Benchmarking Variational Quantum Algorithms for Combinatorial Optimization in Practice

Tim Schwägerl[1,2*], Yahui Chai[1†], Tobias Hartung[3,4†], Karl Jansen[1,5†], Stefan Kühn[1†]

[1]Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, Zeuthen, 15738, Brandenburg, Germany.
[2]Institut für Physik, Humboldt-Universität zu Berlin, Newtonstr. 15, Berlin, 12489, Berlin, Germany.
[3]Devon House, Northeastern University London, 58 St Katharine's Way, London, E1W 1LP, United Kingdom.
[4]Khoury College of Computer Sciences, Northeastern University, 440 Huntington Ave, Boston, 02115, Massachusetts, USA.
[5]Computation-Based Science and Technology Research Center, The Cyprus Institute, 20 Kavafi Street, Nicosia, 2121, Cyprus.

*Corresponding author(s). E-mail(s): tim.schwaegerl@desy.de;
Contributing authors: yahui.chai@desy.de;
tobias.hartung@nulondon.ac.uk; karl.jansen@desy.de;
stefan.kuehn@desy.de;
†These authors contributed equally to this work.

Abstract

Variational quantum algorithms, and in particular variants of the variational quantum eigensolver, have been proposed as approaches to combinatorial optimization (CO) problems. With only shallow ansatz circuits, these methods are considered suitable for current noisy intermediate-scale quantum hardware. Yet, the resources required to train variational circuits often scale superpolynomially with problem size. In this study, we numerically investigate the practical implications of this scaling for CO problems, using Max-Cut and random QUBO instances as benchmarks. For a fixed computational budget, we compare the average performance of training shallow variational circuits, sampling with replacement, and greedy local search. We identify the minimum problem size at which the quantum algorithms consistently outperform random sampling, and

for each size, we characterize their separation from greedy local search. Beyond average-case performance, we analyze correlations between algorithms across individual problem instances. These results strengthen the case for intuitive yet meaningful benchmarks of variational quantum algorithms for CO problems under realistic resource constraints.

Keywords: Variational quantum algorithms, combinatorial optimization, measurement shot noise

1 Introduction

The goal of combinatorial optimization (CO) is to find an optimal or near-optimal solution from a finite set of candidates. CO problems have many practically relevant applications in industry, such as supply chain optimization [1], logistics [2], and chip design [3]. They also arise in physics, for example, in reconstructing particle trajectories in collider experiments [4–10]. Many CO problems are NP-hard [11], meaning that no efficient classical algorithms are known for solving them in general. In practice, they are routinely addressed with heuristic classical algorithms that provide approximate solutions. Even small improvements in accuracy or runtime can have substantial impact, which motivates the continuing search for better algorithms.

Quantum computation offers the potential for algorithmic speedups, with proven advantages for specific problems such as factoring [12] and unstructured search [13]. Whether similar advantages can be achieved for approximating CO problems remains an open question, and is the focus of many theoretical and empirical studies. In the era of noisy intermediate-scale quantum (NISQ) devices [14], most approaches to CO take the form of variational quantum algorithms (VQAs)—hybrid quantum-classical methods that train the parameters of ansatz circuits within the constraints of near-term hardware. Prominent examples include the quantum approximate optimization algorithm (QAOA) [15], the variational quantum eigensolver (VQE) [16], and numerous extensions of both families [17, 18]. Alongside these developments, many proof-of-principle demonstrations and benchmarking studies of VQAs for CO have been reported [19–24].

VQAs often suffer from trainability issues, due to cost concentration [25, 26] and poor local minima [27]. This implies that the number of samples required to train the quantum circuit scales superpolynomially with problem size. In this study we numerically investigate the practical implications of this scaling for solving small instances of the prominent Max-Cut problem on 3-regular graphs and random quadratic unconstrained binary optimization (QUBO) instances. We compare the average performance of different algorithms that are all based on repeatedly evaluating the objective function, which allows for a direct comparison: training a variational quantum circuit, sampling with replacement, and a greedy local search. Our primary motivation for these benchmarks is to provide an intuitive point of reference between VQAs and simple classical routines. Many prior studies evaluate VQA performance in isolation, which makes it challenging to interpret the results in context. Our results contribute to the

understanding of average-case performance metrics for VQAs, such as the approximation ratio and the success probability. Furthermore, we analyze whether good initial points for the greedy algorithm are also good initial points for the VQA by studying the correlation between the performance of both algorithms on individual instances.

This paper is organized as follows. In Sec. 2.1, we describe the construction of the CO problem instances used in our benchmarks. The performance metrics for comparing the different algorithms are introduced in Sec. 2.2, including a metric that provides intuition for the average difference in approximation ratio and a binned-statistics approach to investigate correlations in algorithm performance by instance. Section 3 details the specific VQAs considered in this study, together with the greedy local search and the baseline sampling procedure. We also clarify what is meant by initializing the VQA and the greedy algorithm from the same starting point, which enables meaningful instance-wise comparisons. A brief overview of related work is given in Sec. 4. The setup of our numerical experiments and the corresponding results are presented in Sec. 5. Finally, Sec. 6 concludes with a discussion.

2 Combinatorial optimization

In the following we briefly introduce the Max-Cut problem and the random QUBO instances that we use as a benchmark, and discuss the various performance metrics for characterizing the different algorithms.

2.1 Max-Cut and random QUBO instances

The goal of the Max-Cut problem is to divide the nodes V of an undirected weighted graph $G = (V, E)$ with edges E into two sets such that the sum of the edge weights between the two sets is maximal. Formally, for a graph $G = (V, E)$ with $|V| = N + 1$ nodes and edge weights $w_{ij} > 0$ for $(i, j) \in E$ the problem is to maximize the objective function

$$\tilde{\tilde{O}}(\boldsymbol{x}) = \sum_{(i,j)\in E} w_{ij}[x_i(1 - x_j) + x_j(1 - x_i)]. \tag{1}$$

by assigning $x_i = 0$ or $x_i = 1$ to each node i. We follow the convention $j < i$ for labeling the edges $(i, j) \in E$. An assignment $\boldsymbol{x} = (x_1, \ldots, x_{N+1})$ divides the graph into two sets of nodes according to their labels. Maximizing the objective function corresponds to maximizing the sum of the edge weights between the two sets. The problem has a symmetry under interchanging the labels $0 \to 1$ and $1 \to 0$. Following Ref. [28], we remove this symmetry by setting $x_1 = 0$, leading to the objective function

$$\tilde{O}(\boldsymbol{x}) = \sum_{(i,j=1)\in E} w_{ij}x_i + \sum_{(i\neq 1, j\neq 1)\in E} w_{ij}[x_i(1 - x_j) + x_j(1 - x_i)]. \tag{2}$$

The Max-Cut problem on a graph with $N + 1$ nodes is now encoded as an objective function of N binary variables. Finding the optimal solution $\boldsymbol{x}^* = \operatorname{argmax} \tilde{\tilde{O}}(\boldsymbol{x})$ is NP-hard. Finding a solution $\boldsymbol{x}$ with approximation ratio $\alpha = \tilde{\tilde{O}}(\boldsymbol{x})/\tilde{\tilde{O}}(\boldsymbol{x}^*) > 16/17 \approx$

0.9412 is also NP-hard [29]. This makes Max-Cut a suitable problem for studying algorithms for CO. The Goemans-Williamson (GW) algorithm, the best-known classical semidefinite programming (SDP) algorithm for Max-Cut, can find a solution with approximation ratio of $\alpha \geq 0.87856$ in polynomial time [30].

We follow a similar approach as in Ref. [28] to construct Max-Cut instances for our numerical experiments. For each $N \in \{11, 21, 31, 41, 51, 61\}$, we define 25 Max-Cut instances on random 3-regular simple undirected and connected graphs with weights w_{ij} drawn uniformly from $(0, 1]$. We divide the objective function $\tilde{O}$ of every instance by the objective value O_{GW} achieved by a single run of the GW SDP algorithm using its implementation in Qiskit [31]. This defines the objective function O we use in our numerical experiments:

$$O(\boldsymbol{x}) = \frac{\tilde{O}(\boldsymbol{x})}{O_{GW}}. \tag{3}$$

Now, the objective functions are re-scaled to intervals $[0, \beta]$ where the lower bound 0 is given by the trivial cut $\boldsymbol{x} = 0$. The objective function O is the same as the ratio of the corresponding approximation ratios. Since the GW algorithm gives an approximation ratio of 0.87856 in the worst case and an approximation ratio of 1 in the best case, it is evident that $1 \leq \beta \leq 1/0.87856$. This procedure eliminates possible impacts of differing scales of the objective function on the performance of the algorithms we benchmark.

In addition to Max-Cut, we also consider random QUBO instances [32]. A QUBO problem is specified by an $N \times N$ real symmetric matrix Q and the objective function

$$O_{\mathrm{QUBO}}(\boldsymbol{x}) = -\boldsymbol{x}^\top Q \boldsymbol{x}, \quad \boldsymbol{x} \in \{0, 1\}^N. \tag{4}$$

Throughout this work, we adopt the convention of formulating QUBO as a maximization problem, consistent with our treatment of Max-Cut. Max-Cut can be expressed as a special case of QUBO by choosing the matrix elements Q_{ij} to encode the edge weights of a graph. More generally, QUBO provides a unifying formulation for a broad class of combinatorial optimization problems.

In contrast to Max-Cut, random QUBO instances do not come with hardness-of-approximation guarantees. While solving QUBO is NP-hard in the worst case, the approximability of random instances depends strongly on the distribution from which Q is drawn.

For our numerical experiments, we generate random QUBO instances by drawing the upper-triangular elements Q_{ij} independently from the uniform distribution on $[-1, 1]$ and symmetrizing Q. These instances complement the structured Max-Cut problems and allow us to investigate how algorithm performance differs between problems with established approximation hardness and a problem that lacks such guarantees.

2.2 Performance metrics

To quantify an algorithm's ability to find an approximate solution $\boldsymbol{x}_{\max}$ for the CO problems, we use the approximation ratio:

$$\alpha = \frac{O(\boldsymbol{x}_{\max})}{O(\boldsymbol{x}^*)}. \tag{5}$$

Throughout each algorithm that we run, we keep track of the assignment that produced the highest approximation ratio. The final approximation ratio is then defined as the highest approximation ratio observed during the runtime of the algorithm. In particular, for the case of VQAs we do not use the expected value of the objective function to determine the approximation ratio, which would correspond to an average over multiple assignments, if the current ansatz state does not represent a computational basis state. Instead, we use the best assignment $\boldsymbol{x}_{\max}$ obtained by the measurements throughout the VQA process. The approximations ratios achieved by the VQAs in our numerical experiments follow a highly non-normal distribution over problem instances and initial points. This means that its mean value does not represent the approximation ratio of a typical VQA instance. Furthermore, the sample standard deviation does not include most VQA instances. Thus, we compute the standard error of the mean (SEM) instead of the sample standard deviation, rating how well the sample mean represents the population mean. To aid in understanding what a certain average difference in the approximation ratio between two algorithms means, we use an additional statistical method. We compute mean probabilities and, using Wilson's score method [33] implemented in Python's statsmodels module [34], 95% confidence intervals for an algorithm achieving a higher approximation ratio than another algorithm. This metric does not include information about the value of the difference but gives an intuitive understanding complementing the average difference in approximation ratio. In this metric, a value of $1/2$ means that the algorithms perform equally well.

For small problems, the VQAs are able to find the exact solution. To quantify its ability to do so, we count the number of successful runs where success is defined as observing the exact solution at least once when running the algorithm. Then, we compute the mean success probability and, again using Wilson's score method, the 95% confidence interval for success. When benchmarking the VQAs, this metric is only useful for small problems because it decays rapidly with problem size. For larger problems, one cannot expect to find the exact solution and has to rely on the approximation ratio as a performance metric.

To obtain an understanding of the algorithms beyond the typical average case studies, we analyze the correlation of the approximation ratios achieved by different algorithms by instance. We do so using binned statistics on the instances as outlined in the following. An instance is defined by the problem instance and, if the algorithm accepts an initial point, by the initial point. First, we compute the approximation ratios achieved by the algorithms for every instance. Then, we group the instances into small intervals of equal size in approximation ratio achieved by one algorithm. The x-value is defined by the sample mean of these approximation ratios and its sample

standard deviation. Now, the y-value is given by the sample mean and standard deviation of the approximation ratio achieved by another algorithm for the same instances. This method allows for investigating if instances that are hard for one algorithm are also hard for another algorithm.

3 Algorithms

In this section we introduce the algorithms we compare in our study: training a shallow parametric quantum circuits, sampling with replacement, and a greedy local search starting either from uniformly random states or from the same initial point as the quantum algorithm.

3.1 Variational quantum eigensolver

The VQE is a hybrid quantum-classical algorithm that was originally proposed for computing ground states of molecular Hamiltonians [16]. The expectation value of the Hamiltonian is evaluated on a quantum device using a parametrized circuit as a variational ansatz. The parameters are then updated utilizing a classical optimization algorithm such that the Hamiltonian's expectation value decreases. Running this quantum-classical feedback loop iteratively until certain convergence criteria are matched or when a maximum number of N_{iter} iterations is reached, one obtains an approximation for the ground state of the Hamiltonian. In our study, we apply the VQE to maximize an objective function, which can equivalently be expressed as minimizing the expectation value of a corresponding Hamiltonian.

In the context of CO, an alternative view on the algorithm is preferable because the solution candidates are computational basis states. The quantum circuit M, which we assume to include projective measurements in the computational basis at the end, is a model that maps sets of N_{params} parameters $\boldsymbol{\vartheta}$ to computational basis states of N qubits. Here we consider without loss of generality parametric gates of the form $\exp(i\vartheta_k \mathcal{P})$, where $\mathcal{P}$ is a Pauli string on N qubits, $\mathcal{P} \in \{\mathbb{1}, X, Y, Z\}^{\otimes N}$. Hence, the parameters can be restricted to $[0, 2\pi)$, and the quantum circuit corresponds to

$$M : [0, 2\pi)^{N_{\text{params}}} \to \{0, 1\}^N. \tag{6}$$

The cost function C, the target of the classical optimization algorithm, computes a single cost value for N_{shots} computational basis states

$$C : (\{0, 1\}^N)^{N_{\text{shots}}} \to \mathbb{R}. \tag{7}$$

The classical optimization algorithm OA uses this value and the corresponding set of parameters to calculate a new set of parameters [1]

$$OA : [0, 2\pi)^{N_{\text{params}}} \times \mathbb{R} \to [0, 2\pi)^{N_{\text{params}}}. \tag{8}$$

[1] Note that for simplicity of notation, we assume the optimization algorithm only takes the parameter values and the current cost function value as in input. Additional arguments, which would be required, e.g., for gradient-based optimization algorithms would not affect any of the arguments presented in the following.

The VQE algorithm in terms of these definitions is described in Algorithm 1.

Algorithm 1 Variational Quantum Eigensolver

Input: Quantum circuit M, initial parameter vector $\boldsymbol{\vartheta}$, objective function O, cost function C, optimization algorithm OA
Parameter: Maximum number of iterations N_{iter}, maximum number of shots N_{shots}
Output: Binary vector $\boldsymbol{x}_{\text{max}} \in \{0,1\}^N$

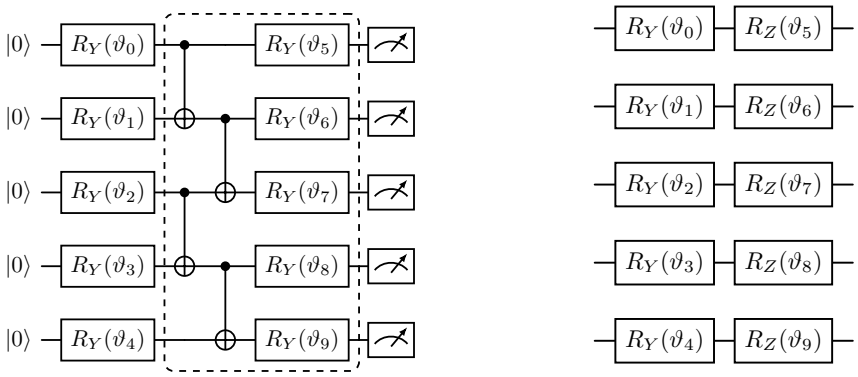
$O_{\text{max}} \leftarrow -\infty$
for iteration $\leftarrow 1$ to N_{iter} **do**
    $X \leftarrow$ empty list
    for shot $\leftarrow 1$ to N_{shots} **do**
        $\boldsymbol{x}_{\text{update}} \leftarrow M(\boldsymbol{\vartheta})$
        Append $\boldsymbol{x}_{\text{update}}$ to X
        $O_{\text{update}} \leftarrow O(\boldsymbol{x}_{\text{update}})$
        if $O_{\text{update}} > O_{\text{max}}$ **then**
            $\boldsymbol{x}_{\text{max}} \leftarrow \boldsymbol{x}_{\text{update}}$
            $O_{\text{max}} \leftarrow O_{\text{update}}$
        end if
    end for
    $\boldsymbol{\vartheta} \leftarrow OA(\boldsymbol{\vartheta}, C(X))$
end for

3.2 Hyper-parameter choices for the VQE

In our study we investigate the performance of simple hardware-efficient circuit architectures that are suitable for noisy intermediate-scale quantum hardware. The circuit starts with a register of N qubits encoding N binary variables initialized in the state $|0\rangle^{\otimes N}$. Then, a single layer of parameterized R_Y rotation gates acts on the qubits. The exclusive use of R_Y rotations results in a low gate count and a circuit that generates real amplitudes. The initial layer is sufficient to express every computational basis state but the corresponding cost landscape suffers from numerous local minima. To generate a possibly more favorable cost landscape, entangling CNOT gates are added in a brick-like pattern as shown in Fig. 1a. This pattern enables the execution of multiple CNOT operations in parallel. The circuit is completed with another layer of R_Y rotations and a measurement of all qubits in the computational basis.

As a more expressive alternative, we also consider the EfficientSU2 circuit, which extends the previously described RealAmplitudes ansatz by including additional R_Z rotations after each R_Y gate in the rotation layers. This augmentation allows the circuit to explore a larger portion of Hilbert space, while still maintaining a hardware-efficient structure with brick-like entangling layers. An example of this rotation layer is illustrated in Fig. 1b. As with the RealAmplitudes ansatz, our study focuses on a single layer, although deeper variants can be constructed by repeating the same block.

Since the VQE was proposed to estimate ground-state energies, its original cost function is the energy expectation value. In the context of CO, this would lead to



(a) RealAmplitudes ansatz with a pairwise entangling layer.

(b) Rotation layer of the EfficientSU2 ansatz.

Fig. 1: Two commonly used hardware-efficient quantum circuits illustrated for $N = 5$ variables. Panel a shows the RealAmplitudes ansatz, which alternates single-qubit rotations with pairwise entangling layers. Panel b highlights the additional R_Z rotations in the EfficientSU2 ansatz that distinguish it from RealAmplitudes. Throughout this work, we refer to a single layer of these circuits, while deeper variants are obtained by repeating the same building block, as indicated by dashed lines.

the cost function C which is the full sample mean of the objective function for N_{shots} computational basis states $\boldsymbol{x}_k$ that we denote with X:

$$C(X) = \frac{1}{N_{\text{shots}}} \sum_{k=1}^{N_{\text{shots}}} O(\boldsymbol{x}_k) \tag{9}$$

In our study, we use the conditional value at risk (CVaR) as a cost function, C_{CVaR}, that was proposed to enhance the performance of VQAs for CO [35]. Assuming that the objective values $O(\boldsymbol{x}_k)$ are sorted in non-increasing order, the CVaR cost function only considers the fraction γ of large objective values:

$$C_{\text{CVaR}}(X) = \frac{1}{\lceil \gamma N_{\text{shots}} \rceil} \sum_{k=1}^{\lceil \gamma N_{\text{shots}} \rceil} O(\boldsymbol{x}_k). \tag{10}$$

The motivation for the CVaR cost function lies in the observation that, for CO, quantum states with large components with high objective values are favorable over states with possibly larger mean objective values but smaller components with high objective values. The case $\lceil \gamma N_{\text{shots}} \rceil = 1$ results in an optimization with respect to the maximal observed objective value, while the case $\gamma = 1$ retrieves the sample mean, which corresponds to the original VQE approach. The former leads to a discontinuous optimization landscape, making the choice of γ an important hyper-parameter.

To update the parameters, we choose the gradient-free Constrained Optimization BY Linear Approximation (COBYLA) algorithm [36]. We use the default hyper-parameters in its SciPy implementation[37].

We are aware that one does not expect the shallow VQE ansätze of our study to show competitive performance. In particular, we expect the resources required to sufficiently train the circuits of Fig. 1 to scale superpolynomially with problem size, as shallow circuits tend to lead to poor local minima. However, our goal is not to claim competitiveness of these specific setups, but to gain intuition for what the observed scaling behavior implies in practice for fixed problem sizes. Importantly, ansätze of this type also appear as components in more sophisticated algorithms, such as Filter-VQE [28] or Layer-VQE [38]. Our benchmarks can be straightforwardly applied to these methods and, more generally, to all variational algorithms that rely on repeated evaluations of an objective function.

Even when more performant ansätze or advanced cost functions are employed, the central question remains whether quantum algorithms offer an advantage over simple classical heuristics. In this sense, our benchmarks based on comparisons to random sampling and greedy local search remain informative, as they provide a baseline for assessing improvements in algorithmic design. Thus, while the specific VQE ansätze considered here may be simple, the benchmarking framework developed in this work can continue to play a valuable role in evaluating the progress of more advanced quantum optimization algorithms.

3.3 Quantum approximate optimization algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm inspired by the adiabatic principle [39]. In adiabatic quantum computing, the system is initialized in the ground state of a simple Hamiltonian H_M and evolved slowly under a time-dependent Hamiltonian $H(t) = f(t)H_C + g(t)H_M$, where H_C encodes the problem of interest. Here, $f(t)$ and $g(t)$ are smooth control functions that interpolate between the two Hamiltonians, with boundary conditions $f(0) = g(T) = 0$ and $f(T) = g(0) = 1$. If the interpolation is sufficiently slow and the spectral gap remains non-vanishing, the system remains in the ground state of $H(t)$ throughout the evolution, yielding the ground state of H_C at the end of the process.

On gate-based quantum devices, this continuous evolution can be approximated by a Trotter decomposition into p discrete layers,

$$U(\nu, \beta) = \prod_{k=1}^{p} e^{-i\beta_k H_M} e^{-i\nu_k H_C}, \tag{11}$$

with $2p$ variational parameters ν, β. The resulting circuit $U(\nu, \beta)$ serves as a variational ansatz and is trained in the same hybrid quantum–classical loop introduced in Sec. 3.1. In this sense, QAOA can be understood as a specific choice of circuit family within the general VQA framework, with the distinctive feature that it is tailored to the structure of combinatorial optimization problems.

A common choice for the mixer Hamiltonian is

$$H_M = \sum_{i \in V} X_i, \tag{12}$$

whose ground state is the uniform superposition of all computational basis states. A wide range of modifications of the original QAOA have been proposed, including alternative mixer Hamiltonians, counter-adiabatic terms, and different parameterizations [17]. In our study, we consider the multi-angle ansatz (ma-QAOA) [40], which introduces separate variational parameters for each term in both the cost and mixer Hamiltonians,

$$U(\nu, \beta) = \prod_{k=1}^{p} e^{-i \sum_n \beta_{k,n} H_{M,n}} e^{-i \sum_m \nu_{k,m} H_{C,m}}, \tag{13}$$

where $H_{C,m}$ and $H_{M,n}$ denote the individual terms in the respective Hamiltonians. Empirical studies suggest that ma-QAOA can perform as well as or better than standard QAOA while requiring shallower circuits [40].

Due to its connection with adiabatic quantum computing, QAOA is guaranteed to reach the optimal solution in the limit $p \to \infty$, though in the worst case this may require exponential depth $p \sim 2^{\mathcal{O}(N)}$. Nonetheless, QAOA remains an important candidate for quantum combinatorial optimization: it admits rigorous approximation guarantees at constant p for specific problems [41], and it provides a structured variational framework that can be directly compared to hardware-efficient ansätze.

3.4 Sampling with replacement

In general, VQAs require many iterations N_{iter} and measurements N_{shots} to converge on average. Since the total number of objective function evaluations is given by $N_{\text{evals}} = N_{\text{iter}} N_{\text{shots}}$, VQAs can be directly compared to other algorithms that also rely on repeated evaluations of the objective function. This benchmarking approach is biased in the sense that it favors the quantum algorithm: it assumes that drawing a random bitstring on a classical computer is as costly as running the variational quantum circuit and measuring a state in the computational basis from it.

As the simplest classical competitor, we consider uniformly sampling computational basis states with replacement, which ignores any problem structure. We use sampling with replacement rather than without replacement because the VQE likewise does not exploit information about which basis states have already been sampled. For a problem of size N, the probability of obtaining the optimal solution in this setting is $N_{\text{evals}}/2^N$. In our study, we numerically assess the approximation ratio achieved by this naive sampling strategy. This benchmark provides a lower bound and rules out the possibility that the VQAs are merely guessing solutions.

3.5 Greedy algorithm

An alternative to training a probabilistic model to generate computational basis states is to define a fixed set of rules to generate states. A possible choice is a greedy local search that updates states in a way that locally maximizes the objective function. The input to the algorithm is a computational basis state $\boldsymbol{x}$. Then, the algorithm computes the objective function for all states that differ from the initial state $\boldsymbol{x}$ by exactly one bit flip, and accepts the state with the largest improvement in objective value as the initial state for the next iteration. If there is no further improvement or if the maximal number of evaluations of the objective function N_{evals} is reached, the algorithm is terminated. The procedure is explained in detail in Algorithm 2. We note that the evaluation of the objective function can be implemented very efficiently for the greedy algorithm, because only the change of the objective function with respect to a single variable has to be computed. This means that the benchmark again has a bias in favor of the quantum algorithm.

Algorithm 2 Greedy Algorithm

Input: Initial binary vector $\boldsymbol{x}_{\text{init}} \in \{0,1\}^N$ sampled uniformly at random or from a quantum circuit, objective function O
Parameter: Maximum number of evaluations N_{evals}
Output: Binary vector $\boldsymbol{x}_{\text{max}} \in \{0,1\}^N$

$\boldsymbol{x}_{\text{max}} \leftarrow \boldsymbol{x}_{\text{init}}$
$O_{\text{max}} \leftarrow O(\boldsymbol{x}_{\text{max}})$
$\boldsymbol{x}_{\text{temp}} \leftarrow \boldsymbol{x}_{\text{max}}$
$n_{\text{evals}} \leftarrow 1$
while $n_{\text{evals}} < N_{\text{evals}}$ **do**
    $\boldsymbol{x}_{\text{update}} \leftarrow$ the single-bit flip of $\boldsymbol{x}_{\text{temp}}$ with the largest objective function value
    $n_{\text{evals}} \leftarrow n_{\text{evals}} +$ number of single bit flips
    $O_{\text{update}} \leftarrow O(\boldsymbol{x}_{\text{update}})$
    if $O_{\text{update}} > O_{\text{max}}$ **then**
        $\boldsymbol{x}_{\text{max}} \leftarrow \boldsymbol{x}_{\text{update}}$
        $O_{\text{max}} \leftarrow O_{\text{update}}$
        $\boldsymbol{x}_{\text{temp}} \leftarrow \boldsymbol{x}_{\text{update}}$
    else
        break
    end if
end while

For the first part of our numerical experiments, the greedy algorithm is initialized from uniformly random bitstrings. In addition, we are interested in studying the correlation of VQE and greedy performance by instance, i.e., defined jointly by the problem instance and the initial state.

To this end, we clarify what we mean by starting both algorithms from the same initial point. For the VQE, the initial point is given by the quantum circuit and its parameter initialization. The greedy algorithm, in contrast, starts from a single

computational basis state. We generate such a state by executing the VQE circuit once with the same initial parameters. The greedy update routine is then applied to this state until no further improvement is possible. If the procedure returns to a previously visited state, a new computational basis state is generated from the quantum circuit. The algorithm keeps track of all visited states and terminates the loop when a repetition occurs.

This procedure ensures that the VQE and the greedy algorithm are compared over the same set of initial points. This is particularly important because the performance of VQE depends strongly on initialization. Moreover, by analyzing correlations between the two algorithms on an instance-by-instance basis, we can assess whether favorable initial points for VQE are also favorable for greedy search.

4 Related work

In this section we summarize related work that we are aware of. Reference [42] compares the performance of a QAOA and a VQE to sampling with replacement for ferromagnetic and disordered Ising chains. They probe the regime to up to around 20 variables where they do not observe a practical advantage of the quantum algorithms, in agreement with our numerical findings in Sec. 5. Furthermore, they propose a parameter initialization strategy to enhance the performance of the QAOA.

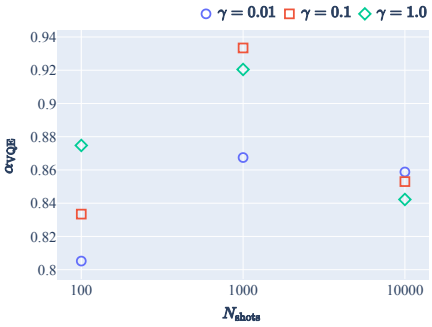
In an extended performance analysis [43] of the Filter-VQE algorithm [28] that also includes a comparison to sampling, the authors conclude that significant algorithmic developments are necessary to be competitive with state-of-the-art solvers, such as Gurobi [44].

The authors of Ref. [45] propose a quantum-enhanced greedy CO solver and compare it to its classical counterpart. The quantum version of the algorithm uses states obtained by the QAOA algorithm as starting points for the greedy procedure, while the classical version starts from a uniform distribution of all computational basis states.

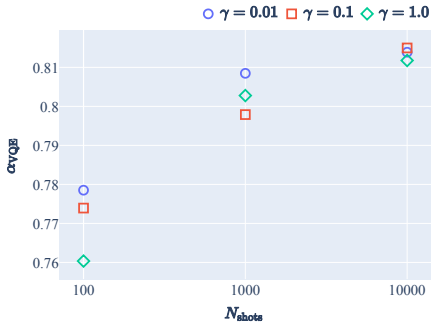
In Ref. [46], the authors propose a novel heuristic for quantum-inspired solvers that relies on encoding variables into correlations of Pauli-operators. Their encoding and the corresponding cost function lead to the best performance in experiment of a VQA for CO so far. They report average approximation ratios that are significantly better than a local search starting from randomly picked graph partitions. This is very similar to our comparison of the VQE to the greedy algorithm, which additionally uses the concept of starting both algorithms from the same initial point in the sense defined in Sec. 3.5.

5 Numerical experiments

We compare the performance of the algorithms described in Sec. 3 on the Max-Cut and random QUBO problems introduced in Sec. 2.1. We consider six different problem sizes ranging from 11 to 61 variables. For each problem type and size, we generate 25 instances and run each algorithm 10 times, using different initial points in the case of the VQAs. In total, this results in 9000 runs, which we use both for average-case convergence studies and for investigating correlations between algorithm performances.



(a) RealAmplitudes ansatz.



(b) ma-QAOA ansatz.

Fig. 2: Impact of hyperparameters on performance for the largest Max-Cut instances with $N = 61$ variables. Panel a shows results for the RealAmplitudes ansatz and panel b for the ma-QAOA ansatz. The plots illustrate the effect of varying the number of measurement shots N_{shots} and the CVaR parameter γ. The number of iterations N_{iter} is determined by the fixed evaluation budget according to $N_{\text{evals}} = N_{\text{iter}} N_{\text{shots}} = 10^6$.

To obtain the optimal solutions of all problem instances, we employ the Gurobi solver [44]. The problem sizes considered in this study are trivial for commercial solvers: Gurobi requires at most a few seconds to find the optimum and certify its optimality.

For the VQAs, we simulate the circuits in an ideal, noise-free setting using the matrix product state (MPS) method of Qiskit Aer [31]. The MPS method is well suited for shallow circuits even for larger qubit numbers. As explained in Sec. 3.2, important hyperparameters are the maximum number of iterations N_{iter} and the number of measurements per iteration N_{shots}. We investigate the algorithms in the regime $N_{\text{evals}} = N_{\text{iter}} N_{\text{shots}} = 10^6$, corresponding to a budget of one million objective-function evaluations. Exploring larger budgets does not significantly improve performance.

By testing different combinations of N_{iter} and N_{shots} on the largest Max-Cut instances of size 61, we found $N_{\text{shots}} = 10^3$ and $N_{\text{iter}} = 10^3$ to perform best on average for the hardware-efficient circuits, and $N_{\text{shots}} = 10^4$ and $N_{\text{iter}} = 10^2$ for the ma-QAOA algorithm. The impact of these hyperparameters, as well as the effect of the CVaR parameter γ, is shown in Fig. 2. For convenience, we summarize all hyperparameter choices in Tab. 1.

For benchmarking sampling with replacement, we use the NumPy random number generator [47] to generate computational basis states. We implemented the greedy algorithm with native Python data structures and NumPy.

Algorithm	**VQE**	**ma-QAOA**
Backend	Matrix product state simulator of Qiskit Aer without noise [31].	
Circuit M	Single layer of the RealAmplitudes or EfficientSU2 ansatz (Fig. 1)	Single layer of the multi-angle QAOA ansatz (Sec. 3.3)
Cost function C	C_{CVaR} cost function (Eq. 10) with $\gamma = 0.1$.	
Max iterations N_{iter}	1000	100
Measurements per iteration N_{shots}	1000	10000
Optimizer OA	Constrained Optimization BY Linear Approximation (COBYLA) with default hyperparameters of its SciPy implementation [37].	

Table 1: Hyperparameters used for the VQE and ma-QAOA algorithms. The total evaluation budget is fixed at $N_{\mathrm{evals}} = N_{\mathrm{iter}} N_{\mathrm{shots}} = 10^6$.

5.1 Performance of the RealAmplitudes ansatz in comparison to classical baselines

For each problem type and size, and for each of the 25 instances, we run the quantum algorithms (VQAs) with ten different initial points. For every such run, we compute the difference in approximation ratio relative to the classical baselines. The comparisons to sampling and greedy local search are based on ten independent runs each. This procedure yields 250 differences per problem size for each comparison, from which we compute the mean and the standard error of the mean (SEM).

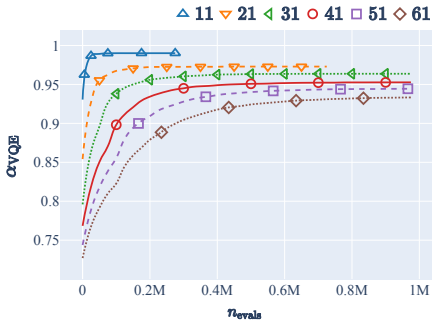
To study convergence, we report results as a function of the number of calls to the objective function, rather than in terms of VQA iterations. For fairness, we compare the VQA after n_{evals} evaluations with the best approximation ratio achieved by sampling or greedy local search within the same budget of n_{evals} evaluations.

We begin by analyzing the performance of the VQE using the RealAmplitudes ansatz and the hyperparameters summarized in Tab. 1. Figure 3a shows the approximation ratio achieved as a function of the number of objective function evaluations, and Fig. 3b reports the corresponding success probability.

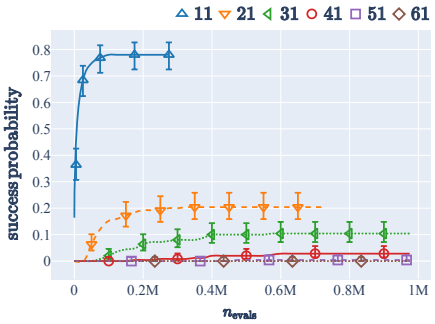
For problem size 11, we observe approximation ratios close to one and success probabilities well above 75%. However, when comparing against classical baselines, these metrics prove misleading. Figure 4 contrasts the VQE with sampling and greedy local search.

In particular, for problem size 11, the high approximation ratio seen in Fig. 3(a) does not reflect a genuine advantage, since VQE performs no better than random sampling. A similar behavior appears at size 21, where VQE performs worse than sampling except at very early stages. As shown in Fig. 4b, after roughly 75,000 evaluations VQE outperforms sampling on about 60% of instances, which corresponds to an average improvement in approximation ratio of only $\sim$0.005 (Fig. 4a). At around 150,000 evaluations, the two methods perform equally well on average, and for larger budgets, sampling consistently surpasses VQE.

For larger problem sizes, a different picture emerges. Once the number of computational basis states exceeds 2^{30}, the performance of random sampling degrades rapidly.



(a) Average approximation ratio achieved by VQE for the Max-Cut problem on 3-regular graphs of varying sizes. Error bars (SEM) are smaller than the markers and therefore not visible.



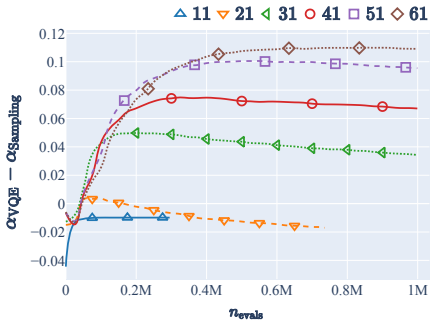
(b) Probability of obtaining the exact solution, shown with mean and 95% confidence interval. The success probability decays rapidly with system size, making it informative only for small problems.

Fig. 3: Standard performance metrics for VQE applied to combinatorial optimization. Results are evaluated after every VQE iteration (lines), with markers added to indicate selected reference points and to display uncertainties clearly. This style of showing continuous results with lines and selected markers is also used in the following plots.

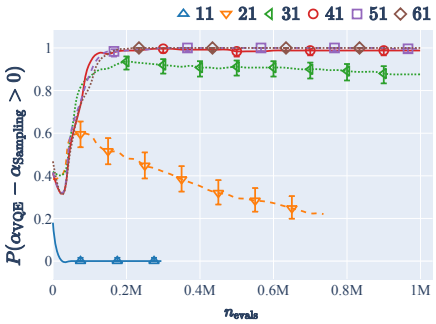
For problem size 31, the VQE outperforms sampling on roughly 95% of instances, corresponding to an average improvement in approximation ratio of about 0.05. For sizes above 40, VQE consistently performs better than sampling after about 10^5 objective function evaluations.

We then turn to the greedy local search described in Sec. 3.5. As a fair baseline, we compare the VQE to the stronger of the two classical heuristics by defining $\alpha_{\text{Classic}} = \max(\alpha_{\text{Sampling}}, \alpha_{\text{Greedy}})$. In practice, sampling only outperforms greedy for problem size 11 and during the early stages at size 21; otherwise, greedy dominates. For larger problem sizes, a characteristic behavior emerges: the greedy algorithm quickly converges to local minima and initially achieves better approximation ratios than VQE. Over time, the VQE slowly catches up and eventually converges to better local minima on average. The evaluation count at which VQE matches greedy increases with problem size—from roughly 100,000 evaluations at size 31 to about 600,000 evaluations at size 61. The maximal advantage of VQE over greedy, however, decreases with problem size. For size 31, VQE achieves a higher approximation ratio than the greedy algorithm on up to 75% of instances, corresponding to an average improvement of about 0.025. By size 61, this drops to around 60% of instances and an average improvement of only about 0.005. These trends are illustrated in Fig. 4c and Fig. 4d. This suggests that although VQE can surpass greedy search, it does so only after substantially more evaluations, and its relative advantage diminishes as problem size increases.

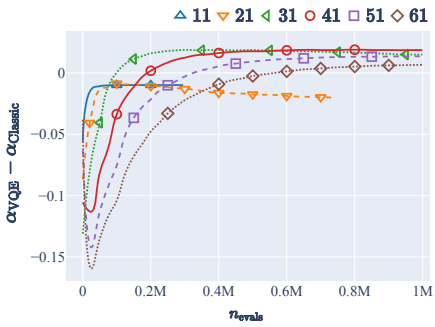
We next analyze the performance of VQAs on random QUBO instances, introduced in Sec. 5.3. The results are shown in Fig. 5. Compared to Max-Cut, VQAs



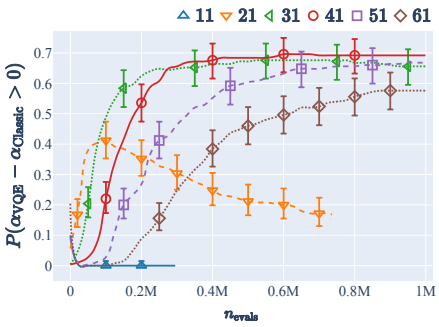
(a) Difference in approximation ratio between VQE and sampling.



(b) Probability that VQE outperforms sampling.



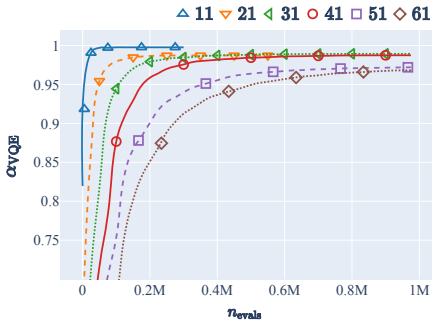
(c) Difference in approximation ratio between VQE and the best of sampling and greedy local search.



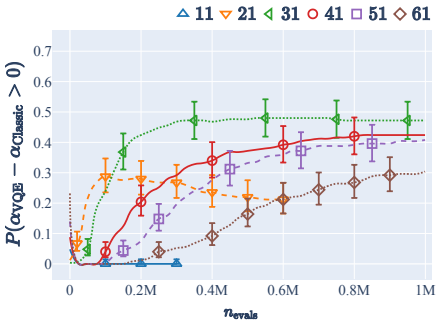
(d) Probability that VQE outperforms the best of sampling and greedy local search.

Fig. 4: Performance comparison of VQE against classical baselines on Max-Cut instances. Panels a and b show the difference in approximation ratio and success probability relative to sampling with replacement. Panels c and d present the same metrics relative to the stronger baseline given by the better of sampling and greedy local search. Different markers indicate problem sizes, and lines connect data points as a guide for the eye.

achieve markedly higher approximation ratios: values remain close to 1 even for problem sizes up to $N = 41$, whereas for Max-Cut the approximation ratio already degrades significantly at much smaller sizes. However, the apparent improvement is misleading. Random QUBO instances are also easier to approximate with classical heuristics, and the greedy local search performs strongly. As a result, the probability that VQE achieves a better approximation ratio than the greedy baseline never exceeds 50% across all sizes studied. Thus, although VQAs appear more successful on QUBO than



(a) Average approximation ratio achieved by VQE on random QUBO instances, reaching values close to 1 up to size 41.



(b) Fraction of instances where VQE outperforms the best of sampling and greedy. This fraction never exceeds 50%.

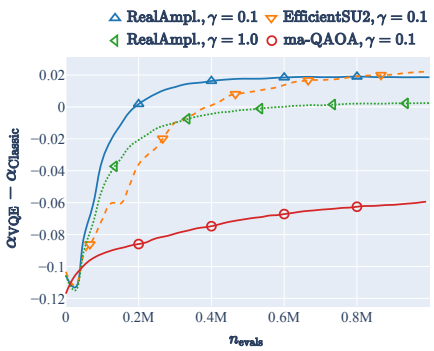
Fig. 5: Performance of VQE on random QUBO instances. While the approximation ratios are higher than for Max-Cut, the problem is also easier for simple classical heuristics, leaving no consistent advantage for VQE.

on Max-Cut, they do not provide a genuine advantage over simple classical methods. This underscores the importance of selecting problems with established hardness of approximation, such as Max-Cut, when assessing the broader significance of VQA performance.

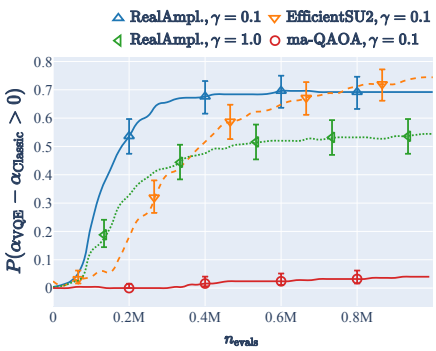
5.2 Scaling behavior with system size and comparison of different ansätze

After this detailed analysis of the VQE with the RealAmplitudes ansatz, we return to the Max-Cut problem and analyze the impact of variations in circuit structure and cost function, as well as a comparison to ma-QAOA, an alternative paradigm for constructing problem-specific circuits introduced in Sec. 3.3. Specifically, we compare the performance of (i) the RealAmplitudes ansatz with the CVaR coefficient $\gamma = 1$, corresponding to the full expectation value, (ii) the same circuit with $\gamma = 0.1$, (iii) the EfficientSU2 circuit with the CVaR cost function, and (iv) a single layer of the ma-QAOA ansatz. The convergence of the approximation ratio for these four algorithms, both in isolation and relative to the classical baselines, is shown in Fig. 6 for a fixed problem size of 41. This comparison provides a representative case study illustrating how the choice of ansatz and cost function influences performance.

We find that the choice of CVaR parameter has a noticeable effect: using $\gamma = 0.1$ yields consistently higher approximation ratios than $\gamma = 1.0$ (the full expectation value), while the convergence rates remain similar. The EfficientSU2 ansatz converges more slowly but eventually reaches a slightly higher approximation ratio than the RealAmplitudes circuit. These differences are more clearly visible in comparison to the classical baselines than in the raw approximation ratios, underscoring the usefulness of our benchmark. Overall, the two hardware-efficient circuits behave similarly,



(a) Average difference in approximation ratio between the four VQA variants (RealAmplitudes with $\gamma = 1.0$ or $\gamma = 0.1$, EfficientSU2, and single-layer ma-QAOA) and the classical baseline.



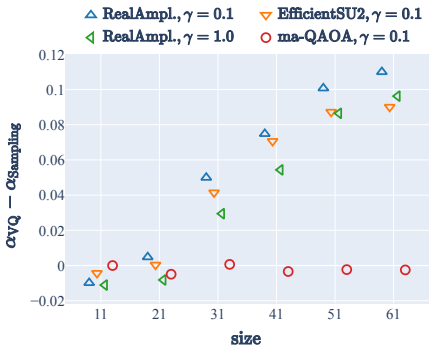
(b) Fraction of instances where each VQA achieves a higher approximation ratio than the classical baseline. The best hardware-efficient circuit outperforms the baseline on about 70% of instances, whereas ma-QAOA remains close to 0%.

Fig. 6: Comparison of different VQA variants for Max-Cut at fixed problem size $N = 41$. The choice of ansatz and cost function affects performance, with hardware-efficient circuits consistently outperforming shallow problem-inspired ma-QAOA.

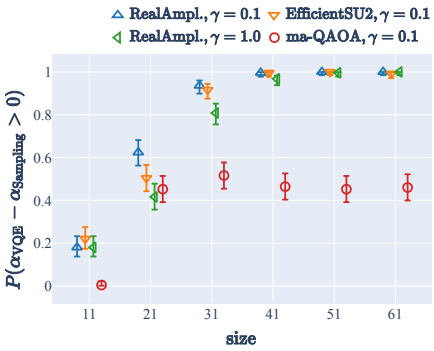
whereas the single-layer ma-QAOA—which already exceeds the depth of the hardware-efficient circuits—performs substantially worse, with approximation ratios below 0.9 compared to about 0.95 for the hardware-efficient circuits. This gap is also evident in the probability of outperforming the stronger classical baseline: the best hardware-efficient ansatz succeeds on roughly 70% of instances, while ma-QAOA stays close to 0%. It is worth noting, however, that deeper ma-QAOA circuits have been shown to achieve significantly improved performance [40], suggesting that additional depth may be required before problem-inspired ansätze outperform shallow hardware-efficient circuits.

After analyzing the convergence behavior of the different algorithms, we now examine their scaling with problem size in comparison to classical baselines. Fig. 7a reports the average difference in approximation ratio between the quantum algorithms and uniform random sampling, while Fig. 7b shows the corresponding probability of achieving a better approximation ratio. The same metrics are shown relative to the stronger classical baseline—the better of sampling and greedy local search—in Fig. 7c and Fig. 7d.

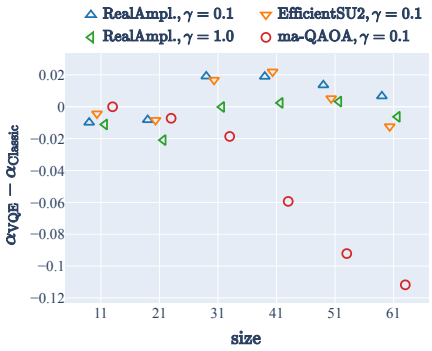
The results reveal a distinct scaling behavior with respect to sampling. For hardware-efficient circuits, the fraction of instances converges to 1 for both choices of the CVaR coefficient γ, indicating a clear advantage over random guessing for Max-Cut instances with more than 30 vertices. Below this regime, the performance is largely indistinguishable from random sampling, with the fraction of favorable instances barely exceeding 0.5. By contrast, for the ma-QAOA ansatz the metric converges to



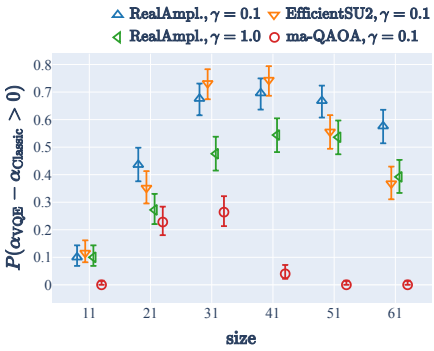
(a) Difference in approximation ratio relative to random sampling.



(b) Probability of outperforming random sampling.



(c) Difference in approximation ratio relative to the stronger classical baseline.

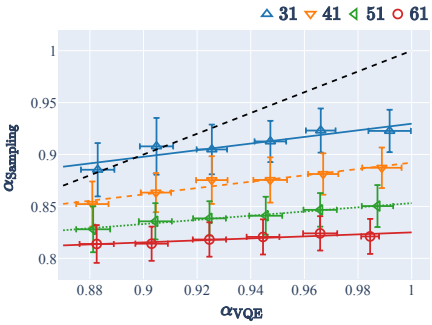


(d) Probability of outperforming the stronger classical baseline.

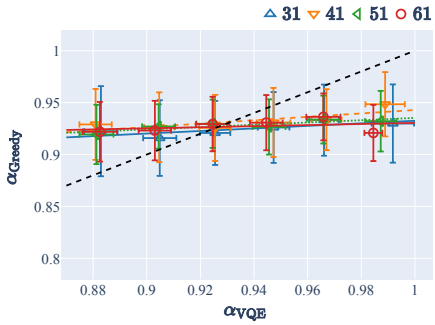
Fig. 7: Scaling behavior of VQAs compared to classical baselines for Max-Cut. Panels (a)–(b) show comparisons to uniform random sampling, while panels (c)–(d) use the stronger baseline defined as $\alpha_{\text{Classic}} = \max(\alpha_{\text{Sampling}}, \alpha_{\text{Greedy}})$.

approximately 0.5, suggesting performance that is statistically indistinguishable from random guessing.

The performance relative to the greedy baseline exhibits a similarly distinct trend. First, the advantage over greedy local search increases with problem size. For hardware-efficient circuits with a suitable CVaR coefficient of $\gamma = 0.1$, the advantage peaks around 40 variables, where the VQAs outperform greedy local search on roughly 70% of the instances. Beyond this point, however, the performance decreases for larger problems. The ma-QAOA ansatz also shows a peak improvement at intermediate problem sizes, but the fraction never exceeds 0.3 and drops to 0 for the largest instances.



(a) Correlation between VQE and sampling. The performance of sampling depends strongly on problem size.



(b) Lack of correlation between VQE and greedy. The average performance of the greedy algorithm remains constant across problem sizes in this regime.

Fig. 8: Binned statistics of the approximation ratios of sampling and the greedy algorithm as functions of the VQE approximation ratio. Error bars indicate the standard deviation within each bin. The dashed black lines correspond to perfect correlation with slope one.

5.3 Instance-wise performance correlations

As a final step of our analysis, we move beyond average-case performance and investigate correlations between the algorithms on an instance-by-instance basis. In Fig. 8, we show the approximation ratios achieved by sampling and by the greedy algorithm as functions of the approximation ratio obtained with the VQE using the RealAmplitudes ansatz and the hyperparameters summarized in Tab. 1. We employ binned statistics as described in Sec. 2.2 and restrict attention to the regime where the VQE outperforms the worst-case guarantee of the classical GW algorithm. To avoid misleading artifacts, we exclude problem sizes 11 and 21, for which the VQE performs strictly worse than both sampling and greedy search. For comparisons with the greedy algorithm, both methods are initialized from the same starting points as explained in Sec. 3.5.

The approximation ratios of sampling and the VQE exhibit a mild correlation that depends on the problem size. This is expected, since sampling is an integral component of the VQE algorithm. As sampling does not rely on initial points, its varying performance directly reflects the instance-dependent sampling hardness of the randomly generated Max-Cut problems. Intuitively, sampling performs better for instances with many near-optimal solutions in the objective landscape. However, its performance decreases rapidly with problem size across the full range of VQE approximation ratios, and the observed correlation weakens correspondingly. This suggests that the variance in instance hardness with respect to sampling diminishes as problem size grows. By contrast, the variation in VQE performance is primarily attributable to differences in initialization.

A completely different picture emerges when comparing VQE with the greedy algorithm. The standard deviation of the greedy approximation ratios within each bin is comparable to the entire range of ratios considered, indicating an absence of meaningful correlation between the two methods. This suggests that VQE and the greedy algorithm exploit fundamentally different mechanisms. Moreover, the average approximation ratio achieved by the greedy algorithm remains essentially independent of the problem size for the instances shown in Fig. 8.

6 Discussion and outlook

In this study, we have conducted extensive numerical experiments to establish intuitive benchmarks for VQAs beyond proof-of-principle demonstrations. For small instances of the widely studied Max-Cut problem on 3-regular graphs, we observed that randomly sampling solutions already yields high approximation ratios. This highlights the importance of performing resource-aware comparisons between VQAs and sampling in this regime. For the specific VQAs considered in our study, we found that they outperform random sampling only for problem sizes larger than approximately 30 variables. This implies that probing larger problem instances is essential for drawing meaningful conclusions about algorithmic performance. Advanced simulation methods, such as tensor networks, make it possible to simulate shallow quantum circuits with many qubits at reasonable computational cost, and thus represent a valuable tool for enabling such studies.

Another intuitive benchmark is provided by comparisons to fixed rule-based heuristics. In our study, such rules define a greedy algorithm that achieves a nearly constant approximation ratio for larger problems. This behavior effectively bounds the regime in which VQAs may offer an advantage: the performance of random sampling defines a lower baseline, while the performance of the greedy algorithm provides an upper reference. By initializing the greedy algorithm from the same starting points as the VQE, as described in Sec. 3.5, we demonstrated that good initializations for the VQE are not necessarily advantageous for the greedy algorithm. For alternative approaches to finding suitable initial points for VQAs, we refer the reader to [48].

Ultimately, the practical utility of an algorithm is determined by its economic efficiency, defined by the quality of its solutions relative to the time and financial resources required. At present, benchmarking VQAs in this way, as is common for state-of-the-art classical solvers, is not yet feasible. This is partly due to limited access to sufficiently powerful hardware, but more importantly because algorithmic development is still in its early stages and current VQAs are not competitive. For this reason, our benchmarks focus on performance comparisons under a fixed budget of objective function evaluations. Such comparisons provide a hardware-independent measure that highlights strengths and weaknesses of different approaches and can help guide algorithm design in the right direction. We emphasize that the benchmark instances typically used in the quantum computing literature are trivial for commercial classical solvers. Nevertheless, VQAs remain an early-stage approach within the rapidly evolving field of quantum computation, where progress is ongoing in both algorithmic design and hardware capabilities. We hope that our work contributes to the development of

meaningful benchmarks for quantum algorithms that extend beyond proof-of-principle demonstrations.

Importantly, our benchmarks remain relevant even for future quantum algorithms that could achieve a computational advantage, since a broad class of heuristic quantum optimization methods ultimately rely on repeated measurements and evaluation of the objective function. In addition, while in this work we compared against only two simple classical baselines—random sampling and a greedy local search—the benchmarking framework can be straightforwardly extended to include more elaborate classical heuristics. This flexibility ensures that evaluation-based benchmarks can continue to provide fair and informative points of reference as both quantum and classical algorithms advance.

An important direction for future work is the incorporation of noise into benchmarking studies. The impact of noise on the training of VQAs has been the focus of significant recent research. For general VQA settings, it has been shown that noise can exacerbate barren plateaus and thereby severely limit trainability [49]. In the specific case of QAOA, noise has been found to counteract the benefits of increasing circuit depth, preventing performance improvements that would otherwise be expected in the noiseless setting [50]. At the same time, strategies such as error mitigation [51], modified cost functions for quantum approximate optimization [52], and circuit designs tailored to hardware constraints [53] are being actively developed to counteract these effects. Our benchmarks are defined in terms of objective function evaluations rather than physical runtime, which makes them naturally applicable in noisy scenarios. Runtime studies are valuable, but they depend strongly on hardware-specific factors such as gate fidelities, qubit connectivity, and the overhead introduced by error mitigation. Since these aspects vary widely across platforms and continue to improve rapidly, runtime comparisons between quantum and classical solvers are inherently hardware-dependent. By contrast, counting objective function evaluations provides a hardware-independent measure of algorithmic efficiency that isolates the intrinsic strengths and weaknesses of different approaches. This perspective can already guide algorithm design today and, once reliable quantum devices become available, can be complemented by runtime studies to provide a complete picture of performance.

Declarations

Availability of data and materials

The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by: (i) the Einstein Research Unit, Perspectives of a Quantum Digital Transformation: Near-term Quantum Computational Devices and

Quantum Processors; (ii) the European Union's Horizon Europe research and innovation programme under the ERA Chair scheme (Grant Agreement No. 101087126) and the European Research Council (ERC) (Grant Agreement No. 787331); (iii) the Helmholtz Association, Innopool Project Variational Quantum Computer Simulations (VQCS); and (iv) the Ministry of Science, Research and Culture of the State of Brandenburg through the Centre for Quantum Technologies and Applications (CQTA).

Authors' contributions

TS prepared the manuscript, conducted the numerical experiments, and carried out the statistical analysis. All authors contributed ideas and guidance, reviewed the manuscript, and approved the final version.

Acknowledgements

Not applicable.

References

[1] Eskandarpour, M., Dejax, P., Miemczyk, J., Péton, O.: Sustainable supply chain network design: An optimization-oriented review. Omega **54**, 11–32 (2015) https://doi.org/10.1016/j.omega.2015.01.006

[2] Sbihi, A., Eglese, R.: Combinatorial optimization and green logistics. Annals of Operations Research **175**, 254–5330 (2010) https://doi.org/10.1007/s10479-009-0651-z

[3] Barahona, F., Grötschel, M., Jünger, M., Reinelt, G.: An Application of Combinatorial Optimization to Statistical Physics and Circuit Layout Design. Operations Research **36**(3), 493–513 (1988)

[4] Saito, M., Calafiura, P., Gray, H., Lavrijsen, W., Linder, L., Okumura, Y., Sawada, R., Smith, A., Tanaka, J., Terashi, K.: Quantum annealing algorithms for track pattern recognition. EPJ Web of Conferences **245**, 10006 (2020) https://doi.org/10.1051/epjconf/202024510006

[5] Gray, H.M.: Quantum pattern recognition algorithms for charged particle tracking. Philosophical Transactions of the Royal Society A: Mathematical, Physical

and Engineering Sciences **380**(2216) (2021) https://doi.org/10.1098/rsta.2021.0103

[6] Zlokapa, A., Anand, A., Vlimant, J.-R., Duarte, J.M., Job, J., Lidar, D., Spiropulu, M.: Charged particle tracking with quantum annealing optimization. Quantum Machine Intelligence **3**(2) (2021) https://doi.org/10.1007/s42484-021-00054-w

[7] Funcke, L., Hartung, T., Heinemann, B., Jansen, K., Kropf, A., Kühn, S., Meloni, F., Spataro, D., Tüysüz, C., Yap, Y.C.: Studying quantum algorithms for particle track reconstruction in the LUXE experiment. Journal of Physics: Conference Series **2438**(1), 012127 (2023) https://doi.org/10.1088/1742-6596/2438/1/012127

[8] Schwägerl, T., Issever, C., Jansen, K., Khoo, T.J., Kühn, S., Tüysüz, C., Weber, H.: Particle track reconstruction with noisy intermediate-scale quantum computers (2023). https://arxiv.org/abs/2303.13249

[9] Crippa, A., Funcke, L., Hartung, T., Heinemann, B., Jansen, K., Kropf, A., Kühn, S., Meloni, F., Spataro, D., Tüysüz, C., Yap, Y.C.: Quantum algorithms for charged particle track reconstruction in the luxe experiment. Computing and Software for Big Science **7**(1), 14 (2023) https://doi.org/10.1007/s41781-023-00109-6

[10] Okawa, H.: Charged particle reconstruction for future high energy colliders with quantum approximate optimization algorithm. In: Cruz, C., Zhang, Y., Gao, W. (eds.) Intelligent Computers, Algorithms, and Applications, pp. 272–283. Springer, Singapore (2024)

[11] Karp, R.M.: In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) Reducibility among Combinatorial Problems, pp. 85–103. Springer, Boston, MA (1972). https://doi.org/10.1007/978-1-4684-2001-2_9

[12] Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700

[13] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96, pp. 212–219. Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237814.237866

[14] Preskill, J.: Quantum Computing in the NISQ era and beyond. Quantum **2**, 79 (2018) https://doi.org/10.22331/q-2018-08-06-79 . arXiv:1801.00862

[15] Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm (2014). https://arxiv.org/abs/1411.4028

[16] Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'Brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature Communications **5**(1), 4213 (2014) https://doi.org/10.1038/ncomms5213

[17] Blekos, K., Brand, D., Ceschini, A., Chou, C.-H., Li, R.-H., Pandya, K., Summer, A.: A Review on Quantum Approximate Optimization Algorithm and its Variants. Physics Reports **1068**, 1–66 (2024) https://doi.org/10.1016/j.physrep.2024.03.002

[18] Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G.H., Tennyson, J.: The variational quantum eigensolver: A review of methods and best practices. Physics Reports **986**, 1–128 (2022) https://doi.org/10.1016/j.physrep.2022.08.003

[19] Stollenwerk, T., Michaud, V., Lobe, E., Picard, M., Basermann, A., Botter, T.: Agile earth observation satellite scheduling with a quantum annealer. IEEE Transactions on Aerospace and Electronic Systems **57**(5), 3520–3528 (2021) https://doi.org/10.1109/TAES.2021.3088490

[20] Chai, Y., Funcke, L., Hartung, T., Jansen, K., Kühn, S., Stornati, P., Stollenwerk, T.: Towards Finding an Optimal Flight Gate Assignment on a Digital Quantum Computer. Physical Review Applied **20**(6), 064025 (2023) https://doi.org/10.1103/PhysRevApplied.20.064025

[21] Chai, Y., Epifanovsky, E., Jansen, K., Kaushik, A., Kühn, S.: Simulating the flight gate assignment problem on a trapped ion quantum computer (2023). https://arxiv.org/abs/2309.09686

[22] Hancock, J., Craven, M.J., McNeile, C., Vadacchino, D.: Investigating techniques to optimise the layout of turbines in a windfarm using a quantum computer (2024). https://arxiv.org/abs/2312.13123

[23] Amaro, D., Rosenkranz, M., Fitzpatrick, N., Hirano, K., Fiorentini, M.: A case study of variational quantum algorithms for a job shop scheduling problem. EPJ Quantum Technology **9**(1), 1–20 (2022) https://doi.org/10.1140/epjqt/s40507-022-00123-4

[24] Nannicini, G.: Performance of hybrid quantum/classical variational heuristics for combinatorial optimization. Physical Review E **99**(1), 013304 (2019) https://doi.org/10.1103/PhysRevE.99.013304 . arXiv:1805.12037

[25] Arrasmith, A., Holmes, Z., Cerezo, M., Coles, P.J.: Equivalence of quantum barren plateaus to cost concentration and narrow gorges. Quantum Science and Technology **7**(4), 045015 (2022) https://doi.org/10.1088/2058-9565/ac7d06 . arXiv:2104.05868

[26] Miao, Q., Barthel, T.: Equivalence of cost concentration and gradient vanishing for quantum circuits: an elementary proof in the riemannian formulation. Quantum Science and Technology **9**(4), 045039 (2024) https://doi.org/10.1088/2058-9565/ad6fca

[27] Anschuetz, E.R., Kiani, B.T.: Quantum variational algorithms are swamped with traps. Nature Communications **13**(1), 7760 (2022) https://doi.org/10.1038/s41467-022-35364-5

[28] Amaro, D., Modica, C., Rosenkranz, M., Fiorentini, M., Benedetti, M., Lubasch, M.: Filtering variational quantum algorithms for combinatorial optimization. Quantum Science and Technology **7**(1), 015021 (2022) https://doi.org/10.1088/2058-9565/ac3e54 . arXiv:2106.10055

[29] Håstad, J.: Some optimal inapproximability results. Journal of the ACM **48**(4), 798–859 (2001) https://doi.org/10.1145/502090.502098

[30] Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM **42**(6), 1115–1145 (1995) https://doi.org/10.1145/227683.227684

[31] Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C.J., Lishman, J., Gacon, J., Martiel, S., Nation, P.D., Bishop, L.S., Cross, A.W., Johnson, B.R., Gambetta, J.M.: Quantum computing with Qiskit (2024). https://doi.org/10.48550/arXiv.2405.08810

[32] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. Journal of Combinatorial Optimization **28**(1), 58–81 (2014) https://doi.org/10.1007/s10878-014-9734-0

[33] Wilson, E.B.: Probable Inference, the Law of Succession, and Statistical Inference. Journal of the American Statistical Association **22**(158), 209–212 (1927) https://doi.org/10.2307/2276774

[34] Seabold, S., Perktold, J.: statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference (2010)

[35] Barkoutsos, P.K., Nannicini, G., Robert, A., Tavernelli, I., Woerner, S.: Improving Variational Quantum Optimization using CVaR. Quantum **4**, 256 (2020) https://doi.org/10.22331/q-2020-04-20-256

[36] Powell, M.J.D.: Direct search algorithms for optimization calculations. Acta Numerica **7**, 287–336 (1998)

[37] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt,

S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020) https://doi.org/10.1038/s41592-019-0686-2

[38] Liu, X., Angone, A., Shaydulin, R., Safro, I., Alexeev, Y., Cincio, L.: Layer VQE: A Variational Approach for Combinatorial Optimization on Noisy Quantum Computers. IEEE Transactions on Quantum Engineering **3**, 1–20 (2022) https://doi.org/10.1109/TQE.2021.3140190

[39] Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum Computation by Adiabatic Evolution (2000). https://arxiv.org/abs/quant-ph/0001106

[40] Herrman, R., Lotshaw, P.C., Ostrowski, J., Humble, T.S., Siopsis, G.: Multi-angle quantum approximate optimization algorithm. Scientific Reports **12**(1), 6781 (2022) https://doi.org/10.1038/s41598-022-10555-8

[41] Basso, J., Farhi, E., Marwaha, K., Villalonga, B., Zhou, L.: The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model. In: Le Gall, F., Morimae, T. (eds.) 17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 232, pp. 7–1721. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://doi.org/10.4230/LIPIcs.TQC.2022.7 . https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2022.7

[42] Scriva, G., Astrakhantsev, N., Pilati, S., Mazzola, G.: Challenges of variational quantum optimization with measurement shot noise. Physical Review A **109**(3) (2024) https://doi.org/10.1103/physreva.109.032408

[43] Marin-Sanchez, G., Amaro, D.: Performance analysis of a filtering variational quantum algorithm (2024). https://arxiv.org/abs/2404.08933

[44] Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2023). https://www.gurobi.com

[45] Dupont, M., Evert, B., Hodson, M.J., Sundar, B., Jeffrey, S., Yamaguchi, Y., Feng, D., Maciejewski, F.B., Hadfield, S., Alam, M.S., Wang, Z., Grabbe, S., Lott, P.A., Rieffel, E.G., Venturelli, D., Reagor, M.J.: Quantum-Enhanced Greedy Combinatorial Optimization Solver. Science Advances **9**(45), 0487 (2023) https://doi.org/10.1126/sciadv.adi0487

[46] Sciorilli, M., Borges, L., Patti, T.L., García-Martín, D., Camilo, G., Anandkumar, A., Aolita, L.: Towards large-scale quantum optimization solvers with few qubits

(2024). https://arxiv.org/abs/2401.09421

[47] Harris, C.R., Millman, K.J., Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M.H., Brett, M., Haldane, A., Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020) https://doi.org/10.1038/s41586-020-2649-2

[48] Chai, Y., Jansen, K., Kühn, S., Schwägerl, T., Stollenwerk, T.: Structure-inspired ansatz and warm start of variational quantum algorithms for quadratic unconstrained binary optimization problems. arXiv:2407.02569, (2024) https://doi.org/10.48550/ARXIV.2407.02569

[49] Wang, S., Fontana, E., Cerezo, M., Sharma, K., Sone, A., Cincio, L., Coles, P.J.: Noise-induced barren plateaus in variational quantum algorithms. Nature Communications **12**(1), 6961 (2021) https://doi.org/10.1038/s41467-021-27045-6

[50] Pellow-Jarman, A., McFarthing, S., Sinayskiy, I., Park, D.K., Pillay, A., Petruccione, F.: The effect of classical optimizers and ansatz depth on qaoa performance in noisy devices. Scientific Reports **14**(1), 16011 (2024) https://doi.org/10.1038/s41598-024-66625-6

[51] Wang, S., Czarnik, P., Arrasmith, A., Cerezo, M., Cincio, L., Coles, P.J.: Can error mitigation improve trainability of noisy variational quantum algorithms? Quantum **8**, 1287 (2024) https://doi.org/10.22331/q-2024-03-14-1287

[52] Weidinger, A., Mbeng, G.B., Lechner, W.: Error mitigation for quantum approximate optimization. Phys. Rev. A **108**, 032408 (2023) https://doi.org/10.1103/PhysRevA.108.032408

[53] Huang, Y., Li, Q., Hou, X., Wu, R., Yung, M.-H., Bayat, A., Wang, X.: Robust resource-efficient quantum variational ansatz through an evolutionary algorithm. Phys. Rev. A **105**, 052414 (2022) https://doi.org/10.1103/PhysRevA.105.052414