required for the auxiliary variable z_n. We want to encode a variable which can take the values from 0 to W. Let $M \equiv \lfloor \log_2(N) \rfloor$. We then require $M+1$ binary variables instead of N binary variables:

$$\sum_{n=1}^{N} n z_n \rightarrow \sum_{n=0}^{M-1} 2^n z_n + \left(N + 1 - 2^M\right) z_M \,. \tag{30}$$

Note that if $N \neq 2^{M+1} - 1$, degeneracies are possible [52]. Within this log formulation, several of the auxiliary variables can be 1, so the first part of Eq. (28) should not be included as this constraint enforces a one-hot encoding (exactly one element of the bitstring is one and the rest are zero) of the bitstrings.

We can make use of the inequality constraint given in the knapsack formulation [see Eq. (28)] to encode the capacity constraints for the HVRP. Therefore, we can neglect H_B [see Eq. (29)] and only consider H_A [see Eq. (28)]. Let Q^v be the maximum capacity of vehicle v. The Hamiltonian then becomes

$$H_A = A \sum_{v} \left(1 - \sum_{k=0}^{Q^v} z_k^v\right)^2 + A \sum_{v} \left(\sum_{k=0}^{Q^v} k \cdot z_k^v - \sum_{\alpha,i} q_i y_{i\alpha}^v\right)^2 , \tag{31}$$

or equivalently using the log formulation,

$$H_A = A \sum_{v} \left(\sum_{k=0}^{M^v-1} 2^k z_k^v + (Q^v + 1 - 2^{M^v}) z_{M^v}^v - \sum_{\alpha,i} q_i y_{i\alpha}^v\right)^2 . \tag{32}$$

Note that by using the log trick, the decision variable z_k^v switches from a one-hot encoding to a binary representation.

C. The full Ising Hamiltonian for the HVRP

We are now ready to write down the full Hamiltonian for the HVRP. The full Ising Hamiltonian $H_{\mathcal{C}}$ contains five terms, where the first term H_A captures the actual optimization problem and the other terms are penalty terms to ensure that invalid configurations are penalized with a high energy:

$$H_{\mathcal{C}} = H_A + H_B + H_C + H_D + H_E \,, \tag{33}$$

$$H_E = E \sum_{v=1}^{V} \left(\sum_{k=0}^{M^v-1} 2^k z_k^v + (Q^v + 1 - 2^{M^v}) z_{M^v}^v - \sum_{\alpha=1}^{N_0} \sum_{i=1}^{N_0} q_i y_{i\alpha}^v\right)^2 \tag{34}$$

For the terms H_A to H_D, see Eqs. (18)–(21).

The formulation presented in this paper combines the capacity problem and the routing problem in one Ising Hamiltonian. Similarly, a unified approach is also attempted in Ref. [42] with the difference that the authors add a constraint for clustering the customers as well. Here, by using the decision variables $y_{i\alpha}^v$ that indicate the position in a prospective cycle instead of x^v that is 1 if and only if a vehicle traverses from customer i to j, we circumvent the subtour-elimination constraint. This constraint needs to loop through all possible subtours as it is presented in Ref. [43]. Moreover, a solution obtained with our mapping does not necessarily use all the vehicles that are available. It can find the most cost efficient subset of vehicles needed to solve the task.

D. Resources

The required resources (qubits) for solving the HVRP with our approach can be separated into three parts. The first part comes from encoding the connections between the customers and scales with $N_0^2 \cdot V$. Additionally, auxiliary qubits are required for the constraining term H_E. The overall number of qubits, $\#q$, required are

$$\#q = N_0^2 \cdot V + \underbrace{\sum_{v=1}^{V} \lfloor \log_2 Q^v \rfloor + 1}_{H_E} \,. \tag{35}$$

Using the alternative formulation for H_D [see Eq. (24)] adds more auxiliary qubits ($N_0 \cdot V$), yielding

$$\#q = (N_0 + 1) N_0 V + \sum_{v=1}^{V} \lfloor \log_2 Q^v \rfloor + 1 \,. \tag{36}$$

As a comparison, we note that modern high-performance optimizers (classical computers) for the HVRP can solve problem instances with more than 1,000 customers [54, 55]. For a quantum computer to solve problem instances of this size, it would need at least millions of controllable qubits. Systems of this size are likely still several years away from being realized.

IV. THE QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

The QAOA belongs to the class of hybrid quantum-classical algorithms, which combine quantum and classical processing. The closed-loop optimization of the classical and quantum devices is visualized in Fig. 3. The quantum subroutine, operating on n qubits, consists of a consecutive application of two non-commuting operators defined as [10]

$$U(\gamma) \equiv e^{-i\gamma H_C} \qquad \gamma \in [0, 2\pi] \,, \tag{37}$$

$$U(\beta) \equiv e^{-i\beta H_M} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x} \qquad \beta \in [0, \pi] \,, \tag{38}$$

where σ^x denotes the Pauli X matrix [56]. This operation is analogous to the classical NOT gate. It changes the $|0\rangle$ state to the $|1\rangle$ state, and vice versa. The operator $U(\gamma)$ gives a phase rotation to each bit string depending on the cost of the string, while the mixing term $U(\beta)$ scrambles the bit strings. We call H_C the cost Hamiltonian and H_M the mixing Hamiltonian. The bounds for γ and β are valid if H_C has integer eigenvalues [10]. The formulation of H_C for the HVRP is given by Eq. (33).

TABLE I. Information about the three different problem instances used in simulations.

Problem instance	I	II	III
Number of cities	3	4	3
Number of trucks	1	1	2
Number of qubits for routing	9	16	18
Number of qubits for capacities	2	3	3
Total number of qubits	11	19	21

The initial state for the algorithm is a superposition of all possible computational basis states. This superposition can be obtained by first preparing the system in the initial state $|0\rangle^{\otimes n} = |00\ldots0\rangle$ for all qubits and then applying the Hadamard gate on each qubit:

$$\left(\tilde{H}\,|0\rangle\right)^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} \equiv |+\rangle^{\otimes n} \,, \tag{39}$$

where $\otimes$ denotes the tensor product and $\tilde{H}$ the Hadamard gate.

For any integer $p \geq 1$ and $2p$ angles $\gamma_1 \ldots \gamma_p \equiv \boldsymbol{\gamma}$ and $\beta_1 \ldots \beta_p \equiv \boldsymbol{\beta}$, we define the angle-dependent quantum state

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(\beta_p) U(\gamma_p) \ldots U(\beta_1) U(\gamma_1) \, |+\rangle^{\otimes n} \,. \tag{40}$$

The quantum circuit parametrized by $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ is then optimized in a closed loop using a classical optimizer. The objective is to minimize the expectation value of the cost Hamiltonian H_C [10], i.e.,

$$(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) = \underset{\boldsymbol{\gamma}, \boldsymbol{\beta}}{\operatorname{argmin}} \; E(\boldsymbol{\gamma}, \boldsymbol{\beta}) \,, \tag{41}$$

$$E(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | \, H_C \, | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle \,. \tag{42}$$

The problem of calculating the energy of $2^{\#q}$ possible bit strings (solutions) is thus reduced to a variational optimization over $2p$ parameters.

V. BENCHMARKING QUANTUM APPROXIMATE OPTIMIZATION FOR THE HETEROGENEOUS VEHICLE ROUTING PROBLEM

In this section, we show numerical results from noise-free simulations of the QAOA applied to the HVRP. We examine three different problem instances, labelled I, II, and III, which use 11, 19, and 21 qubits, respectively. Table I contains information about the number of cities, available trucks, and the overall number of qubits needed to encode these problem instances in an Ising Hamiltonian using the scheme we have described in Section III. For the simulations, we consider realistic fuel consumption, gas prices, and fixed costs for each truck type, as detailed in Appendix A. A graphical representation of the problem instances is shown in Fig. 4.

For the simulations we consider two different approaches. One is to only solve for satisfying the constraints. The other is to solve the full problem, optimizing not only for feasible solutions, but for the best

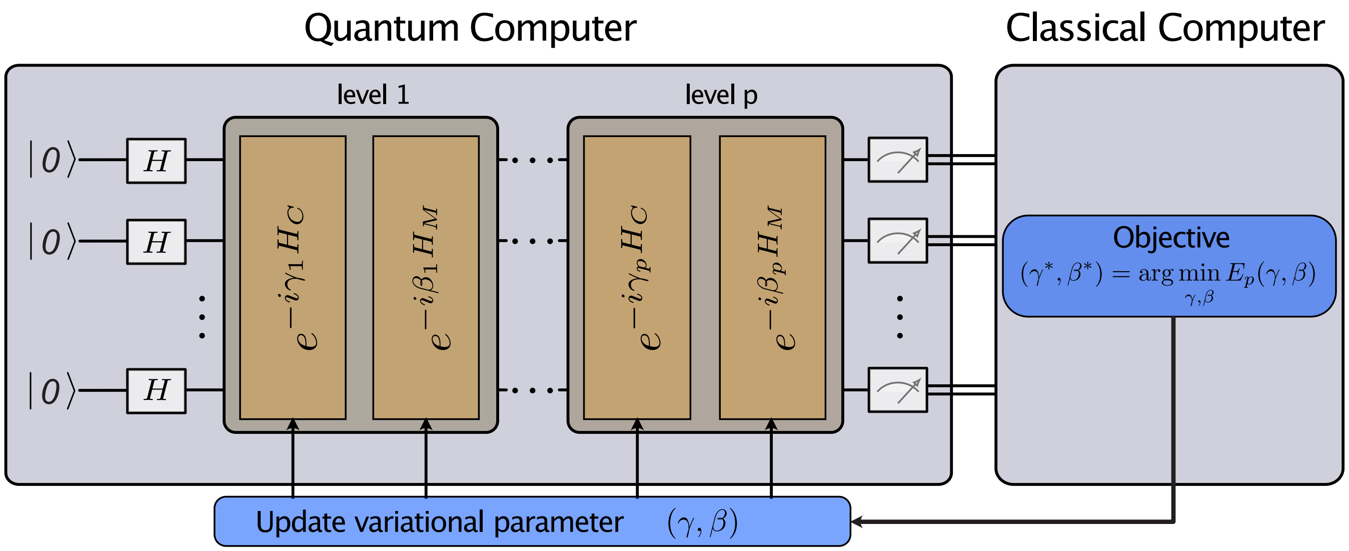

FIG. 3. A schematic description of the QAOA, visualizing the interplay between the quantum device and the classical computer. The quantum computer implements a variational state formed by applying p parameterized layers of operations. Each layer has operations involving the cost Hamiltonian H_C and a mixing Hamiltonian H_M, weighted by the angles γ and β, respectively. Measurements of the variational state and calculations of its resulting energy are used to guide the classical optimizer, which minimizes the energy in a closed-loop optimization.

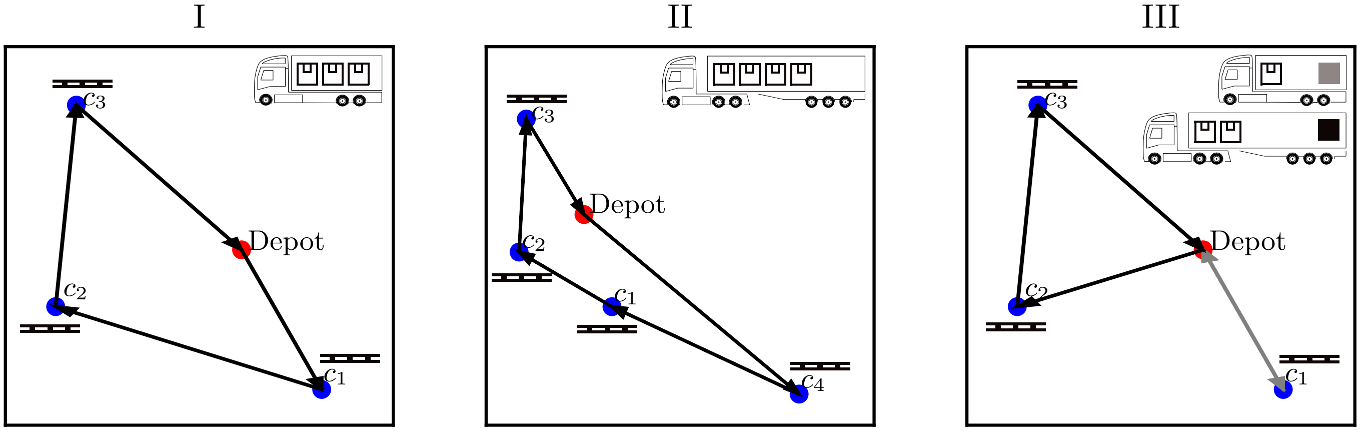

FIG. 4. The three problem instances and some solutions for them obtained with low-depth QAOA. A visual representation, with correct scaling, of the three problem instances that are considered for the simulations. The optimal solution is shown. Each truck carries a predefined amount of goods and brings it to the respective customers. The pallets indicate that one item has to be carried to the customer. The crates show how much goods is carried by each truck. A color coding indicates the route assignment for problem instance III. The optimal order in which the customers are visited is indicated by arrows (the reverse order is also optimal).

solution. This stepwise approach, starting with the constraints [Eqs. (20), (21), and (34)] and then including the optimization part [(Eqs. (18)–(19)], helps us understand whether some parts of the full problem contribute more to its difficulty than others.

For the first approach, finding satisfying solutions, we neglect the actual optimization part of the problem, i.e., minimizing the cost of the routing for the solution. We set the prefactors of the different parts of the Hamiltonian [Eqs. (20), (21), and (34)] to 1. The eigenvalues of the Hamiltonian are integers. This allows us to restrict the search space for β and γ in Eqs. (37)–(38) to $[0, \pi]$ and $[0, 2\pi]$, respectively. For the full problem, we cannot make use of this simplification.

The second approach is to solve the full problem with the optimization included [Eqs. (18)–(21) and Eq. (34)]. Additionally, we rescale the cost function of the optimization [Eqs. (18)–(19)] such that it only takes values between 0 and 1. Note that for rescaling the cost, we have to evaluate the cost for each possible solution, which makes it necessary to brute-force the problem. This is only feasible for small problem instances. In a real-world application, this rescaling procedure cannot be applied and therefore it might be necessary to optimize over the penalty weights as well [53, 57]. The eigenvalues are not integer values anymore and therefore the entire search space for the variational parameters must be explored.

A. Energy landscape

To illustrate the difficulty of finding good variational parameters for the QAOA, we show in Fig. 5 the energy landscape for $p = 1$ for each problem instance considering the full problem [Fig. 5(a)] and the capacity constraint in isolation [Fig. 5(b)]. We evaluate the expectation value $E(\gamma, \beta)$ of the cost Hamiltonian on a grid $\{\gamma, \beta\} \in [0, 2\pi] \times [0, \pi]$. Note that for the full problem, the entire space of variational parameters must be considered, but for this visualization we constrain it to the range mentioned.

The states with the lowest energy are marked with dark blue in Fig. 5. Each surface plot shows four distinct optima. Moreover, the variational parameters are concentrated in the same region for all three problem instances, both for the full problem and for only the capacity constraint. The range for optimal parameters narrows with increasing problem size. Similar behaviour has been observed in Ref. [58]. The overall shape of the energy landscape does not change significantly with varying problem size, but the overall expectation value increases. This is not surprising, since with increasing problem size there are many more constraints to satisfy, goods to deliver, and trucks to choose from.

As discussed in Section III, the HVRP consists of two problems, a routing problem and a capacity problem. The capacity problem is analogous to the constraints of the knapsack problem [52]. To better understand the energy landscapes for the capacity part of the three problem instances shown in Fig. 5(b), we now take a closer look at the landscape of a particular knapsack problem. The problem is: given a knapsack with a maximum capacity of 5, choose from the list of items $[4, 3, 2, 1]$ the ones that satisfy the capacity constraint. Seven qubits are needed to encode the problem, making use of the log trick introduced in Section III B 1.

Figure 6 shows the energy landscape for this knapsack problem. The plot shows a rapidly oscillating energy landscape. It is clear that many optimizers will struggle to find the global optimum in this landscape. We argue that with increasing complexity of the problem instances for the HVRP, maneuvering the landscape of the capacity constraint becomes a difficult problem. In Fig. 5(b) we do not observe this behaviour yet, but this is simply due to the fact that the problem instances we consider are small (see Table I). To obtain a landscape that is easier to handle for the classical optimizer, it might be necessary to relax the knapsack constraint or to find a different formulation to encode the capacity constraint [59].

B. Increasing the circuit depth

It has been shown that for a circuit depth of $p = 1$, the QAOA cannot outperform classical optimization algorithms [10, 60]. For actual applications of the QAOA, it is therefore necessary to go to a circuit depth beyond $p = 1$.

Before we investigate $p > 1$, we start with the lowest possible circuit depth of $p = 1$. In Fig. 7(a) and Fig. 7(b), we show a histogram of the probability distribution created by the variational circuit for finding solutions satisfying all constraints, in the cases where the cost function encodes only the constraints and the full problem, respectively. We show the probability of sampling bitstrings with a specific cost. Note that there can be several bitstrings leading to a particular cost. The probability of sampling any of the feasible bitstrings is shown in red and the rest of the optimized distribution is depicted in blue. As a comparison, we show the probability distribution for a variational state in the $|+\rangle$ state, meaning all bitstrings are sampled with uniform probability (orange). The difference between these two distributions is marginal. Thus, adding the cost terms [Eqs. (18)–(19)] to the optimization problem does not alter the overall performance of the algorithm when it comes to satisfying the constraints. This is perhaps not so surprising when considering that due to the rescaling of the cost Hamiltonian, the costs not associated with constraints impact the overall shape of the energy landscape less.

The optimized probability distribution is shifted to the left compared to the random distribution. Thus, sampling from the optimized variational state gives solutions with overall lower energy compared to random sampling of bitstrings. Moreover, for the small problem instance with 11 qubits, it is possible to sample valid bitstrings with a probability of approximately 3 %. For the 19- and 21-qubit problem instances, the probability of sampling a valid bitstring is about an order of magnitude less. This is not surprising since we are limiting the algorithm to a shallow circuit depth.

Next, we go to a circuit depth above 1. Here we consider in Fig. 8 and Fig. 9 finding feasible solutions and the best solutions, respectively. We shows how the variational state changes with increasing circuit depth from $p = 1$ to $p = 5$ for the three problem instances [see Fig. 4]. Additionally, we show for Fig. 8 an inset focusing on the low-energy part of the distribution of the variational state and for Fig. 9 an inset focusing on all feasible bitstrings. Again, the bitstrings we are aiming to sample are marked in red and the rest of the optimized distribution is depicted in blue.

For the 11-qubit instance, the probability of sampling a valid bitstring reaches values up to 18 % for $p = 5$. For the slightly larger instances (19 and 21 qubits), the probability of sampling a valid bitstring does not exceed 5 % (see Fig. 8). For the more difficult problem of actually finding the optimal tour (not just a feasible tour), the probability drops to 9 % for the 11-qubit instance,

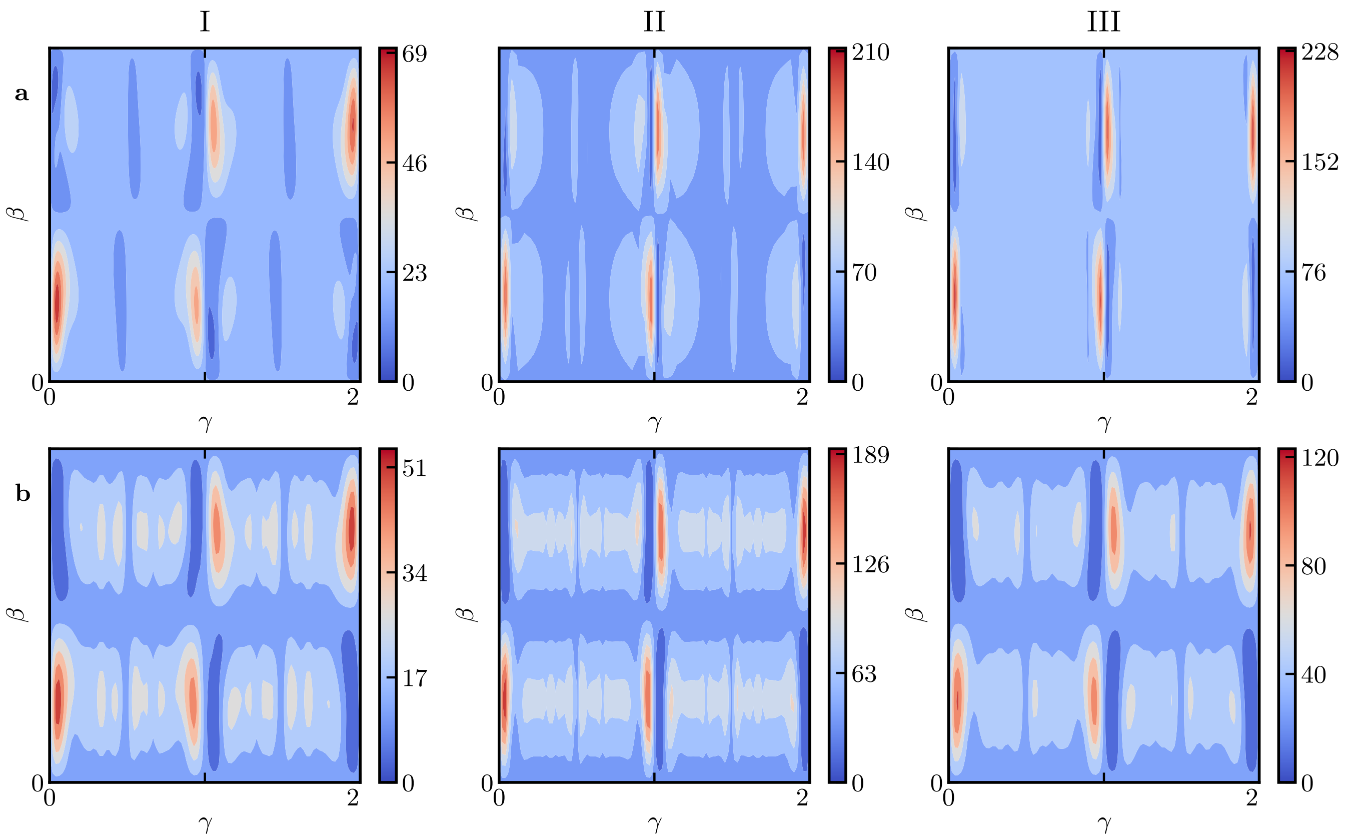

FIG. 5. Energy landscapes for the three problem instances with circuit depth $p = 1$. (a) The energy landscape for the full HVRP [Eqs. (18)–(21) and Eq. (34)]. The expectation value $E(\gamma, \beta)$ for the total cost depends on the classically optimized angles γ and β. The periodicity is broken due to the non-integer eigenvalues for the cost Hamiltonian. (b) The energy landscape for the capacity constraint only [Eq. (34)]. Here, the expectation value $E(\gamma, \beta)$ describes the energy penalty for breaking the capacity constraint.

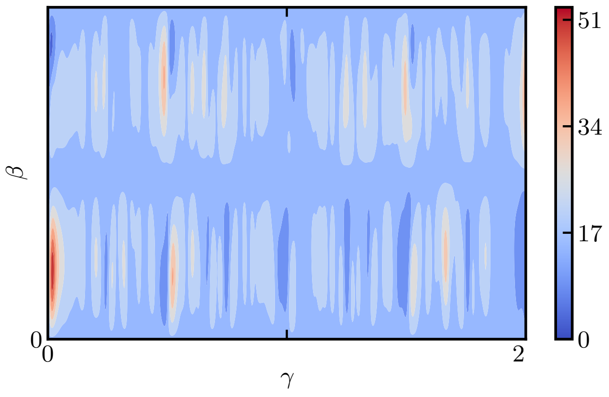

FIG. 6. The energy landscape for the knapsack problem [see Eq. (28)] discussed in the main text with a circuit depth of $p = 1$. The energy landscape is highly non-convex and finding the global optimum is difficult for a classical optimizer.

while for the 19- and 21-qubit instances the probability of sampling the ideal bitstring is below 1 % (see Fig. 9).

The inset in Fig. 9 shows that the algorithm cannot distinguish between the different feasible solutions and thus fails to optimize for the best solution. This might be due to the very small energy gap between the lowest and next lowest energy eigenstate, which is an artifact of the rescaling of the cost we introduced earlier. The point of this rescaling is to ensure that all feasible solutions have lower energy than any solution violating any constraint. Rigorous hyperparameter optimization might be necessary to weight the cost and the constraints to circumvent the problems introduced by the chosen rescaling [53].

The probability of sampling bitstrings with low cost increases with increasing circuit depth, meaning that the probability distribution shifts to lower-energy eigenstates. This trend is visible in Fig. 8 and Fig. 9 as the probability distribution shifts increasingly to the left with increasing depth. It might be possible with increasing circuit depth to obtain higher probabilities of sampling optimal solutions at the expense of optimizing more variational parameters. This is in accordance with the theory of adiabatic quantum optimization (AQO) — the performance of the algorithm becomes better with more variational parameters. The drawback is that the optimization problem becomes increasingly difficult and time-

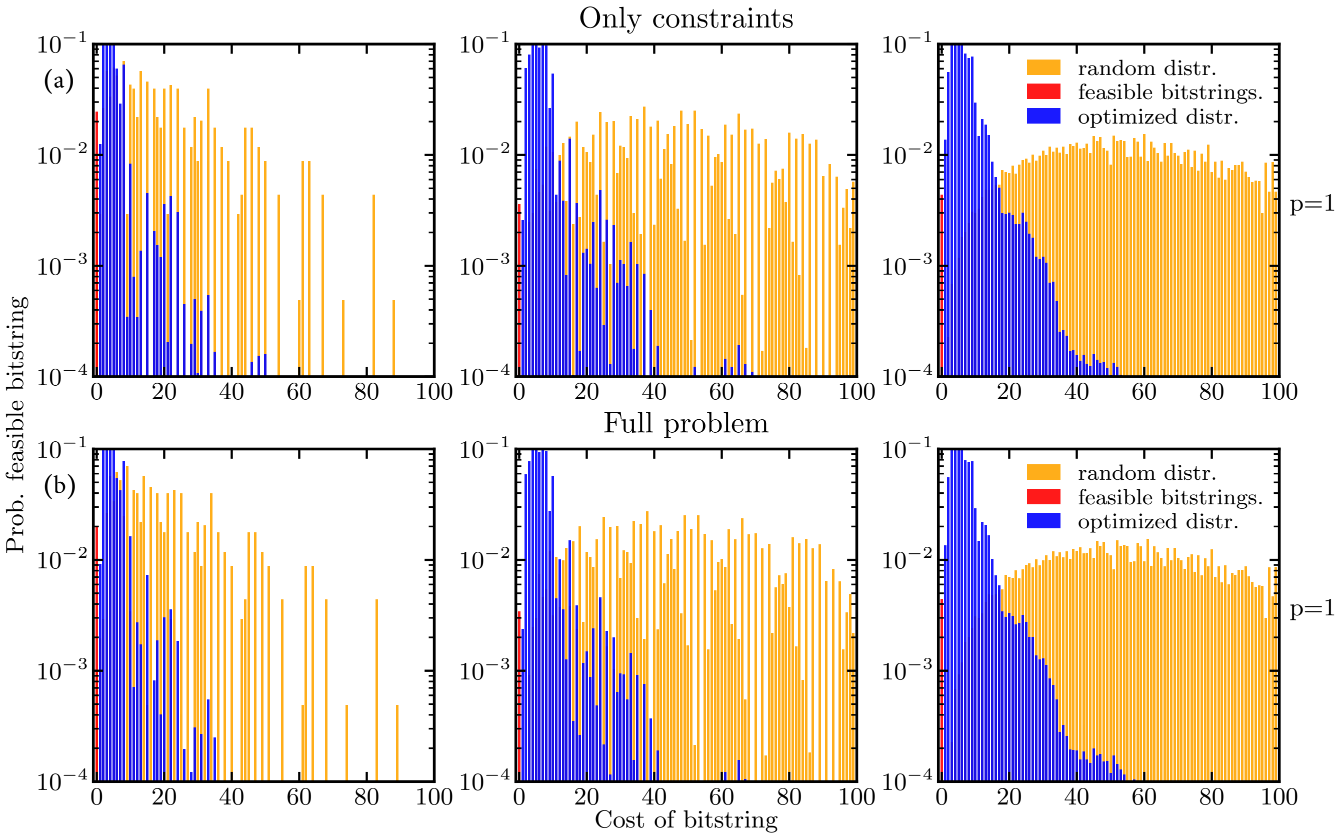

FIG. 7. (a) The probability distribution of the optimized variational state (red, blue) for $p = 1$. The color red shows the probability for finding a valid solution to the routing problem, i.e., a solution satisfying all constraints. The blue color indicates the overall outline of the optimized probability distribution. As a comparison, we show the probability distribution for a variational state in the $|+\rangle^{\otimes n}$ state, meaning all bitstrings are sampled with uniform probability (orange). (b) The probability distribution of the optimized variational state considering the full problem (red, blue) for $p = 1$. The probability distribution is binned such that i is the largest possible integer where $i \leq x$ holds, also denoted as $\lfloor x \rfloor$. That results in all feasible solutions being binned to zero and all the others being binned to integers indicating the number of constraint violations.

consuming. It is therefore important to select a good optimization method.

C. Performance comparison of classical optimizers

Optimizers for finding the variational parameters play an important role in the context of VQAs. Much of the current research aims to find optimizers that perform well on these quantum circuits [61–64]. There is a rich literature on optimizers for variational circuits proposing different optimizers for different problems [35, 65].In the following, we analyze the performance of simulations using four different well-known optimizers: Nelder-Mead [66], Powell [67], differential evolution [68], and basinhopping [69]. The selected optimizers work differently: some use global search mechanisms consisting of multiple random initial guesses while others use only a single random initial guess as a starting point for the optimization. These characteristics are crucial to understand why the performance of the optimizer varies.

The Nelder-Mead method uses a geometrical shape called a simplex to search the function space. With each step of the optimization, the simplex shifts, ideally, towards the region with a minimum. The Nelder-Mead algorithm belongs to the class of gradient-free optimizers. The Powell optimzer works for non-differentiable functions; no derivatives are needed for the optimization. The method minimises the function by a bi-directional search along each search vector. The initial search vectors are typically the normals aligned to each axis. The differential evolution algorithm is stochastic in nature and does not rely on derivatives to find the minimum. This algorithm often requires larger numbers of function evaluations than conventional gradient-based techniques. The basinhopping optimization algorithm is a two-phase method, which couples a global search algorithm with a local minimization at each step. For the simulations here (including in the preceding sections), we used the BFGS algorithm [70] as a local optimizer. This framework has been proven useful for hard nonlinear optimization problems with multiple variables [71].

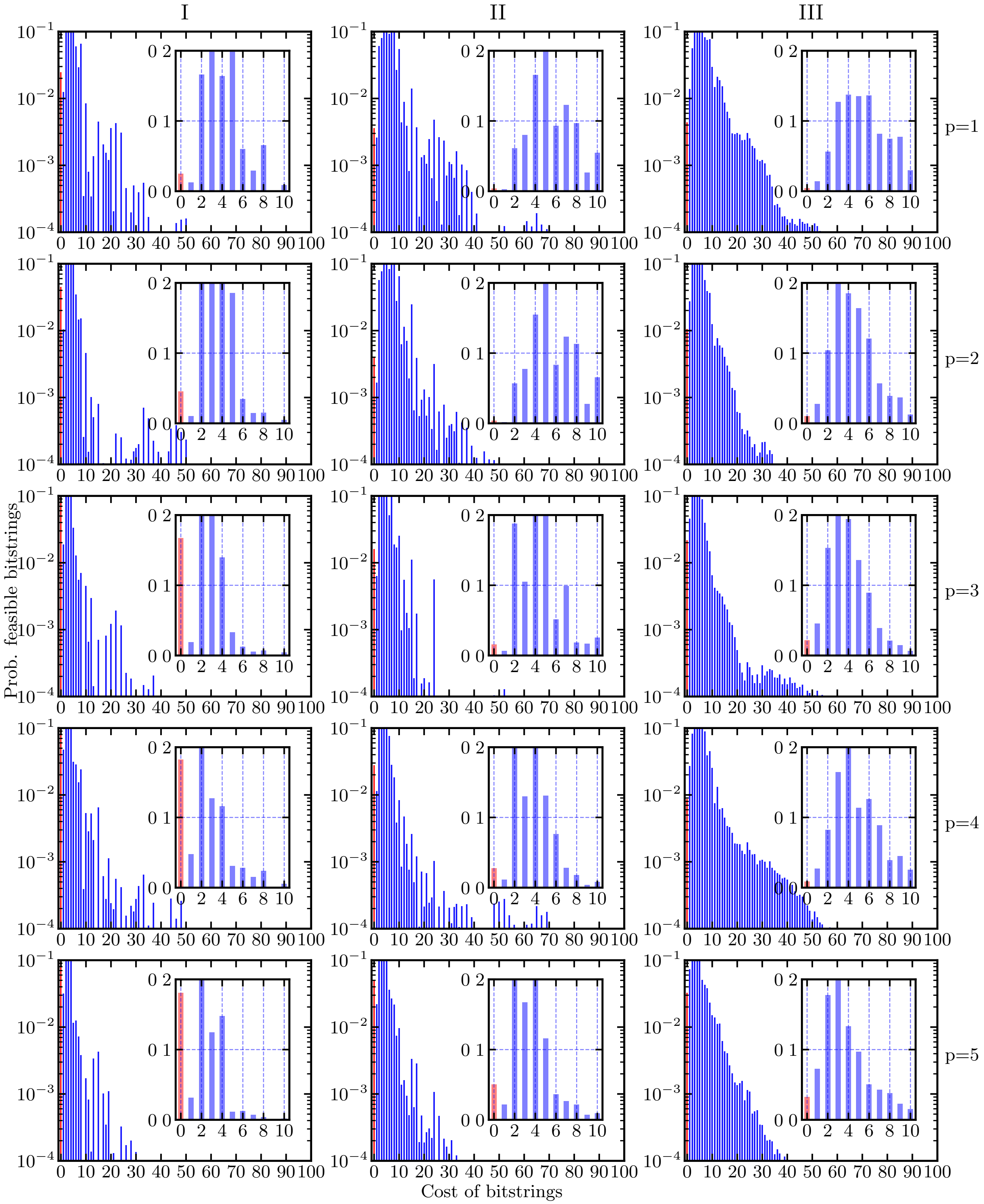

FIG. 8. The probability distribution of the variational state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ for the 11- (left), 19- (middle) and 21-qubit (right) problem instances as a function of the circuit depth p for obtaining feasible tours. The circuit depth increases from top $p = 1$ to bottom $p = 5$ and shifts to lower-energy eigenstates with increasing circuit depth. The probability of sampling bitstrings that encode the optimal solution is marked with red. The simulations were conducted with the classical optimizer basinhopping with the local optimizer BFGS (see Section V C).

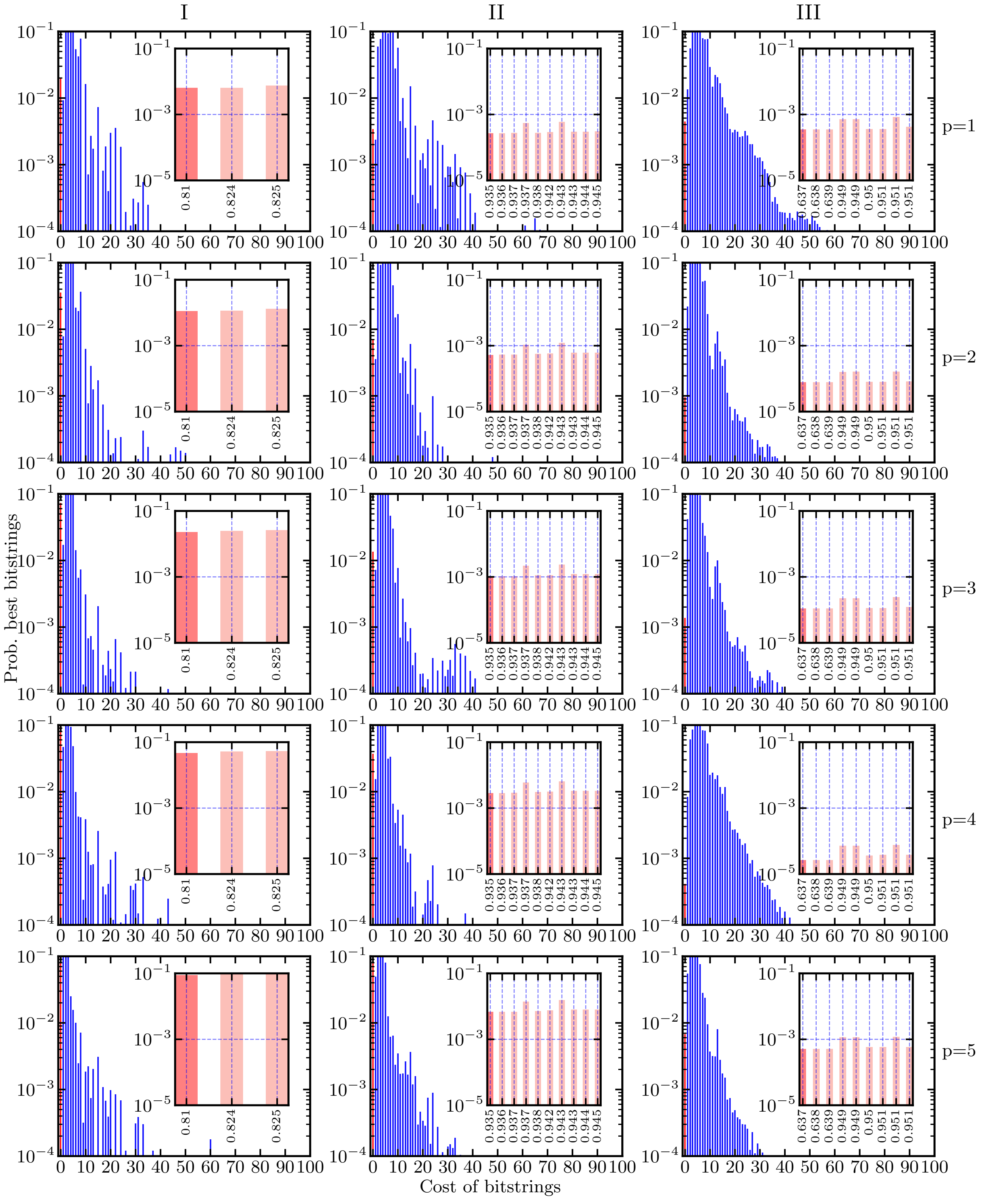

FIG. 9. The probability distribution of the variational state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ for the 11- (left), 19- (middle), and 21-qubit (right) problem instances as a function of the circuit depth p for finding the best tour. The circuit depth increases from top $p = 1$ to bottom $p = 5$ and shifts to lower-energy eigenstates with increasing circuit depth. The probability of sampling the best bitstrings is marked with red. The binning is done in the same way as in Fig. 7(b). The inset shows the probability for sampling any of the two best bitstrings (dark red, leftmost bin) and the probabilities for sampling any of the other feasible bitstrings (light red). The simulations were conducted with the classical optimizer basinhopping with the local optimizer BFGS (see Section V C).

To speed up the optimization routines, we use the optimized parameters from $p-1$ as an intial guess for the optimization of the variational circuits for $p > 1$, as well as the optimized parameters from the 11-qubit problem instance as an initial guess for the 19- and 21-qubit instances. The concentration of variational parameters for the QAOA has been observed by many researchers [58, 72]. It has been shown for a circuit depth of $p = 1, 2$ that the variational parameters for small problem instances can be used to infer parameters for larger problem instances [72]. This can significantly speed up the optimization of the QAOA.

In Fig. 10(a), we show the success probability of finding a viable bitstring as a function of the circuit depth p for the 11-qubit problem instance. Similarly, we show the success probability of sampling one of the two optimal tours as a function of the circuit depth p in Fig. 10(b). One of the optimal tours is shown in Fig. 4. The other optimal solution is following the same path in reverse order.

The results show in Fig. 10 clearly favours the optimizers basinhopping and differential evolution. They perform significantly better than the Nelder-Mead and Powell optimization algorithms. This difference in performance might be due to the global optimization routine that these latter two algorithms use to find the optima. The performance of the Nelder-Mead algorithm strongly depends on the initial simplex and the simplex is usually randomly generated. Depending on the starting point, the performance can vary, but it usually cannot compete with the solution quality of the basinhopping or differential-evolution algorithms.

D. Runtime comparison of the classical optimizers

For applications it is also important to understand the scaling of the runtime of the optimization routine with increasing circuit depth. Here we benchmark the classical optimizers from the preceding subsection on this measure. The result is shown in Fig. 11.

We see that the optimizers basinhopping and differential evolution need roughly one order of magnitude more time for the optimization than the Nelder-Mead and Powell algorithms. This is due to the many circuit queries the former optimizers have to perform. The tradeoff between the runtime of the algorithm and its performance becomes evident, as the slowest optimizers in terms of runtime performed best in terms of success probability (see Fig. 10). Moreover, the linear increase for all optimizers on the semi-log scale in Fig. 11 indicates that the amount of time needed for the optimzation scales exponentially with p. However, further investigation is needed to confirm this statement. It might become a difficult problem to tackle when a circuit depth beyond $p = 20$ is considered.

As mentioned earlier, researchers are already investigating how the optimization could be simplified or circumvented completely [58, 72]. The results here further underlines the importance of such research.

VI. CONCLUSION

We have derived an Ising formulation for the heterogeneous vehicle routing problem (HVRP) under consideration of all relevant constraints, enabling this problem to be solved on a quantum computer. In our formulation, the number of qubits needed to encode a problem instance scales quadratically with the number of customers. Therefore, quantum computers will need to have at least millions of qubits to use our suggested encoding scheme to solve problem instances that are at the limit of what today's classical high-performance optimizers can solve. A quantum advantage could still be had with fewer qubits for smaller problems if they could be solved faster on a quantum computer than on a classical one, but the present work did not give indications of such speed-ups for small problems.

We simulated solving small instances of the HVRP and with the quantum approximate optimization algorithm (QAOA). We considered three distinct problems, requiring 11, 19, and 21 qubits, respectively. We investigated the performance of the algorithm with respect to two design choices: the classical optimizer and the depth p of the quantum circuit.

For the choice of optimizer, we found that the basinhopping and differential-evolution algorithms seem well suited to optimize the variational parameters of the quantum circuit. However, this performance came at the expense of comparably long optimization times. Furthermore, our data indicates that the optimization time needed to find suitable angles for the variational quantum circuit increases exponentially with p, but further investigation is needed to verify this scaling.

We have seen that routing with additional capacity constraints is a difficult problem for the hybrid quantum-classical approach to handle. The problem becomes more evident when isolating the inequality constraint. Then we can see that the energy landscape has multiple scattered local minima. In future work, one might want to consider a different formulation for the capacity constraint [59].

Moreover, Fig. 9 shows that the QAOA fails to distinguish solutions that satisfy all constraints but differ in cost. This failure may be due to the small energy gap between the different feasible solutions and could be circumvented by restating the problem such that feasible solutions are separated by larger energy gaps. One way of achieving this goal is to conduct a rigorous hyperparameter optimization to weight the cost and the constraints accordingly. Similar ideas were explored for the knapsack problem in Ref. [53].

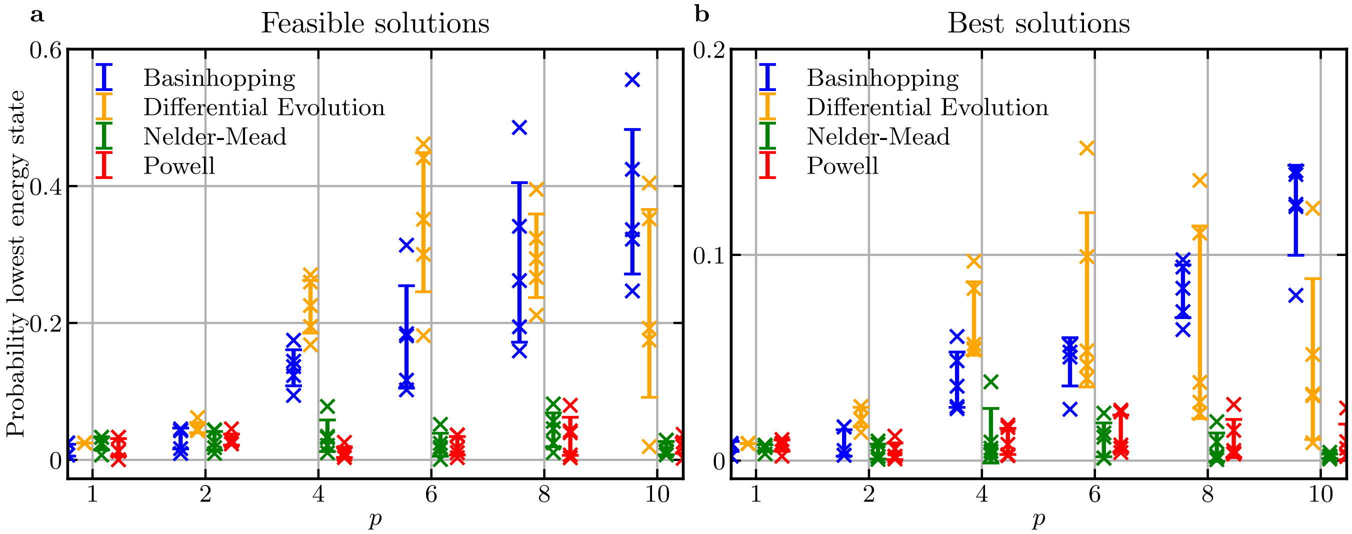

FIG. 10. Comparing different optimizers for the HVRP on the 11-qubit problem instance shown in Fig. 4. A probability of 1 means that a feasible bitstring is always sampled. (a) Success probability for finding a valid solution with increasing circuit depth p. (b) Success probability for finding the best solution with increasing circuit depth p.

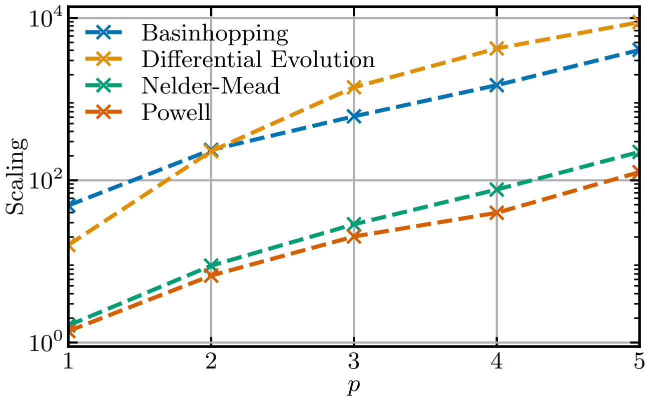

FIG. 11. Runtime comparison for the four classical optimizers tested in this work. The plot shows the runtime as a function of circuit depth p.

VII. OUTLOOK

The classical optimization procedure is a central component in all variational algorithms and key to their success. Therefore, finding new optimizers is one interesting area of research. Several works have investigated machine-learning-based optimizers [64, 73, 74]. A study of such optimizers for the HVRP could boost the performance of the algorithm.

Even though we assumed a noise-free system, the performance is not competitive with modern high-performance heuristics [26, 54] which can solve instances with more than 1,000 customers. Such optimizers are not limited in the number of decision variables, but rather by the running time for the optimization. However, the problems we considered are too small for a meaningful comparison. Furthermore, the assumption of a noise-free system does not hold for NISQ computers and a decrease in performance can be expected if the algorithm is executed on such a device [12, 75]. An interesting topic for future work would be to investigate the HVRP in combination with QAOA under the assumption of a noisy system. Moreover, running small problems on an actual quantum computer could give a better view on the applicability of the QAOA to the HVRP and its competitiveness with classical heuristics.

In Fig. 6, we could see that the knapsack constraint creates an energy landscape with rapidly oscillating local minima. This makes it difficult for many optimizers to find a good approximate solution. It would be interesting to investigate why this particular problem is difficult and if it is possible to relax the knapsack constraint or reformulate it such that the optimization landscape becomes easier to maneuver.

Furthermore, the QAOA can be expanded to the quantum alternating operator ansatz [76]. Investigating different mixer Hamiltonians for the HVRP could lead to a better overall performance. Ideally, the mixer Hamiltonian would provide a framework that keeps the algorithm in the subspace of allowed solutions [77]. Currently, the standard mixer Hamiltonian used in this work makes the algorithms explore every possible bitstring.

Finally, we note that for real-world applications it might be necessary to consider multiple depots. Therefore, another avenue to explore are more Ising formulations that allow for solving different variations of VRPs.

TABLE II. Fuel consumption f and corresponding cost c_{road} per 100km, the price c_{chass} of the chassis as well as the loading capacity [78] for each of the two different truck types considered in the simulations of this paper. Note that the loading capacity does not take into account the size or weight of items that can be carried by the truck. The information in this table is used to determine the variable and fixed costs in Eqs. 18 and 19.

	`rt`	`ts`
f [l/100 km]	28.6 [79]	34.5 [80]
c_{road} [€/100 km][a]	34.32	41.4
c_{chass}[€]	75,000	150,000
m_{max} [items]	3	4

[a] The price per liter is taken to be 1.2 € [81].

ACKNOWLEDGMENTS

DF, MG, and AFK acknowledge financial support from the Knut and Alice Wallenberg foundation through the Wallenberg Centre for Quantum Technology (WACQT). Computations were performed on the Vera cluster at Chalmers Centre for Computational Science and Engineering (C3SE).

Appendix A: Additional information for the simulation of the QAOA

The information used to determine the fixed and variable cost for the QAOA simulations in this work is shown in Table II. The table shows the fuel consumption and cost per 100 km, the price of the chassis, and loading capacity [78] for the two truck types we consider in Fig. 4: a rigid truck (`rt`) and a tractor–semitrailer (`ts`).

[1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (IEEE Comput. Soc. Press, 2002) pp. 124–134.

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum Algorithm for Linear Systems of Equations," Physical Review Letters **103**, 150502 (2009), arXiv:0811.3171.

[3] A. Montanaro, "Quantum algorithms: An overview," npj Quantum Information **2**, 15023 (2016), arXiv:1511.04206.

[4] G. Wendin, "Quantum information processing with superconducting circuits: a review." Reports on progress in physics. Physical Society (Great Britain) **80**, 106001 (2017), arXiv:1610.02208.

[5] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, "Noisy intermediate-scale quantum (NISQ) algorithms," (2021), arXiv:2101.08448.

[6] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," Nature **574**, 505 (2019), arXiv:1911.00577.

[7] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," Applied Physics Reviews **6**, 021314 (2019), arXiv:1904.04178.

[8] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting Qubits: Current State of Play," Annual Review of Condensed Matter Physics **11**, 369–395 (2020), arXiv:1905.13641.

[9] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, "Quantum computational advantage using photons," Science **370**, 1460 (2020), arXiv:2012.01625.

[10] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," (2014), arXiv:1411.4028.

[11] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "Thirty years of heterogeneous vehicle routing," *European Journal of Operational Research*, **249**, 1–21 (2016).

[12] J. Preskill, "Quantum Computing in the NISQ era and beyond," Quantum **2**, 79 (2018), arXiv:1801.00862.

[13] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, "Perspectives of quantum annealing: Methods and implementations," Reports on Progress in Physics **83**, 054401 (2020), arXiv:1903.06559.

[14] B. Golden, S. Raghavan, and E. Wasil, "The vehicle routing problem: Latest advances and new challenges," Operations Research/ Computer Science Interfaces Series **43**, 3–27 (2008).

[15] A. M. Caunhye, X. Nie, and S. Pokharel, "Optimization models in emergency logistics: A literature review," Socio-Economic Planning Sciences **46**, 4–13 (2012).

[16] T. Ghandriz, B. Jacobson, M. Islam, J. Hellgren, and L. Laine, "Transportation-Mission-Based Optimization of Heterogeneous Heavy-Vehicle Fleet Including Electrified Propulsion," Energies **14**, 3221 (2021).

[17] J. K. Lenstra and A. H. Kan, "Complexity of vehicle routing and scheduling problems," Networks **11**, 221–227 (1981).

[18] G. Laporte, S. Ropke, and T. Vidal, "Chapter 4: Heuristics for the Vehicle Routing Problem," in *Vehicle Routing*, September (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014) pp. 87–116.

[19] L. G. Tavares, H. S. Lopes, and C. R. Lima, "Construction and improvement heuristics applied to the capacitated vehicle routing problem," 2009 World Congress on Nature and Biologically Inspired Computing, NABIC

2009 - Proceedings , 690–695 (2009).

[20] C. P. Hwang, B. Alidaee, and J. D. Johnson, "A tour construction heuristic for the travelling salesman problem," Journal of the Operational Research Society **50**, 797–809 (1999).

[21] A. Van Breedam, "Improvement heuristics for the Vehicle Routing Problem based on simulated annealing," European Journal of Operational Research **86**, 480–490 (1995).

[22] C. Brandstätter and M. Reimann, "Performance analysis of a metaheuristic algorithm for the line-haul feeder vehicle routing problem," Journal on Vehicle Routing Algorithms **1**, 121–138 (2018).

[23] G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research **12**, 568–581 (1964).

[24] R. Lougee, "The common optimization interface for operations research: Promoting open-source software in the operations research community," IBM Journal of Research and Development **47**, 57–66 (2003).

[25] C. Groër, B. Golden, and E. Wasil, "A library of local search heuristics for the vehicle routing problem," Math. Prog. Comp **2**, 79–101 (2010).

[26] L. Perron and V. Furnon, "Or-tools," .

[27] R. Baldacci and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," Mathematical Programming **120**, 347–380 (2009).

[28] F. Jazaeri, A. Beckers, A. Tajalli, and J. M. Sallese, "A Review on Quantum Computing: From Qubits to Front-end Electronics and Cryogenic MOSFET Physics," in *Proceedings of the 26th International Conference "Mixed Design of Integrated Circuits and Systems", MIXDES 2019* (IEEE, 2019) pp. 15–25, arXiv:1908.02656v1.

[29] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, "Demonstration of the trapped-ion quantum CCD computer architecture," Nature **592**, 209 (2021), arXiv:2003.01293.

[30] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, "Whole-device entanglement in a 65-qubit superconducting quantum computer," (2021), arXiv:2102.11521.

[31] S. Blinov, B. Wu, and C. Monroe, "Comparison of Cloud-Based Ion Trap and Superconducting Quantum Computer Architectures," (2021), arXiv:2102.00371.

[32] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Höfer, C. Wächter, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz, "A compact ion-trap quantum computing demonstrator," PRX Quantum **2**, 020343 (2021), arXiv:2101.11390.

[33] Y. Wu *et al.*, "Strong quantum computational advantage using a superconducting quantum processor," (2021), arXiv:2106.14734.

[34] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature **549**, 242–246 (2017), arXiv:1704.05018.

[35] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational Quantum Algorithms," *arXiv*, (2020), arXiv:2012.09265.

[36] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, "Option pricing using quantum computers," Quantum **4**, 291 (2020), arXiv:1905.02666.

[37] P. Vikstål, M. Grönkvist, M. Svensson, M. Andersson, G. Johansson, and G. Ferrini, "Applying the quantum approximate optimization algorithm to the tail-assignment problem," Physical Review Applied **14**, 034009 (2020), arXiv:1912.10499.

[38] L. Braine, D. J. Egger, J. Glick, and S. Woerner, "Quantum Algorithms for Mixed Binary Optimization Applied to Transaction Settlement," *IEEE Transactions on Quantum Engineering*, **2**, 3101208 (2021), arXiv:1910.05788.

[39] H. Choi, K. S. Sohn, M. Pyo, K. C. Chung, and H. Park, "Predicting the Electrochemical Properties of Lithium-Ion Battery Electrode Materials with the Quantum Neural Network Algorithm," Journal of Physical Chemistry C **123**, 4682–4690 (2019).

[40] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, "Quantum Computing for Finance: State-of-the-Art and Future Prospects," *IEEE Transactions on Quantum Engineering*, **1**, 3101724 (2021), arXiv:2006.14510.

[41] A. Syrichas and A. Crispin, "Large-scale vehicle routing problems: Quantum Annealing, tunings and results," Computers and Operations Research **87**, 52–62 (2017).

[42] S. Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Mauerer, and C. Linnhoff-Popien, "A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer," Frontiers in ICT **6**, 13 (2019), arXiv:1811.07403.

[43] R. Harikrishnakumar, S. Nannapaneni, N. H. Nguyen, J. E. Steck, and E. C. Behrman, "A Quantum Annealing Approach for Dynamic Multi-Depot Capacitated Vehicle Routing Problem," (2020), arXiv:2005.12478.

[44] Utkarsh, B. K. Behera, and P. K. Panigrahi, "Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm," (2020), arXiv:2002.01351.

[45] M. P. Harrigan *et al.*, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," Nature Physics **17**, 332–336 (2021), arXiv:2004.04197.

[46] A. Bengtsson, P. Vikstål, C. Warren, M. Svensson, X. Gu, A. F. Kockum, P. Krantz, C. Križan, D. Shiri, I.-M. Svensson, G. Tancredi, G. Johansson, P. Delsing, G. Ferrini, and J. Bylander, "Improved Success Probability with Greater Circuit Depth for the Quantum Approximate Optimization Algorithm," Physical Review Applied **10**, 34010 (2020).

[47] D. M. Abrams, N. Didier, B. R. Johnson, M. P. Silva, and C. A. Ryan, "Implementation of XY entangling gates with a single calibrated pulse," Nature Electronics **3**, 744–750 (2020).

[48] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, "Benchmarking the quantum approximate optimization algorithm," Quantum Information Processing **19**, 197 (2020), arXiv:1907.02359.

[49] J. S. Otterbach *et al.*, "Unsupervised Machine Learning on a Hybrid Quantum Computer," *arXiv*, (2017), arXiv:1712.05771.

[50] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," Zeitschrift für Physik **31**, 253 (1925).

[51] S. G. Brush, "History of the Lenz-Ising Model," Reviews of Modern Physics **39**, 883 (1967).

[52] A. Lucas, "Ising formulations of many NP problems," Frontiers in Physics **2**, 5 (2014), arXiv:1302.5843.

[53] C. Roch, A. Impertro, T. Phan, T. Gabor, S. Feld, and C. Linnhoff-Popien, "Cross entropy hyperparameter optimization for constrained problem hamiltonians applied to QaoA," Proceedings - 2020 International Conference on Rebooting Computing, ICRC 2020 , 50–57 (2020), arXiv:2003.05292.

[54] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian, "New benchmark instances for the Capacitated Vehicle Routing Problem," European Journal of Operational Research **257**, 845–858 (2017).

[55] S. E. Barman, P. Lindroth, and A.-B. Strömberg, "Modeling and Solving Vehicle Routing Problems with Many Available Vehicle Types," in *Optimization, Control, and Applications in the Information Age*, edited by A. Migdalas and A. Karakitsiou (Springer International Publishing, Cham, 2015) pp. 113–138.

[56] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2000).

[57] M. Svensson, M. Andersson, M. Grönkvist, P. Vikstål, D. Dubhashi, G. Ferrini, and G. Johansson, "A Heuristic Method to solve large-scale Integer Linear Programs by combining Branch-and-Price with a Quantum Algorithm," (2021), arXiv:2103.15433.

[58] 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), arXiv:1908.08862.

[59] P. D. de la Grand'rive and J.-F. Hullo, "Knapsack Problem variants of QAOA for battery revenue optimisation," (2019), arXiv:1908.02210.

[60] E. Farhi, D. Gamarnik, and S. Gutmann, "The Quantum Approximate Optimization Algorithm Needs to See the Whole Graph: Worst Case Examples," (2020), arXiv:2005.08747.

[61] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, "Improving Variational Quantum Optimization using CVaR," Quantum **4**, 256 (2020), arXiv:1907.04769.

[62] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, "An adaptive optimizer for measurement-frugal variational algorithms," Quantum **4**, 263 (2019), arXiv:1909.09083.

[63] J. C. Spall, *IEEE Transactions on Automatic Control*, Tech. Rep. 3 (1992).

[64] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, "Learning to learn with quantum neural networks via classical neural networks," (2019), arXiv:1907.05415.

[65] W. Lavrijsen, A. Tudor, J. Muller, C. Iancu, and W. De Jong, "Classical Optimizers for Noisy Intermediate-Scale Quantum Devices," in *Proceedings - IEEE International Conference on Quantum Computing and Engineering, QCE 2020* (2020) pp. 267–277, arXiv:2004.03004.

[66] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," The Computer Journal **7**, 308–313 (1965).

[67] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," The Computer Journal **7**, 155–162 (1964).

[68] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," Journal of Global Optimization (1996).

[69] D. Wales and J. Doye, "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms," The Journal of Physical Chemistry A **101**, 5111–5116 (1998), arXiv:9803344 [cond-mat].

[70] J. Frédéric Bonnans, J. Charles Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects* (2006) pp. 1–494.

[71] B. Olson, I. Hashmi, K. Molloy, and A. Shehu, "Basin Hopping as a General and Versatile Optimization Framework for the Characterization of Biological Macromolecules," Advances in Artificial Intelligence **2012**, 674832 (2012).

[72] V. Akshay, D. Rabinovich, E. Campos, and J. Biamonte, "Parameter concentrations in quantum approximate optimization," Physical Review A **104**, L010401 (2021), arXiv:2103.11976.

[73] J. Yao, M. Bukov, and L. Lin, "Policy Gradient based Quantum Approximate Optimization Algorithm," Proceedings of Machine Learning Research **107**, 1–30 (2020), arXiv:2002.01068.

[74] A. Garcia-Saez and J. Riu, "Quantum Observables for continuous control of the Quantum Approximate Optimization Algorithm via Reinforcement Learning," (2019), arXiv:1911.09682.

[75] M. Streif, M. Leib, F. Wudarski, E. Rieffel, and Z. Wang, "Quantum algorithms with local particle-number conservation: Noise effects and error correction," Physical Review A **103**, 042412 (2021), arXiv:2011.06873.

[76] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz," Algorithms **12**, 34 (2019), arXiv:1709.03489.

[77] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, "XY mixers: Analytical and numerical results for the quantum alternating operator ansatz," Physical Review A **101**, 012320 (2020), arXiv:1904.09314.

[78] T. Ghandriz, *Transportation Mission Based Optimization of Heavy Vehicle Fleets including Propulsion Tailoring*, Licentiate thesis, Chalmers University of Technology (2018).

[79] "Average fuel consumption in Australia," (2020).

[80] U. Clausen, H. Friedrich, C. Thaller, and C. Geiger, *Commercial Transport: Proceedings of the 2nd Interdisciplinary Conference on Production Logistics and Traffic 2015*, Lecture Notes in Logistics (Springer International Publishing, 2015).

[81] "Germany Diesel prices, 09-Mar-2020," (2020).