



## **Applying quantum approximate optimization to the heterogeneous vehicle routing problem**

Downloaded from: <https://research.chalmers.se>, 2024-11-16 02:13 UTC

Citation for the original published paper (version of record):

Fitzek, D., Ghandriz, T., Laine, L. et al (2024). Applying quantum approximate optimization to the heterogeneous vehicle routing problem. *Scientific Reports*, 14(1).

<http://dx.doi.org/10.1038/s41598-024-76967-w>

N.B. When citing this work, cite the original published paper.



## OPEN Applying quantum approximate optimization to the heterogeneous vehicle routing problem

David Fitzek<sup>1,2</sup>✉, Toheed Ghandriz<sup>2,3</sup>, Leo Laine<sup>2,3</sup>, Mats Granath<sup>4</sup>✉ & Anton Frisk Kockum<sup>1</sup>✉

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 show 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.

Devices utilizing quantum-mechanical effects provide a new computational paradigm that enables novel algorithms and heuristics<sup>1–4</sup>. The ongoing development of such devices<sup>5–8</sup> 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)<sup>9</sup> can be employed to find approximate solutions for the HVRP<sup>10</sup>. Our approach can be utilized on both noisy intermediate-scale quantum (NISQ)<sup>11</sup> computers and quantum annealers<sup>12</sup>. It also paves the way for implementing challenging instances of the HVRP on larger quantum computers in the future.

The HVRP belongs to the extensively studied class of optimization problems known as the vehicle routing problem (VRP)<sup>13,14</sup> in the field of logistics. The VRP is 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<sup>15,16</sup>. Variants of the VRP include the capacitated vehicle routing problem (CVRP), in which the vehicles have a limited carrying capacity<sup>13</sup>, and the HVRP studied here, in which the vehicles that are to be deployed need to be determined from a fixed set of vehicles and capacity constraints are given<sup>10,17</sup>. All these VRPs are challenging since they belong to the complexity class NP-hard<sup>18</sup>.

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<sup>19,20</sup>, e.g., construction heuristics<sup>21,22</sup>, improvement heuristics<sup>23</sup>, and metaheuristic top-level strategies<sup>24</sup>. In construction heuristics, e.g., the Clarke and Wright savings algorithm<sup>25</sup>, 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<sup>26–28</sup>. Moreover, exact methods for solving the VRP and its variants have also been investigated<sup>29</sup>.

In this article, we instead investigate a heuristic method for solving the HVRP on a quantum computer. Such devices, including both programmable quantum processors<sup>2</sup> and quantum annealers<sup>12</sup>, are gradually becoming available due to recent advances in controlling quantum systems. The current programmable gate-based quantum computers known as NISQ devices<sup>11</sup> are largely limited by their intermediate number (several tens or hundreds<sup>5,7,8,30</sup>) of controllable qubits, limited connectivity, imperfect qubit control, short coherence times, and minimal error correction. On the other hand, quantum annealers have been able to leverage thousands of qubits,

<sup>1</sup>Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden. <sup>2</sup>Volvo Group Trucks Technology, 405 08 Gothenburg, Sweden. <sup>3</sup>Department of Mechanics and Maritime Sciences, Chalmers University of Technology, 412 96 Gothenburg, Sweden. <sup>4</sup>Department of Physics, University of Gothenburg, 412 96 Gothenburg, Sweden. ✉email: davidfi@chalmers.se; mats.granath@physics.gu.se; anton.frisk.kockum@chalmers.se

and have been successfully used to solve larger combinatorial problem instances<sup>31</sup>. However, only a subset of quantum algorithms can run on NISQ devices<sup>4,32</sup>; other algorithms require more advanced hardware.

The heuristic method we apply to the HVRP here is an example of a variational quantum algorithm (VQA)<sup>3</sup>, which is a promising class of hybrid quantum-classical algorithms that are compatible with NISQ devices. From a description of the problem, the first step is to define a cost (or loss) function  $C$ , which encodes the quality of a solution to the problem. Next, one proposes an ansatz, i.e., a quantum operation depending on a set of continuous or discrete parameters  $\theta$ . This ansatz is then optimized in a hybrid quantum-classical loop to approach  $\theta^* = \underset{\theta}{\operatorname{argmin}} C(\theta)$ . Such algorithms have emerged as a leading contender for obtaining quantum advantage<sup>3</sup> within the constraints of NISQ devices. Variational quantum algorithms have been proposed for numerous applications envisioned for quantum computers, e.g., in chemistry, logistics, and finance<sup>33–35</sup>.

The type of VQA we employ here is the QAOA<sup>9</sup>, which is a heuristic that can approximate the solution to many combinatorial problems, including VRPs. Current research in this area<sup>36</sup> ranges from applications on large-scale VRP instances with a quantum annealer<sup>37</sup> to more specific variants of the VRP, such as the CVRP<sup>31</sup>, the multi-depot capacitated VRP<sup>38</sup>, and the VRP with time windows<sup>39</sup>. These approximation algorithms have been tested on quantum annealers<sup>31</sup> and NISQ devices<sup>40</sup>. There have also been several experimental realizations of the QAOA applied to other optimization problems<sup>41–44</sup>, including the travelling salesperson problem (TSP)<sup>45</sup>, which is of fundamental importance for VRPs.

However, a problem description suited for the QAOA, an Ising Hamiltonian<sup>46–48</sup> (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 QAOA applied to the HVRP, we simulate problem instances with up to 21 qubits, which corresponds to three customers and two trucks. We check how the solution quality 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. We first introduce the HVRP and its mathematical formulation. Then we develop the Ising formulation of the HVRP. Thereafter, we review the QAOA and describe how it can be used to find approximate solutions to the HVRP. We then 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.

### The heterogeneous vehicle routing problem

The HVRP can be formulated as follows, as originally proposed in Ref.<sup>10</sup> inspired by earlier works<sup>49,50</sup>. 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,  $\mathcal{N} = \{0, \dots, n\}$  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 arcs. Each customer  $i$  has a positive demand  $q_i$ .

The set of available vehicle types is  $\mathcal{V} = \{1, \dots, k\}$ , with  $m_v$  vehicles of type  $v \in \mathcal{V}$ . When using these vehicles to deliver goods to meet the customer demand, several costs and constraints need to be taken into account. There is the fixed vehicle cost  $t^v$ , i.e., the cost that is independent of the distance travelled by a vehicle of type  $v$ . Then, there is the vehicle capacity  $Q^v$ . Different vehicle types can have the same capacities, but differ in, e.g., the type of powertrain used<sup>17</sup>. Finally, there is the cost  $c_{ij}^v$  of travelling on edge  $(i, j)$  with a vehicle of type  $v$ . To describe the constraints, it is 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  $(i, j)$ . We denote by  $f_{ij}^v$  the amount of goods that are leaving node  $i$  to go to node  $j$  using a truck of type  $v$ , while the amount of goods entering the node is denoted  $f_{ji}^v$ .

Using this notation, the HVRP is to minimize the cost

$$C_{\text{tot}} = \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}_0} t^v x_{0j}^v + \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{E}} c_{ij}^v x_{ij}^v, \tag{1}$$

subject to the constraints

$$\sum_{j \in \mathcal{N}_0} x_{0j}^v \leq m_v \quad v \in \mathcal{V}, \tag{2}$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}} x_{ij}^v = 1 \quad i \in \mathcal{N}_0, \tag{3}$$

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

$$\sum_{j \in \mathcal{N}_0} x_{j0}^v = \sum_{j \in \mathcal{N}_0} x_{0j}^v \quad v \in \mathcal{V}, \tag{5}$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}} f_{ji}^v - \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}} f_{ij}^v = q_i \quad i \in \mathcal{N}_0, \tag{6}$$

$$q_j x_{ij}^v \leq f_{ij}^v \leq (Q_v - q_i) x_{ij}^v \quad (i, j) \in \mathcal{E}, v \in \mathcal{V}, \tag{7}$$

$$x_{ij}^v \in \{0, 1\} \quad (i, j) \in \mathcal{E}, v \in \mathcal{V}, \tag{8}$$

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

The objective function in Eq. (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. (2) ensures that the maximum number of available vehicles for a specific vehicle type is not exceeded. The constraints in Eqs. (3) and (4) make sure that each customer is visited exactly once, and the constraint in Eq. (5) sees to that all vehicles leaving the depot return to it after delivering their goods. The constraints in Eqs. (6) and (7) ensure a correct commodity flow that meets all customer demands: Eq. (6) specifies that the amount of goods arriving to node  $i$  minus the amount of goods leaving node  $i$  should equal the customer demand  $q_i$  of customer  $i$ , and Eq. (7) describes that the amount of goods  $f_{ij}^v$  being carried from node  $i$  to node  $j$  needs to be at least enough to match customer  $j$ 's demand  $q_j$  and should not exceed the capacity remaining in the vehicle after it delivered its goods to fulfil the demands  $q_i$  of customer  $i$ . Finally, the constraints in Eqs. (8) and (9) enforce the binary form and non-negativity restrictions on the variables.

### 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 Hamiltonian<sup>48</sup>. 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.

#### Routing problem

For the routing problem, we start from the TSP formulation in Ref.<sup>48</sup> with the Hamiltonian  $H^{\text{TSP}} = H_A^{\text{TSP}} + H_B^{\text{TSP}}$ , where

$$H_A^{\text{TSP}} = A \sum_{i=1}^N \left( 1 - \sum_{\alpha=1}^N y_{i\alpha} \right)^2 + A \sum_{\alpha=1}^N \left( 1 - \sum_{i=1}^N y_{i\alpha} \right)^2 + A \sum_{(i,j) \notin \mathcal{E}} \sum_{\alpha=1}^N y_{i\alpha} y_{j\alpha+1}, \tag{10}$$

$$H_B^{\text{TSP}} = B \sum_{(i,j) \in \mathcal{E}} W_{ij} \sum_{\alpha=1}^N y_{i\alpha} y_{j\alpha+1}, \tag{11}$$

encoding the total cost. Here  $N = |\mathcal{N}|$  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  $\alpha$  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. Minimizing the first term in Eq. (10) leads to each node being visited exactly once and minimizing the second term in Eq. (10) ensures that each cycle has exactly one entry for each order. The last term in Eq. (10), which ensures that non-existent arcs are not used, can be neglected for the problem we investigate because we assume a fully connected graph. The Hamiltonian  $H_B^{\text{TSP}}$  in Eq. (11) adds up the total distance travelled.

Above, we used  $y_{i\alpha}^v$  as a routing variable that tracks the order  $\alpha$  in which node  $i$  is visited within a cycle. To blend this framework with the mathematical structure of the HVRP, formulated in Eqs. (1)–(9), we introduce a transformation from the decision variable  $y$  to  $x$  as follows:

$$x_{ij}^v = \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v y_{j\alpha+1}^v, \tag{12}$$

$$x_{0i}^v = y_{i1}^v + \sum_{\alpha=2}^{N_0} \left( 1 - \sum_{\substack{j=1 \\ j \neq i}}^{N_0} y_{j\alpha-1}^v \right) y_{i\alpha}^v, \quad x_{i0}^v = y_{iN_0}^v + \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v \left( 1 - \sum_{\substack{j=1 \\ j \neq i}}^{N_0} y_{j\alpha+1}^v \right). \tag{13}$$

The summation in Eq. (12) is not equal 0 if and only if  $i$  and  $j$  are subsequent stops on the same route. Equation (13) ensures that the first and last stops are automatically connected to the depot (assuming a single depot). By using the decision variables  $y_{i\alpha}^v$  that indicate the position in a prospective cycle instead of  $x_{ij}^v$  that is 1 if and only if a vehicle traverses from customer  $i$  to  $j$ <sup>38</sup>, we take care of the subtour-elimination constraint encoded in Eq. (6), which otherwise allows for closed loops between customers, without returning to the depot.

The transformation from  $y_{i\alpha}^v$  to  $x_{ij}^v$  as defined in Eqs. (12)–(13) is unique. However, the inverse transformation, i.e., from the HVRP model variables  $x_{ij}^v$  to the quantum model variables  $y_{i\alpha}^v$ , is not unique. This becomes evident if we consider that any permutation of the sequence  $y_{i\alpha}^v$  for a fixed  $i$  will yield the same  $x_{ij}^v$  values. Hence, a single solution in the HVRP model could correspond to multiple valid solutions in the Ising model. Despite this non-

uniqueness, every feasible solution in the HVRP model has a corresponding set of solutions in the Ising model, ensuring the robustness and validity of our approach.

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)<sup>48,51</sup>. 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 Eqs. (1)–(9)], 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 Eqs. (1)–(9)]. The Ising Hamiltonian  $H = H_A + H_B + H_C$  we derive is then merging Eqs. (12)–(13) with Eqs. (1)–(9):

$$H_A = A \sum_{v=1}^V \sum_{i=1}^{N_0} \sum_{j=1}^{N_0} c_{ij}^v \sum_{\alpha=1}^{N_0-1} y_{i\alpha}^v y_{j\alpha+1}^v + A \sum_{v=1}^V \sum_{i=1}^{N_0} (t_v + c_{0i}^v) \left[ y_{i1}^v + \sum_{\alpha=2}^{N_0} \left( 1 - \sum_{\substack{j=1 \\ j \neq i}}^{N_0} y_{j\alpha-1}^v \right) y_{i\alpha}^v \right] \tag{14}$$

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

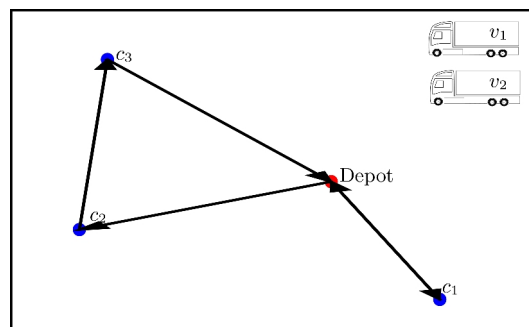
$$H_B = B \sum_{i=1}^{N_0} \left( 1 - \sum_{\alpha=1}^{N_0} \sum_{v=1}^V y_{i\alpha}^v \right)^2, \tag{15}$$

$$H_C = C \sum_{\alpha=1}^{N_0} \left( 1 - \sum_{i=1}^{N_0} \sum_{v=1}^V y_{i\alpha}^v \right)^2. \tag{16}$$

The Hamiltonian is composed of different parts.  $H_A$  in Eq. (14) captures the total cost of the original mathematical formulation [Eq. (1)], i.e., the minimization of the variable and fixed 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. Note that the fixed cost is included in the second term as well, since all the vehicles leaving the depot generate the fixed cost.

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. (2) from the original formulation, which ensures that the number of vehicles of a specific type does not exceed the number of available vehicles.

The constraint given by  $H_B$  in Eq. (15) ensures that each city is visited exactly once [i.e., satisfies the constraints in Eqs. (3) and (4) in the original HVRP formulation]. Furthermore,  $H_C$  in Eq. (16) 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  $0 < \max(|H_A|) < B, C$ . If these inequalities are not fulfilled, we risk obtaining unphysical solutions that have lower energies than valid ones. One notable technicality about the formulation is that certain solutions that may be considered valid are excluded by the constraint in Eq. (15). However, the excluded solutions are physically equivalent to some allowed solution, as illustrated by the example in Fig. 1 with two trucks and three cities:



**Fig. 1.** Visualization of a problem instance with a suggested solution (other solutions are possible; we assume a fully connected graph). The first truck  $v_1$  visits cities  $c_2$  and  $c_3$  before returning to the depot. The second truck  $v_2$  only visits city  $c_1$ .

$$y = \left[ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right] \quad \text{and} \quad y = \left[ \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right], \quad (17)$$

where the matrix rows correspond to the index  $i$  (the node visited), the matrix columns correspond to the index  $\alpha$  (the order in which the nodes are visited), and each matrix corresponds to a vehicle  $v$ .

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 ( $v_1 : c_2 \rightarrow c_3$ ), 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. (15), 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. (15), we can replace that constraint by a reformulated one,

$$H'_B = B \sum_{v=1}^V \sum_{\alpha=1}^{N_0} \left( u_{\alpha}^v - \sum_{i=1}^{N_0} y_{i\alpha}^v \right)^2, \quad (18)$$

where we have introduced  $N_0^2$  additional auxiliary qubits  $u_{\alpha}^v$ . 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. (18) a viable route for implementations, but use Eq. (15) for the simulations later in this article.

### 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.<sup>48</sup> 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  $\mathcal{W} = \sum_{i=1}^N w_i x_i$  with a total value of  $\mathcal{C} = \sum_{i=1}^N c_i x_i$ . The knapsack problem is to maximize  $\mathcal{C}$  under the constraint  $\mathcal{W} \leq W$ . This problem belongs to the complexity class NP-hard<sup>48</sup>.

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<sup>48</sup>. The Hamiltonian  $H^K = H_A^K + H_B^K$  whose energy we seek to minimize is then

$$H_A^K = A \left( 1 - \sum_{n=1}^W z_n \right)^2 + A \left( \sum_{n=1}^W n z_n - \sum_{i=1}^N w_i x_i \right)^2, \quad (19)$$

$$H_B^K = -B \sum_{i=1}^N c_i x_i. \quad (20)$$

To ensure that the hard constraint is respected, we require  $0 < \max(|H_B^K|) < A$ .

#### 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  $M \equiv \lceil \log_2(W) \rceil$ . We then require  $M + 1$  binary variables instead of  $W$  binary variables:

$$\sum_{n=1}^W n z_n \rightarrow \sum_{m=0}^{M-1} 2^m z_m + (W + 1 - 2^M) z_M. \quad (21)$$

It is important to note that in this encoding method, multiple bitstrings  $z_m$  can represent the same number. This situation arises due to the transition from one-hot encoding to a binary representation. In one-hot encoding, each integer from 0 to  $W$  would have a unique bitstring representation. However, in our binary representation, where we encode numbers using fewer bits, different bitstrings can be interpreted as the same integer. This degeneracy is possible if  $W \neq 2^{M+1} - 1$ .

We can make use of the inequality constraint given in the knapsack formulation [Eq. (19)] to encode the capacity constraints for the HVRP. Therefore, we can neglect  $H_B^K$  [Eq. (20)] and only consider  $H_A^K$  [Eq. (19)]. Let  $Q^v$  be the maximum capacity of vehicle  $v$ . The Hamiltonian then becomes

$$H^K = A \sum_v \left( 1 - \sum_{k=0}^{Q^v} z_k^v \right)^2 + A \sum_v \left( \sum_{k=0}^{Q^v} k z_k^v - \sum_{\alpha,i} q_i y_{i\alpha}^v \right)^2, \quad (22)$$

or equivalently, using the log formulation,

$$H^K = A \sum_v \left( \sum_{k=0}^{M^v-1} 2^k z_k^v + (Q^v + 1 - 2^{M^v}) z_{M^v}^v - \sum_{\alpha,i} q_i y_{i\alpha}^v \right)^2, \quad (23)$$

where  $M^v \equiv \lceil \log_2(Q^v) \rceil$ . Note that by using the log trick, the decision variable  $z_k^v$  switches from a one-hot encoding to a binary representation. The lowest energy possible for the Hamiltonian in Eq. (23) is zero; this minimum is reached when the  $z_k^v$  are chosen such that each vehicle  $v$  carries an amount of goods, given by the first inner sum in Eq. (23), not exceeding its maximum capacity, and that amount exactly corresponds to the demand of the customers it visits along its route, given by the second inner sum in Eq. (23).

### 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 four 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:

$$H_{\mathcal{C}} = H_A + H_B + H_C + H_D, \quad (24)$$

$$H_D = D \sum_{v=1}^V \left( \sum_{k=0}^{M^v-1} 2^k z_k^v + (Q^v + 1 - 2^{M^v}) z_{M^v}^v - \sum_{\alpha=1}^{N_0} \sum_{i=1}^{N_0} q_i y_{i\alpha}^v \right)^2 \quad (25)$$

For the terms  $H_A$  to  $H_C$ , see Eqs. (14)–(16). Note that  $H_D$ , together with  $H_B$ , ensures that we satisfy the commodity-flow constraints in Eqs. (6) and (7) in the original HVRP formulation.

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.<sup>31</sup> 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.<sup>38</sup>. 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.

### Resources

The required resources (qubits) for solving the HVRP with our approach can be separated into two parts. The first part comes from representing all the possible routes between the customers in the routing part of the problem and scales with  $N_0^2 \cdot V$ . Additional qubits are required for the constraining term  $H_D$ . The total number of qubits required is

$$\#q = N_0^2 \cdot V + \underbrace{\sum_{v=1}^V \lceil \log_2 Q^v \rceil + 1}_{H_D}. \quad (26)$$

Using the alternative formulation for  $H_B$  [see Eq. (18)] adds more auxiliary qubits ( $N_0 \cdot V$ ), yielding

$$\#q = (N_0 + 1)N_0V + \sum_{v=1}^V \lceil \log_2 Q^v \rceil + 1. \quad (27)$$

Furthermore, we can estimate the number of single- and two-qubit gates required to apply the QAOA (see next section) to the problem in this formulation. For the routing part of the problem, each layer of the QAOA needs  $O(N_0^2 \cdot V)$  single-qubit gates (i.e., proportional to the number of qubits) and  $O(N_0^3 \cdot V)$  two-qubit gates (i.e., proportional to  $N_0$  times the number of qubits). For the capacity part of the problem, the number of single-qubit gates is the same, while the number of two-qubit gates is  $O(N_0^2 \cdot Q)$ . What this translates to in circuit depth will depend on the connectivity of the quantum hardware.

As a comparison, we note that modern high-performance optimizers (classical computers) for the HVRP can find near-optimal solutions for problem instances with more than 1,000 customers<sup>52,53</sup>. Typically, these optimizers exhibit polynomial or pseudo-polynomial time complexity. 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.

## 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. 2. The quantum subroutine, operating on  $n$  qubits, consists of a consecutive application of two non-commuting operators defined as<sup>9</sup>

$$U(\gamma) \equiv e^{-i\gamma H_C} \quad \gamma \in [0, 2\pi], \quad (28)$$

$$U(\beta) \equiv e^{-i\beta H_M} = \prod_{j=1}^n e^{-i\beta\sigma_j^x} \quad \beta \in [0, \pi], \quad (29)$$

where  $\sigma^x$  denotes the Pauli X matrix. 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 [Eq. (29) is a transverse-field mixer]. The bounds for  $\gamma$  and  $\beta$  are valid if  $H_C$  has integer eigenvalues<sup>9</sup>. The formulation of  $H_C$  for the HVRP is given by Eq. (24).

The initial state for the algorithm is a superposition of all possible computational basis states. This superposition can be obtained, as shown in Fig. 2, by first preparing the system in the initial state  $|0\rangle^{\otimes n} = |00\dots 0\rangle$  for all qubits and then applying the Hadamard gate on each qubit:

$$(\tilde{H}|0\rangle)^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} \equiv |+\rangle^{\otimes n}, \quad (30)$$

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

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

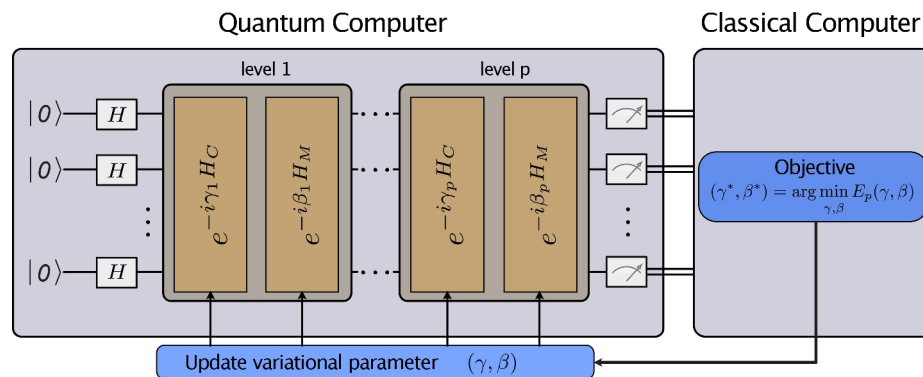
$$|\gamma, \beta\rangle = U(\beta_p)U(\gamma_p) \dots U(\beta_1)U(\gamma_1) |+\rangle^{\otimes n}. \quad (31)$$

The quantum circuit parametrized by  $\gamma$  and  $\beta$  is then optimized in a closed loop using a classical optimizer. The objective is to minimize the expectation value of the cost Hamiltonian  $H_C$ <sup>9</sup>, i.e.,

$$(\gamma^*, \beta^*) = \operatorname{argmin}_{\gamma, \beta} E(\gamma, \beta), \quad E(\gamma, \beta) = \langle \gamma, \beta | H_C | \gamma, \beta \rangle. \quad (32)$$

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

The integer  $p$  plays a crucial role in the overall solution quality. The performance of the algorithm is expected to become better with more variational parameters. Furthermore, there exists a close connection between QAOA and adiabatic quantum computing: the QAOA can reproduce a Trotterized version of the time evolution in adiabatic quantum computing<sup>9,54</sup>.

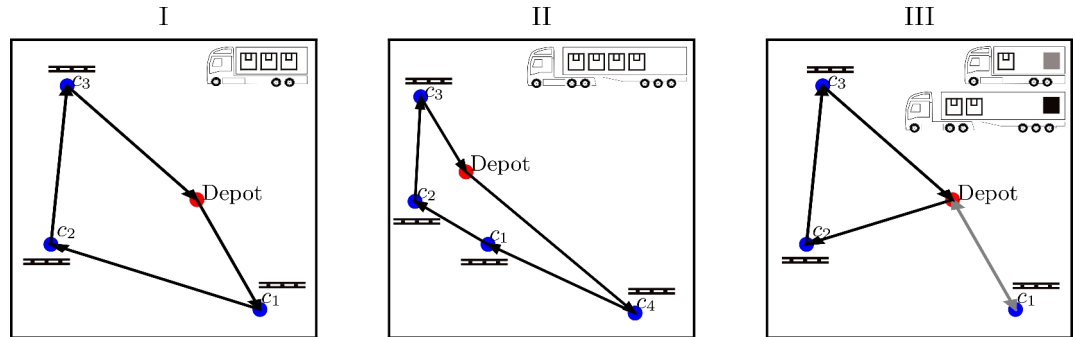


**Fig. 2.** A schematic description of the QAOA, visualizing the interplay between the quantum device and the classical computer. The quantum computer implements a variational state formed by applying  $p$  parameterized layers of operations. Each layer has operations involving the cost Hamiltonian  $H_C$  and a mixing Hamiltonian  $H_M$ , weighted by the angles  $\gamma$  and  $\beta$ , 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.



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

**Table 1.** Information about the three different problem instances used in simulations.



**Fig. 3.** 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).

## Benchmarking quantum approximate optimization for the heterogeneous vehicle routing problem

In this section, we show numerical results from noise-free simulations (exact state-vector simulations implemented in Python, without decoherence during the time evolution and without shot noise [finite-sampling effects] in the measurement results) 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. For these problem sizes, we can obtain the optimal solution from a classical brute-force search and compare it with the solutions obtained from the QAOA. Table 1 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 above. For the simulations, we consider realistic fuel consumption, gas prices, and fixed costs for each truck type, as detailed in the [Supplementary Information](#). A graphical representation of the problem instances is shown in Fig. 3.

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 solution. This stepwise approach, starting with the constraints [Eqs. (15), (16), and (25)] and then including the optimization part [(Eq. (14))], 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. We set the prefactors of the different parts of the Hamiltonian [Eqs. (15), (16), and (25)] to 1. The eigenvalues of the Hamiltonian are integers. This allows us to restrict the search space for  $\beta$  and  $\gamma$  in Eqs. (28)–(29) 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 (cost minimization) included [Eqs. (14)–(16) and Eq. (25)]. We rescale the cost function of the optimization [Eq. (14)] such that it only takes values between 0 and 1. Note that for this rescaling, we have to find the highest cost allowed by the cost function, which requires evaluating the cost for each possible solution in a brute-force approach. This is only feasible for small problem instances. In a real-world application, this rescaling procedure thus cannot be applied (although it may be possible to carry it out by estimating the highest possible cost). It may then be necessary to optimize over the penalty weights  $A$ ,  $B$ ,  $C$ , and  $D$  as well<sup>51,55</sup>. The eigenvalues of the full Ising Hamiltonian [Eq. (24)] are then no longer integers and thus we cannot set bounds for the variational parameters (in our simulations below, we initialize  $\gamma$  and  $\beta$  in the range  $[0, 2\pi]$ , but after optimization they may end up with values outside that range).

For real-world applications, a small probability for finding the optimal bitstring can still be sufficient, since we can efficiently evaluate the cost of the bitstring with a classical computer.

## Energy landscape

To illustrate the difficulty of finding good variational parameters for the QAOA, we show in Fig. 4 the energy landscape for  $p = 1$  for each problem instance, considering the full problem [Fig. 4a] and the capacity constraint in isolation [Fig. 4b]. 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, no such bounds can be set for the variational parameters, but for this visualization we constrain it to the range mentioned.

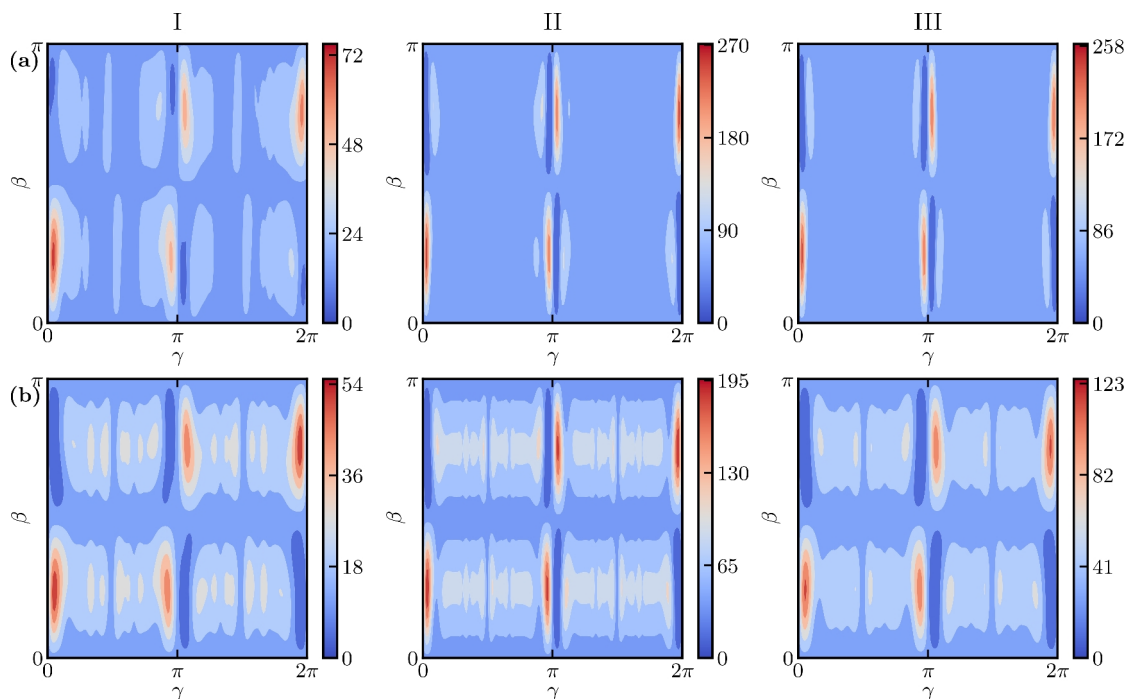
The states with the lowest energy are marked with dark blue in Fig. 4. Each surface plot shows four distinct optima. Moreover, the variational parameters are concentrated in the same region for all three problem instances, both for the full problem and for only the capacity constraint. The range for optimal parameters narrows with increasing problem size. Similar behaviour has been observed in several other studies, e.g., in Refs.<sup>56,57</sup>. 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 earlier in this article, the HVRP consists of two problems, a routing problem and a capacity problem. The latter is analogous to the constraints of the knapsack problem<sup>48</sup>. To better understand the energy landscapes for the capacity part of the problem instances shown in Fig. 4b, 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 Eq. (21). Figure 5 shows the (rapidly oscillating) energy landscape for this knapsack problem. It is clear that many optimizers may struggle to find the global optimum quickly 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. 4b, 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 1). 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<sup>58</sup>.

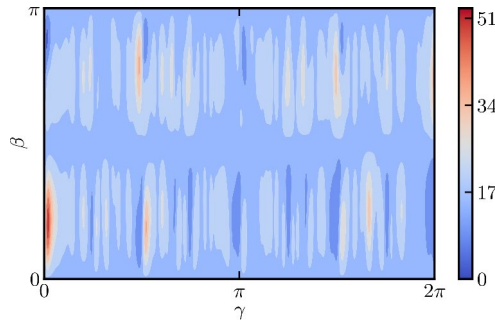
## Increasing the circuit depth

It has been shown that for a circuit depth of  $p = 1$ , the QAOA cannot outperform classical optimization algorithms<sup>9,59</sup>. For actual applications of the QAOA, it is therefore necessary to go to a circuit depth beyond  $p = 1$ .

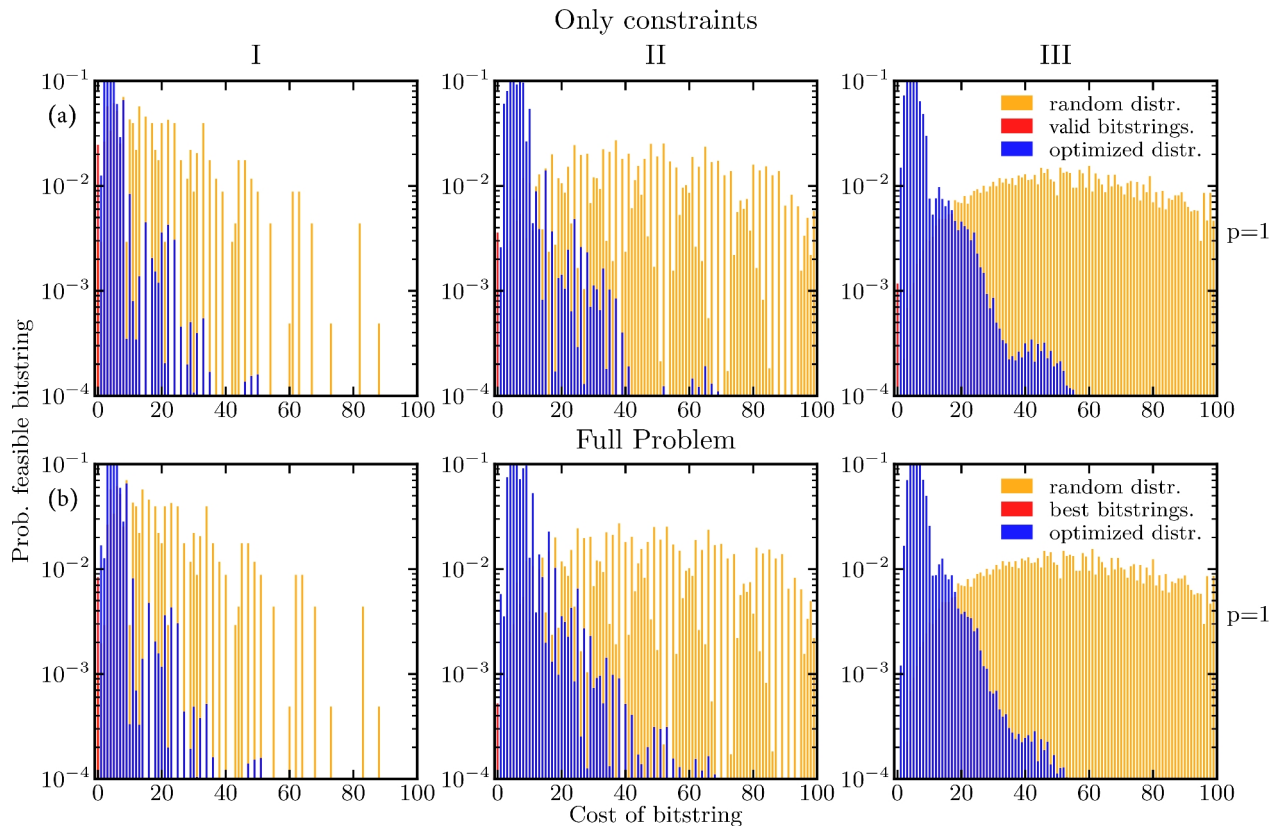
Before we investigate  $p > 1$ , we start with the lowest possible circuit depth of  $p = 1$ . In Fig. 6, we show histograms of the probability distributions created by the variational circuit for finding solutions satisfying all constraints, in the cases where the cost function encodes only the constraints and the full problem. We show the probability of sampling bitstrings with a specific cost (there can be several bitstrings leading to a particular



**Fig. 4.** Energy landscapes for the three problem instances with circuit depth  $p = 1$ . **(a)** The energy landscape for the full HVRP [Eqs. (14)–(16) and Eq. (25)]. The expectation value  $E(\gamma, \beta)$  for the total cost depends on the classically optimized angles  $\gamma$  and  $\beta$ . The periodicity is broken due to the non-integer eigenvalues for the cost Hamiltonian. **(b)** The energy landscape for the capacity constraint only [Eq. (25)]. Here, the expectation value  $E(\gamma, \beta)$  describes the energy penalty for breaking the capacity constraint.



**Fig. 5.** The energy landscape for the knapsack problem [see Eq. (19)] discussed in the main text with a circuit depth of  $p = 1$ . The energy landscape is highly non-convex and finding the global minimum quickly can be difficult for a classical optimizer.



**Fig. 6. (a)** The probability distribution of the optimized variational state (red, blue) for  $p = 1$ ; the goal of the optimization is to minimize the cost. 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 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.

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 the  $|+\rangle^{\otimes n}$  state, meaning all bitstrings are sampled with uniform probability (orange). The difference between the distributions in Fig. 6a and Fig. 6b is marginal. Thus, adding the cost term [Eq. (14)] 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 consider  $p \geq 1$ . In Figs. 7 and 8, we look at feasible solutions and the best solutions, respectively. We show how the variational state changes with increasing circuit depth from  $p = 1$  to  $p = 5$  for the three problem instances [see Fig. 3]. Additionally, we show for Fig. 7 an inset focusing on the low-energy part of the distribution of the variational state and for Fig. 8 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 25 % for  $p = 5$ . For the slightly larger instances (19 and 21 qubits), the probability of sampling a valid bitstring does not exceed 10 % (see Fig. 7). For the more difficult problem of actually finding the optimal tour (not just a feasible tour), the probability drops to 10 % for the 11-qubit instance, while for the 19- and 21-qubit instances the probability of sampling the ideal bitstring is around 1 % or below (see Fig. 8).

The inset in Fig. 8 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 the first excited energy eigenstate, which is an artifact of the rescaling of the cost, compared to the energy penalty for violating constraints, that 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<sup>51</sup>.

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 Figs. 7 and 8 as the probability distribution and the average energy (marked by the dashed black line in each panel) of the optimized state shifts increasingly to the left with increasing depth. It may 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 QAOA—the performance of the algorithm improves with more variational parameters (increasing depth). The drawback is that the optimization problem becomes increasingly difficult and time-consuming. It is therefore important to select a good optimization method.

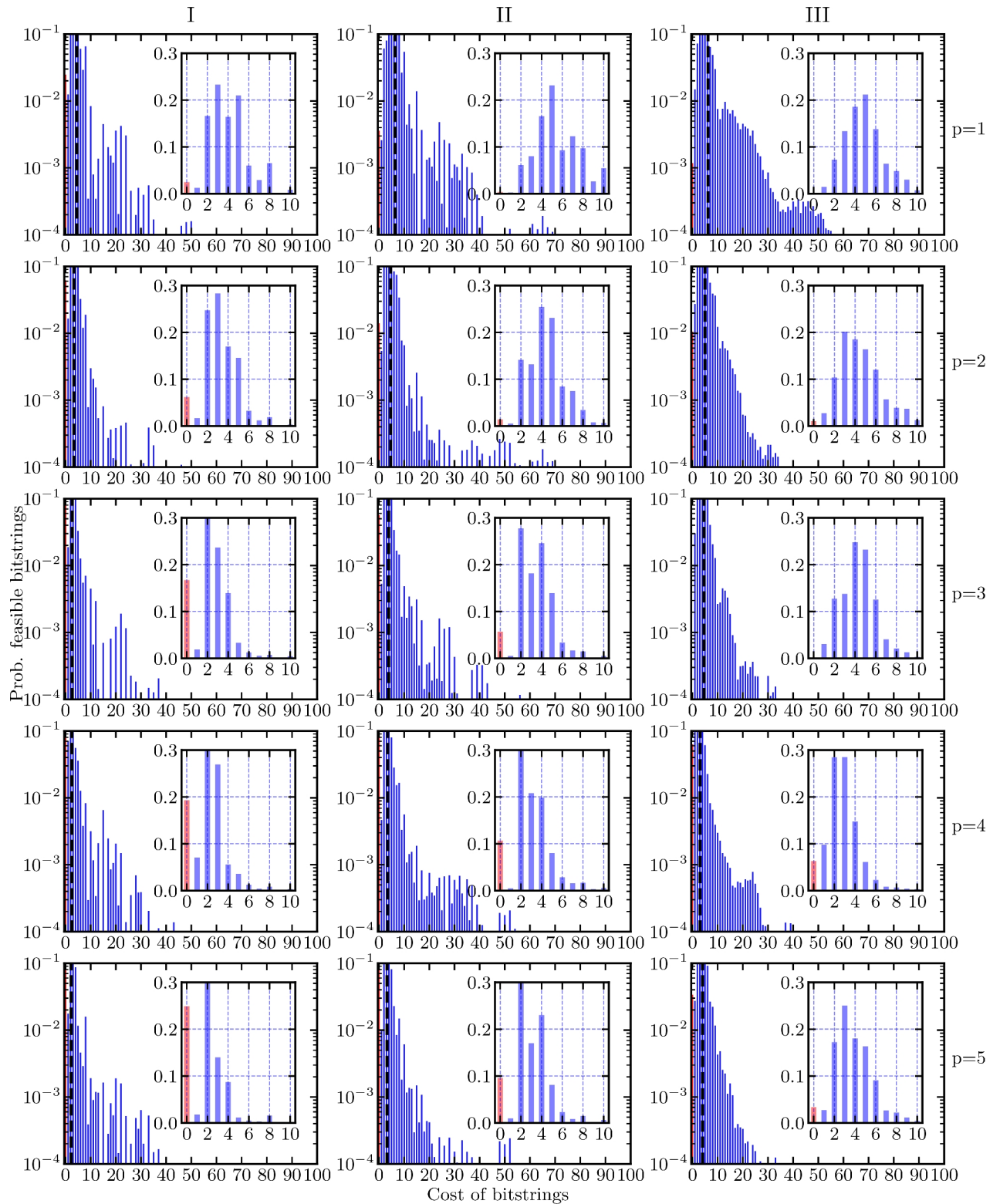
### Performance comparison of classical optimizers

Optimizers for finding the variational parameters play an important role in the context of variational quantum algorithms (VQAs). Much of the current research aims to find optimizers that perform well on such quantum circuits<sup>60–63</sup> and there is a rich literature proposing different optimizers for different problems<sup>3,64</sup>. In the following, we analyze the performance of simulations using four different well-known optimizers: Nelder–Mead<sup>65</sup>, Powell<sup>66</sup>, differential evolution<sup>67</sup>, and basinhopping<sup>68</sup>. The selected optimizers work differently: some use global search mechanisms, e.g., multiple random initial guesses (100 were used for differential evolution), 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.

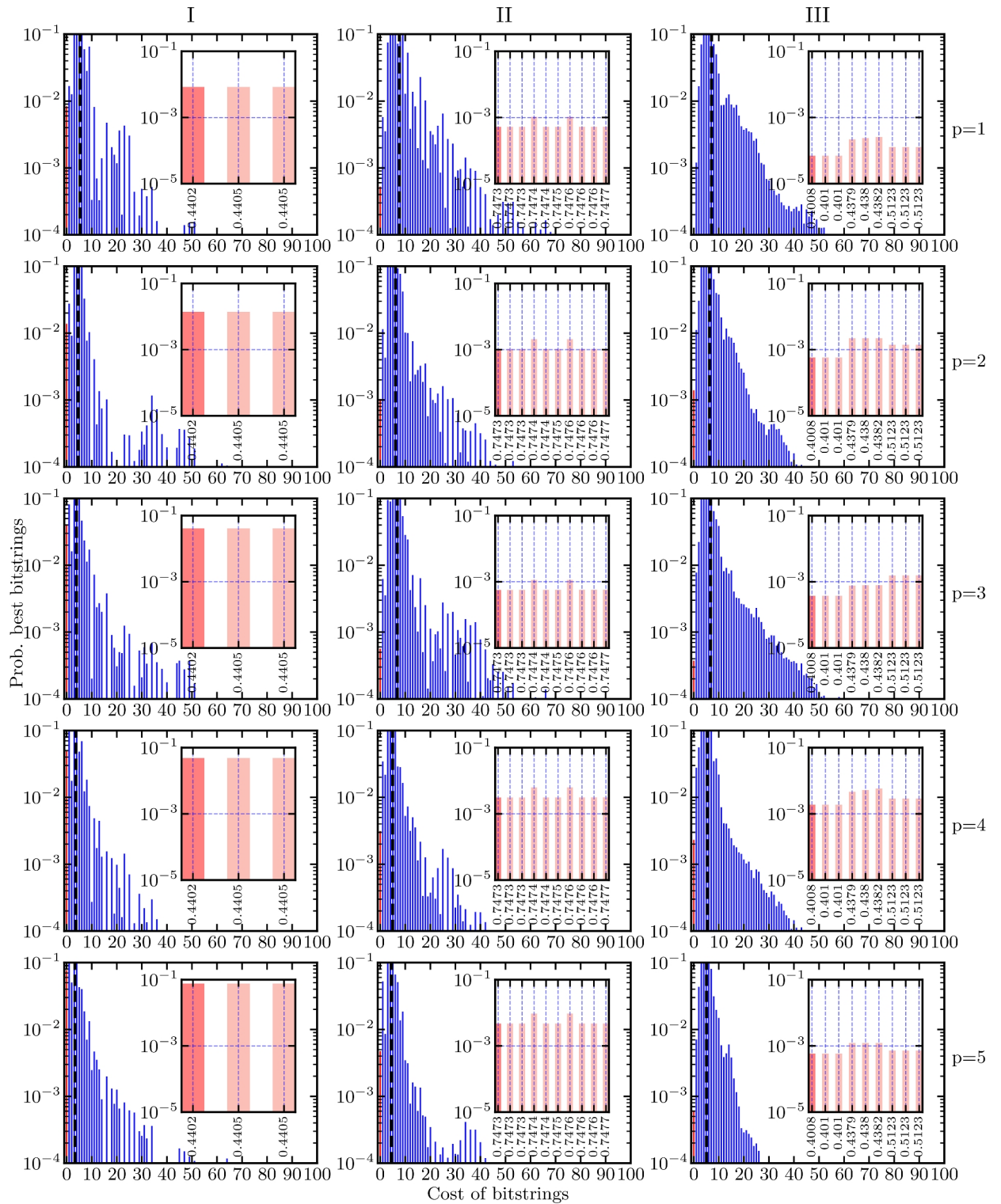
The Nelder–Mead method uses a geometrical shape called a simplex to search the function space. With each step of the optimization, the simplex shifts, ideally, towards the region with a minimum. The Nelder–Mead algorithm belongs to the class of gradient-free optimizers. The Powell optimizer works for 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 Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm<sup>69</sup> as a local optimizer. This algorithm approximates the descent direction by gradually improving the approximation of the Hessian matrix of the loss function determined by the (approximated) gradient. This framework has been proven useful for hard nonlinear optimization problems with multiple variables<sup>70</sup>.

To speed up the optimization routines, we use the optimized parameters from  $p - 1$  as an initial 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<sup>56,57</sup>. 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<sup>57</sup>. This can significantly speed up the optimization of the QAOA. In Fig. 9a, we show the success probability of finding a viable bitstring (giving the lowest energy for the Hamiltonian specifying the constraints) 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 (giving the lowest energy for the full Hamiltonian) as a function of the circuit depth  $p$  in Fig. 9b. One of the optimal tours is shown in Fig. 3. The other optimal solution is following the same path in reverse order.

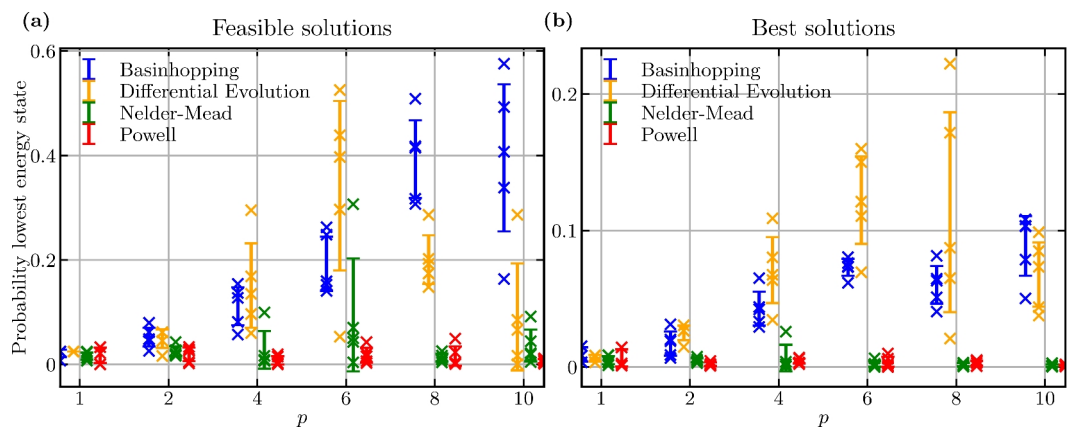
The results in Fig. 9 clearly favour the optimizers basinhopping and differential evolution. They perform significantly better than the Nelder–Mead and Powell optimization algorithms. Indeed, the Nelder–Mead and



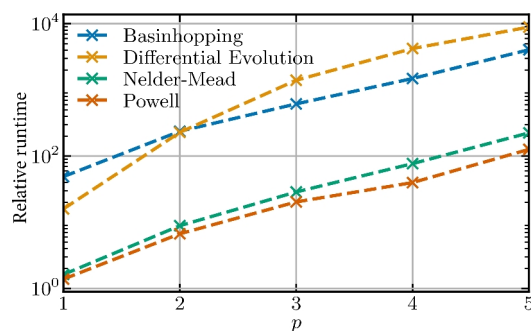
**Fig. 7.** The probability distribution of the optimized variational state  $|\gamma, \beta\rangle$  for the 11- (left), 19- (middle) and 21-qubit (right) problem instances as a function of the circuit depth  $p$  for obtaining feasible tours. The circuit depth increases from top  $p = 1$  to bottom  $p = 5$ , leading to a shift towards lower-energy eigenstates. The average energy of the variational state is marked by a dashed black line in each panel. The probability of sampling bitstrings that encode the optimal (lowest-energy) solution is marked with red. The simulations were conducted with the classical optimizer basinhopping with the local optimizer Broyden-Fletcher-Goldfarb-Shanno (BFGS) (see below).



**Fig. 8.** The probability distribution of the optimized variational state  $|\gamma, \beta\rangle$  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  $p = 5$ , leading to a shift towards lower-energy eigenstates. The average energy of the variational state is marked by a dashed black line in each panel. The probability of sampling the best (lowest-energy) bitstrings is marked with red. The binning is done in the same way as in Fig. 6b. 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 below).



**Fig. 9.** Comparing different optimizers for the HVRP on the 11-qubit problem instance shown in Fig. 3. 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  $p$ . For each optimizer and each  $p$ , 10 runs are performed, each from a random initialization and each result marked with a cross. The error bars indicate  $\pm$  one standard deviation around the mean of the 10 runs. The Nelder–Mead and Powell optimizers run until convergence. The differential evolution and basinhopping optimizers are capped at 1,000 and 50 iterations, respectively; their global search methods tend not to converge.



**Fig. 10.** Runtime comparison for the four classical optimizers tested in this work. The plot shows the wall-clock time for a full optimization run as a function of circuit depth  $p$  for the 11-qubit instance. A time unit, corresponding to a single optimization step of the Powell algorithm with  $p = 1$ , is 10 seconds.

Powell algorithms fail to improve the success probability when  $p$  increases, which suggests they are not suitable for QAOA. 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.

### Runtime comparison of the classical optimizers

For applications it is 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. 10.

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 algorithm runtime and performance becomes evident, as the slowest optimizers in terms of runtime performed best in terms of success probability (see Fig. 9). Moreover, the linear increase for all optimizers on the semi-log scale in Fig. 10 indicates that the amount of time needed for the optimization 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<sup>56,57</sup>. The results here further underline the importance of such research.

## 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 or a quantum annealer. 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 basin-hopping and differential-evolution algorithms seem well suited to optimize the variational parameters of the quantum circuit. However, this performance came at the expense of relatively long optimization times. 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. This 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<sup>58</sup>.

Moreover, Fig. 8 shows that the QAOA fails to distinguish solutions that satisfy all constraints but differ in cost. In approximate optimization we are not seeking the optimal solution, just any solution close to the optimum, and Fig. 8 shows that the QAOA successfully shifts the probability distribution towards lower-energy solutions. However, if there is no difference in probability between the worst solution satisfying all constraints and the best solution satisfying all constraints, the QAOA is not useful for the HVRP. The failure we observe may be due to the small energy gap between the different feasible solutions; as such, it 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.<sup>51</sup>.

## 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<sup>63,71,72</sup>. 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<sup>28,52</sup> 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<sup>11,73</sup>. 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. 5, 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<sup>74</sup>, which recently has been applied to the CVRP<sup>75</sup>. 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<sup>76</sup>. We can envision an extension to the so-called XY mixer that respects the constant Hamming weight in subsections of the solution bitstring. It can be noted here that the XY mixer has been investigated with noise taken into consideration<sup>73</sup>, and the numerical results in that work indicate that the presence of noise can cause the state to leave the subspace of allowed solutions. Even so, the algorithm might still lead to higher-quality solutions since the optimization does not need to explore every possible bit string. However, it should be noted that finding mixer Hamiltonians for general constraints has been shown to be an NP-complete problem<sup>77</sup>. Furthermore, both the QAOA and the quantum alternating operator ansatz have been shown to suffer from barren plateaus<sup>78</sup>, i.e., their optimization landscapes may become exponentially flat; it would be important to understand to what extent this is the case for the HVRP.

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.

## Data availability

Data is available from D.F. upon reasonable request.



Received: 5 March 2024; Accepted: 18 October 2024

Published online: 25 October 2024

## References

- Montanaro, A. Quantum algorithms: an overview. *npj Quantum Inf.* **2**, 15023. <https://doi.org/10.1038/npjqi.2015.23> (2016).
- Wendin, G. Quantum information processing with superconducting circuits: a review. *Rep. Prog. Phys.* **80**, 106001. <https://doi.org/10.1088/1361-6633/aa7e1a> (2017).
- Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625. <https://doi.org/10.1038/s42254-021-00348-9> (2021).
- Bharti, K. et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **94**, 015004. <https://doi.org/10.1103/RevModPhys.94.015004> (2022).
- Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505. <https://doi.org/10.1038/s41586-019-1666-5> (2019).
- Bruzewicz, C. D., Chiaverini, J., McConnell, R. & Sage, J. M. Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.* **6**, 021314. <https://doi.org/10.1063/1.5088164> (2019).
- Zhong, H.-S. et al. Quantum computational advantage using photons. *Science* **370**, 1460. <https://doi.org/10.1126/science.abe8770> (2020).
- Bluvstein, D. et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **626**, 58. <https://doi.org/10.1038/s41586-023-06927-3> (2024).
- Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. [arXiv:1411.4028](https://arxiv.org/abs/1411.4028) (2014).
- Koç, Ç., Bektaş, T., Jabali, O. & Laporte, G. Thirty years of heterogeneous vehicle routing. *Eur. J. Oper. Res.* **249**, 1. <https://doi.org/10.1016/j.ejor.2015.07.020> (2016).
- Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79. <https://doi.org/10.22331/q-2018-08-06-79> (2018).
- Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H. & Oliver, W. D. Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* **83**, 054401. <https://doi.org/10.1088/1361-6633/ab85b8> (2020).
- Golden, B. et al. (eds) *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43 (Springer, 2008).
- Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N. & Van Woensel, T. Two-echelon vehicle routing problems: a literature review. *Eur. J. Oper. Res.* **304**, 865–886. <https://doi.org/10.1016/j.ejor.2022.02.022> (2023).
- Caunhye, A. M., Nie, X. & Pokharel, S. Optimization models in emergency logistics: A literature review. *Socio-Econ. Plan. Sci.* **46**, 4–13. <https://doi.org/10.1016/j.seps.2011.04.004> (2012).
- Kurowski, K. et al. Application of quantum approximate optimization algorithm to job shop scheduling problem. *Eur. J. Oper. Res.* **310**, 518. <https://doi.org/10.1016/j.ejor.2023.03.013> (2023).
- Ghandriz, T., Jacobson, B., Islam, M., Heggren, J. & Laine, L. Transportation-mission-based optimization of heterogeneous heavy-vehicle fleet including electrified propulsion. *Energies* **14**, 3221. <https://doi.org/10.3390/en14113221> (2021).
- Lenstra, J. K. & Kan, A. H. G. R. Complexity of vehicle routing and scheduling problems. *Networks* **11**, 221–227. <https://doi.org/10.1002/net.3230110211> (1981).
- Laporte, G., Ropke, S. & Vidal, T. Chapter 4: Heuristics for the vehicle routing problem. In *Vehicle Routing: Problems, Methods, and Applications* 2nd edn (eds Toth, P. & Vigo, D.) 87–116 (Society for Industrial and Applied Mathematics, 2014). <https://doi.org/10.1137/1.9781611973594>.
- Tavares, L. G., Lopes, H. S. & Lima, C. R. E. Construction and improvement heuristics applied to the capacitated vehicle routing problem. In *2009 World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 690–695. <https://doi.org/10.1109/NABIC.2009.5393467> (IEEE, 2009).
- Hwang, C. P., Alidaee, B. & Johnson, J. D. A tour construction heuristic for the travelling salesman problem. *J. Oper. Res. Soc.* **50**, 797–809. <https://doi.org/10.2307/3010339> (1999).
- Stavropoulou, F., Repoussis, P. & Tarantilis, C. The vehicle routing problem with profits and consistency constraints. *Eur. J. Oper. Res.* **274**, 340–356. <https://doi.org/10.1016/j.ejor.2018.09.046> (2019).
- Van Breedam, A. Improvement heuristics for the vehicle routing problem based on simulated annealing. *Eur. J. Oper. Res.* **86**, 480–490. [https://doi.org/10.1016/0377-2217\(94\)00064-J](https://doi.org/10.1016/0377-2217(94)00064-J) (1995).
- Brandstätter, C. & Reimann, M. Performance analysis of a metaheuristic algorithm for the line-haul feeder vehicle routing problem. *J. Veh. Rout. Algorithms* **1**, 121–138. <https://doi.org/10.1007/s41604-018-0009-2> (2018).
- Clarke, G. & Wright, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12**, 568–581 (1964).
- Lougee-Heimer, R. The Common Optimization Interface for Operations Research: Promoting open-source software in the operations research community. *IBM J. Res. Dev.* **47**, 57–66. <https://doi.org/10.1147/rd.471.0057> (2003).
- Gröer, C., Golden, B. & Wasil, E. A library of local search heuristics for the vehicle routing problem. *Math. Program. Comput.* **2**, 79–101. <https://doi.org/10.1007/s12532-010-0013-5> (2010).
- Perron, L. & Furnon, V. Or-tools (2022).
- Baldacci, R. & Mingozzi, A. A unified exact method for solving different classes of vehicle routing problems. *Math. Program.* **120**, 347–380. <https://doi.org/10.1007/s10107-008-0218-9> (2009).
- Kim, Y. et al. Evidence for the utility of quantum computing before fault tolerance. *Nature* **618**, 500. <https://doi.org/10.1038/s41586-023-06096-3> (2023).
- Feld, S. et al. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front. ICT* **6**, 13. <https://doi.org/10.3389/fict.2019.00013> (2019).
- Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242. <https://doi.org/10.1038/nature23879> (2017).
- McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C. & Yuan, X. Quantum computational chemistry. *Rev. Mod. Phys.* **92**, 015003. <https://doi.org/10.1103/RevModPhys.92.015003> (2020).
- Egger, D. J. et al. Quantum computing for finance: state-of-the-art and future prospects. *IEEE Trans. Quantum Eng.* **1**, 3101724. <https://doi.org/10.1109/TQE.2020.3030314> (2020).
- Vikstål, P. et al. Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Phys. Rev. Appl.* **14**, 034009. <https://doi.org/10.1103/PhysRevApplied.14.034009> (2020).
- Phillipson, F. Quantum computing in logistics and supply chain management—an overview. [arXiv:2402.17520](https://arxiv.org/abs/2402.17520) (2024).
- Syrichas, A. & Crispin, A. Large-scale vehicle routing problems: Quantum annealing, tunings and results. *Comput. Oper. Res.* **87**, 52. <https://doi.org/10.1016/j.cor.2017.05.014> (2017).
- Harikrishnakumar, R., Nannapaneni, S., Nguyen, N. H., Steck, J. E. & Behrman, E. C. A Quantum annealing approach for dynamic multi-depot capacitated vehicle routing problem. [arXiv:2005.12478](https://arxiv.org/abs/2005.12478) (2020).
- Harwood, S. et al. Formulating and solving routing problems on quantum computers. *IEEE Trans. Quantum Eng.* **2**, 3100118. <https://doi.org/10.1109/TQE.2021.3049230> (2021).
- Azad, U., Behera, B. K., Ahmed, E. A., Panigrahi, P. K. & Farouk, A. Solving vehicle routing problem using quantum approximate optimization algorithm. *IEEE Trans. Intell. Transport. Syst.* **24**, 7564. <https://doi.org/10.1109/TITS.2022.3172241> (2023).
- Otterbach, J. S. et al. Unsupervised machine learning on a hybrid quantum computer. [arXiv:1712.05771](https://arxiv.org/abs/1712.05771) (2017).

42. Bengtsson, A. et al. Improved success probability with greater circuit depth for the quantum approximate optimization algorithm. *Phys. Rev. Appl.* **14**, 034010. <https://doi.org/10.1103/PhysRevApplied.14.034010> (2020).
43. Willsch, M., Willsch, D., Jin, F., De Raedt, H. & Michielsen, K. Benchmarking the quantum approximate optimization algorithm. *Quantum Inf. Process.* **19**, 197. <https://doi.org/10.1007/s11128-020-02692-8> (2020).
44. Harrigan, M. P. et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **17**, 332. <https://doi.org/10.1038/s41567-020-01105-y> (2021).
45. Qian, W. et al. Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem. *Entropy* **25**, 1238. <https://doi.org/10.3390/e25081238> (2023).
46. Ising, E. Beitrag zur Theorie des Ferromagnetismus. *Z. Phys.* **31**, 253. <https://doi.org/10.1007/BF02980577> (1925).
47. Brush, S. G. History of the Lenz-Ising model. *Rev. Mod. Phys.* **39**, 883. <https://doi.org/10.1103/RevModPhys.39.883> (1967).
48. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 5. <https://doi.org/10.3389/fphy.2014.00005> (2014).
49. Baldacci, R., Battarra, M. & Vigo, D. *Routing a Heterogeneous Fleet of Vehicles* 3–27 (Springer, 2008).
50. Gheysens, F., Golden, B. L. & Assad, A. A. A comparison of techniques for solving the fleet size and mix vehicle routing problem. *Oper. Res. Spektrum* **6**, 207–216. <https://doi.org/10.1007/BF01720070> (1984).
51. Roch, C. et al. Cross entropy hyperparameter optimization for constrained problem hamiltonians applied to QAOA. In *2020 International Conference on Rebooting Computing (ICRC)*, 50–57. <https://doi.org/10.1109/ICRC2020.2020.00009> (2020).
52. Uchoa, E. et al. New benchmark instances for the capacitated vehicle routing problem. *Eur. J. Oper. Res.* **257**, 845–858. <https://doi.org/10.1016/j.ejor.2016.08.012> (2017).
53. Barman, S. E., Lindroth, P. & Strömberg, A.-B. Modeling and Solving Vehicle Routing Problems with Many Available Vehicle Types. In *Optimization, control, and applications in the information age* (eds Migdalas, A. & Karakitsiou, A.) 113–138 (Springer, 2015). [https://doi.org/10.1007/978-3-319-18567-5\\_6](https://doi.org/10.1007/978-3-319-18567-5_6).
54. Sun, Y., Zhang, J.-Y., Byrd, M. S. & Wu, L.-A. Adiabatic quantum simulation using trotterization. [arXiv:1805.11568](https://arxiv.org/abs/1805.11568) (2018).
55. Svensson, M. et al. A hybrid quantum-classical heuristic to solve large-scale integer linear programs. [arXiv:2103.15433](https://arxiv.org/abs/2103.15433) (2021).
56. Streif, M. & Leib, M. Training the quantum approximate optimization algorithm without access to a quantum processing unit. *Quantum Sci. Technol.* **5**, 034008. <https://doi.org/10.1088/2058-9565/ab8c2b> (2020).
57. Akshay, V., Rabinovich, D., Campos, E. & Biamonte, J. Parameter concentrations in quantum approximate optimization. *Phys. Rev. A* **104**, L010401. <https://doi.org/10.1103/PhysRevA.104.L010401> (2021).
58. de la Grandrive, P. D. & Hullo, J.-F. Knapsack problem variants of QAOA for battery revenue optimisation. [arXiv:1908.02210](https://arxiv.org/abs/1908.02210) (2019).
59. Farhi, E., Gamarnik, D. & Gutmann, S. The quantum approximate optimization algorithm needs to see the whole graph: worst case examples. [arXiv:2005.08747](https://arxiv.org/abs/2005.08747) (2020).
60. Barkoutsos, P. K., Nannicini, G., Robert, A., Tavernelli, I. & Woerner, S. Improving variational quantum optimization using CVaR. *Quantum* **4**, 256. <https://doi.org/10.22331/q-2020-04-20-256> (2020).
61. Kübler, J. M., Arrasmith, A., Cincio, L. & Coles, P. J. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum* **4**, 263. <https://doi.org/10.22331/q-2020-05-11-263> (2019).
62. Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* **37**, 332–341. <https://doi.org/10.1109/9.119632> (1992).
63. Verdon, G. et al. Learning to learn with quantum neural networks via classical neural networks. [arXiv:1907.05415](https://arxiv.org/abs/1907.05415) (2019).
64. Lavrijsen, W., Tudor, A., Müller, J., Iancu, C. & de Jong, W. Classical optimizers for noisy intermediate-scale quantum devices. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 267–277. <https://doi.org/10.1109/QCE49297.2020.00041> (2020).
65. Nelder, J. A. & Mead, R. A simplex method for function minimization. *Comput. J.* **7**, 308–313. <https://doi.org/10.1093/comjnl/7.4.308> (1965).
66. Powell, M. J. D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7**, 155–162. <https://doi.org/10.1093/comjnl/7.2.155> (1964).
67. Storn, R. & Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359. <https://doi.org/10.1023/A:1008202821328> (1997).
68. Wales, D. & Doye, J. Global optimization by Basin-Hopping and the lowest energy structures of Lennard–Jones clusters containing up to 110 atoms. *J. Phys. Chem. A* **101**, 5111–5116. <https://doi.org/10.1021/jp970984n> (1998).
69. Frédéric Bonnans, J., Gilbert, J. C., Lemaréchal, C. & Sagastizábal, C. A. *Numerical Optimization: Theoretical and Practical Aspects* (Springer, 2006).
70. Olson, B., Hashmi, I., Molloy, K. & Shehu, A. Basin hopping as a general and versatile optimization framework for the characterization of biological macromolecules. *Adv. Artif. Intell.* **2012**, 674832. <https://doi.org/10.1155/2012/674832> (2012).
71. Yao, J., Bukov, M. & Lin, L. Policy gradient based quantum approximate optimization algorithm. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference, Proceedings of Machine Learning Research*, vol. 107 (eds Lu, J. & Ward, R.) 605–634 (PMLR, 2020).
72. Garcia-Saez, A. & Riu, J. Quantum observables for continuous control of the quantum approximate optimization algorithm via reinforcement learning. [arXiv:1911.09682](https://arxiv.org/abs/1911.09682) (2019).
73. Streif, M., Leib, M., Wudarski, F., Rieffel, E. & Wang, Z. Quantum algorithms with local particle-number conservation: Noise effects and error correction. *Phys. Rev. A* **103**, 042412. <https://doi.org/10.1103/PhysRevA.103.042412> (2021).
74. Hadfield, S. et al. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34. <https://doi.org/10.3390/a12020034> (2019).
75. Xie, N. et al. A feasibility-preserved quantum approximate solver for the capacitated vehicle routing problem. [arXiv:2308.08785](https://arxiv.org/abs/2308.08785) (2023).
76. Wang, Z., Rubin, N. C., Dominy, J. M. & Rieffel, E. G. XY mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* **101**, 012320. <https://doi.org/10.1103/PhysRevA.101.012320> (2020).
77. Leipold, H. & Spedalieri, F. M. Constructing driver Hamiltonians for optimization problems with linear constraints. *Quantum Sci. Technol.* **7**, 015013. <https://doi.org/10.1088/2058-9565/ac16b8> (2021).
78. Larocca, M. et al. A review of barren plateaus in variational quantum computing. [arXiv:2405.00781](https://arxiv.org/abs/2405.00781) (2024).

## Acknowledgements

The computations were enabled by resources provided by Chalmers e-Commons at Chalmers University of Technology. D.F., M.G., and A.F.K. acknowledge financial support from the Knut and Alice Wallenberg foundation through the Wallenberg Centre for Quantum Technology (WACQT). AFK is also supported by the Swedish Research Council (grant number 2019-03696), the Horizon Europe programme HORIZON-CL4-2022-QUANTUM-01-SGA via the project 101113946 OpenSuperQPlus100, and the Swedish Foundation for Strategic Research.

### Author contributions

D.F. performed all numerical simulations. D.F. derived the analytical expressions with support from M.G. and input from all authors. D.F., M.G., and A.F.K. analyzed the results with input from T.G. and L.L. D.F. and A.F.K. wrote the manuscript with input from all authors. L.L., M.G., and A.F.K. supervised the work.

### Funding

Open access funding provided by Chalmers University of Technology.

### Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-76967-w>.

**Correspondence** and requests for materials should be addressed to D.F., M.G. or A.F.K.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024