Unsupervised strategies for identifying optimal parameters in Quantum Approximate Optimization Algorithm

Charles Moussa (c.moussa@liacs.leidenuniv.nl)
Leiden University
Hao Wang
Leiden University
Thomas Bäck
Leiden University
Vedran Dunjko (v.dunjko@liacs.leidenuniv.nl)
Leiden University

Research Article

Keywords: Quantum Computing, Combinatorial Optimization, Quantum Approximate Optimization Algorithm, Clustering

Posted Date: March 3rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1407056/v1

RESEARCH

Unsupervised strategies for identifying optimal parameters in Quantum Approximate Optimization Algorithm

Charles Moussa*, Hao Wang, Thomas Bäck and Vedran Dunjko*

Abstract

As combinatorial optimization is one of the main quantum computing applications, many methods based on parameterized quantum circuits are being developed. In general, a set of parameters are being tweaked to optimize a cost function out of the quantum circuit output. One of these algorithms, the Quantum Approximate Optimization Algorithm stands out as a promising approach to tackle combinatorial problems, due to many interesting properties. However finding the appropriate parameters is a difficult task. Although QAOA exhibits concentration properties, they can depend on instances characteristics which may not be easy to identify, but may nonetheless offer useful information to find good parameters. In this work, we study unsupervised Machine Learning approaches for setting these parameters without optimization. We perform clustering with the angle values but also instances encodings (using instance features or the output of a variational graph autoencoder), and compare different approaches. These angle-fiding strategies can be used to reduce calls to quantum circuits when leveraging QAOA as a subroutine. We showcase them within Recursive-QAOA up to depth 3 where the number of QAOA parameters used per iteration is limited to 3, achieving a median approximation ratio of 0.94 for MaxCut over 200 Erdős-Rényi graphs. We obtain similar performances to the case where we extensively optimize the angles, hence saving numerous circuit calls.

Keywords: Quantum Computing; Combinatorial Optimization; Quantum Approximate Optimization Algorithm; Clustering

1 Introduction

Noisy Intermediate-Scale Quantum (NISQ) era hardware [1] faces many limiting challenges preventing fault-tolerant quantum algorithm execution (e.g., the number of qubits, decoherence, etc.). Hence near-term hybrid quantum-classical algorithms were designed as an alternative for applications such as quantum chemistry problems [2], quantum machine learning [3] and combinatorial optimization [4]. One or many so-called variational quantum circuits (or parameterized quantum circuits), are submitted to quantum devices with a fixed circuit architecture while the parameters of individual gates are adjusted in a classical loop to achieve a computational objective.

With a user-specified depth p, the Quantum Approximate Optimization Algorithm (QAOA) [4] consists of a quantum circuit involving $2p$ real parameters (or angles). In the limit of infinite depth, given optimal parameter settings, it converges to the optimum of a combinatorial problem. It exhibits a few properties that makes it interesting for combinatorial optimization [5, 6, 7, 8]. These can be useful when using it as a subroutine [9, 10, 11, 12, 13], especially the concentration of parameters [6]. This concentration property suggests that optimal parameters found for one instance can be reused on another. Most importantly, this means we can reduce the classical optimization loop and number of calls to a quantum device (saving runtime of QAOA-featured algorithms).

Many works have studied or illustrated this concentration property [6, 7, 14, 15, 16, 17, 18, 19, 20]. However, it comes with an underlying assumption that the instances (most often formulated as Ising problems that can be visualized as graphs) should come from a «reasonable» distribution [6]. On the one hand, the «reasonable» aspect may not be straightforward to characterize [18]. On the other hand, in many QAOA-featured algorithms [9, 10, 11, 12, 13], many instances

*Correspondence: c.moussa@liacs.leidenuniv.nl; v.dunjko@liacs.leidenuniv.nl
LIACS, Leiden University, Leiden, Netherlands
Full list of author information is available at the end of the article

are generated and could give rise to many distributions. Finally, within a distribution, several areas of concentration may rise. Hence, balancing between finding good QAOA parameters and reducing circuit calls will be key to QAOA-featured algorithms.

Taking into account these aforementioned considerations, we propose to apply unsupervised learning for setting QAOA angles, namely clustering. We consider different approaches depending on the type of data fed, and compare them on two types of problems: MaxCut on Erdős-Rényi graphs and Quadratic Unconstrained Binary Problems on random dense matrices. We demonstrate that our techniques can be used to learn to set QAOA parameters with respectively a less than $1-2\%$ reduction (in relative value) in approximation ratio in cross-validation, while reducing circuit calls. Finally, we demonstrate their usage in Recursive-QAOA (RQAOA) [13] up to depth 3 on the Erdős-Rényi graphs, where we limit the number of QAOA circuit calls per iteration to 3 (in contrast to a de-novo optimization which could require many more calls), and achieve a 0.94 median approximation ratio. With our approaches, we obtain similar performances to the case where we extensively optimize the angles, hence saving numerous circuit calls.

The structure of the paper is as follows. Section 2 provides the necessary background and related works. Section 3 analyses the optimal angles found in both problems, pointing concentration effects and the suitability of clustering. Section 4 shows different unsupervised learning strategies using different data encoding for clustering and the comparison between them. Section 5 sums up our experiments on rqaoa. We conclude this work with a discussion in Section 6.

2 Background

2.1 QUBO and QAOA

Quadratic Unconstrained Binary Optimization (QUBO) problems are specified by the formulation $\min_{x\in\{0,1\}^n} \sum_{i\leq j} x_i Q_{ij} x_j$ where n is the dimensionality of the problem and $Q \in \mathbb{R}^{n\times n}$. This formulation is connected to the task of finding so called «ground states»of «Ising models», i.e., configurations of binary labels $\{1,-1\}$ minimising the energy of spin Hamiltonians, commonly tackled in statistical physics and quantum computing, i.e.,:

$$\min_{s\in\{-1,1\}^n} \sum_i h_i s_i + \sum_{j>i} J_{ij} s_i s_j, \tag{1}$$

where h_i are the biases and J_{ij} the interactions between spins. QUBO can express an exceptional variety of combinatorial optimization (CO) problems such as Quadratic Assignment, Constraint Satisfaction Problems, Graph Coloring, Maximum Cut [21].

The QAOA algorithm [4] was inspired by adiabatic quantum computing with the goal to tackle CO problems. It consists of a quantum circuit whose construction depends on the classical cost function. Indeed, the latter is encoded in a quantum Hamiltonian defined on N qubits by replacing each variable s_i in Eq. (1) by the single-qubit operator σ_i^z:

$$H_C = \sum_i h_i \sigma_i^z + \sum_{j>i} J_{ij} \sigma_i^z \sigma_j^z. \tag{2}$$

Here, the bitstring corresponding to the ground state of H_C also minimizes the cost function. Another Hamiltonian named *mixer* $H_B = \sum_{j=1}^N \sigma_j^x$ is also employed in QAOA. These operators are then used for building a quantum circuit with real parameters and organized as layers. This circuit is initialized in the $|+\rangle^{\otimes N}$ state, corresponding to all bitstrings in superposition with equal probability of being measured. Then, applying p layers sequentially yields the following quantum state:

$$|\gamma,\beta\rangle = e^{-i\beta_p H_B} e^{-i\gamma_p H_C} \cdots e^{-i\beta_1 H_B} e^{-i\gamma_1 H_C} |+\rangle^{\otimes N},$$

defined by $2p$ real parameters $\gamma_i, \beta_i, i = 1...p$ or *QAOA angles* as they correspond to angles of parameterized quantum gates. Such output corresponds to a probability distribution over all possible bitstrings. The classical optimization challenge of QAOA is to find the sequence of angles γ, β minimizing the expected value of the cost function from the measurement outcome. In the limit of infinite depth, the distribution will converge to the global optimum.

An interesting property of the algorithm is the concentration of the QAOA objective for fixed angles [6] due to typical instances having (nearly) the same value of the objective function. Additionally, the QAOA landscape is instance-independent when instances come from a «reasonable» distribution (with the number of certain types of subgraphs of fixed size themselves concentrate, which in turn implies the values concentrate). Hence, we can focus on finding good parameters on a subset of instances that could be re-applied to new ones, with a few extra calls to the quantum device in order to refine. As stated earlier, in the most general case, characterizing distributions which are «reasonable» may be involved, or even characterizing the distribution at hand may be hard. Previous work [6, 19, 20] referenced in [18] reported concentrations over optimal parameters even when QAOA is applied on random instances. These distributions over optimal parameters are empirically shown to behave non-trivially with respect to n. [18] pointed out this problem as «folklore of concentrations» .

Hence, even though angles concentrate in many settings asymptotically, for finite size problems, different

areas of concentration may rise. Therefore, choosing good angle values is challenging, especially when considering runtime of quantum algorithms. As such, some studies built on this property and resorted to using Machine Learning (ML) or characterizing instances by some properties for finding good QAOA parameters. We present a few of them in the next subsection.

2.2 Related Work

Many previous works have extensively employed the concentration property [7, 14, 15, 16, 17, 18, 19, 20]. Among them, a few employed ML or designed strategies for setting good QAOA parameters for diferent objectives. In [14], a simple kernel density model was trained on the best angles and instances solved by QAOA to exhibits better QAOA optimization than the Nelder-Mead optimizer. Parameter fixing strategies for QAOA are also studied in [15, 7] where the best found angles at depth p are used as starting points for depth $p+1$ before using a classical optimizer.

[19] present a strategy to find good parameters for QAOA based on topological properties of the problem graph and tensor network techniques. [16] point out that the success of transferability of parameters between different problem instances can be explained and predicted based on the types of subgraphs composing a graph. Finally, meta-learning is used in [17] to learn good initial angles for QAOA. They focused on initialization-based meta-learners in which a single set of parameters is used for a distribution of problems as initial parameters of a gradient-based optimizer. The meta-learner is a simple neural network that takes as inputs some meta-features of the QAOA circuit to predict the angles to apply (depth and which angle to output the value). However, no instance-related features are involved in their work.

In our case, we focus on clustering with the goal of proposing many parameter values to try for new QAOA circuits. In contrast to all the approaches we discussed above, we do not use a classical optimization loop after setting them. Hence, our approaches allow balancing between circuit calls of small quantum computers and performances. Such settings for instance naturally occur in divide-and-conquer-type schemes to enable smaller quantum computers to improve optimization [9, 10, 11, 12], or in Recursive-QAOA [13] as we demonstrate later.

3 Revisiting the concentration property

In contrast to previous related works, we propose unsupervised approaches that also exploit these concentration effects. We take a data-driven approach where from examples of good angles, we will infer new good angles for new instances. Namely, we use clustering in order to obtain clusters that can be used to reduce calls to the quantum device to small numbers (in our case, less than 10) when applying QAOA on new instances, without further optimization.

We take a usual ML approach to this problem. First, from generated instances, we apply *exploratory data analysis* [22] (EDA) that suggests clustering may be a good approach for recommending good angles to new instances. Namely, we look at the density of angle values and apply t-SNE [23] (a dimensionality reduction technique) for visualizing concentration effects. We follow by explaining how clustering is used in order to recommend angles for new instances. The approaches we outline differ in input to the clustering algorithm. We consider clustering from the angle values directly but also from instance encodings. Finally, we compare these approaches allowing us to provide recommendations for their usage.

3.1 Data generation

We generated two datasets that shows different concentration behavior. The first one consists of 200 Erdős-Rényi graphs for MaxCut problems. The graphs have $10, 12, 14, 16$ and 18 nodes. We utilized the following probabilities of edge creation: $0.5, 0.6, 0.7, 0.8$. We have generated 10 graphs per number of nodes and probability. The second dataset consists of 100 instances of QUBO problems, specified by their weight matrix Q (20 per aforementioned number of nodes). Their coefficients are sampled uniformly in $[-1, 1]$. For the purpose of computing approximation ratios, we are interested in C_{opt} – the maximal value of the MaxCut (or QUBO) – over all possible bit configurations, and as a reference, this was computed using brute-force.

We then obtained for each problem the best set of angles by running the BFGS optimizer [24] 1000 times for $p = 1, 2, 3$, and selecting the ones which achieves the best QAOA objective. BFGS with random restarts is deemed a very good optimizer for continuous differentiable functions. These angles are saved as a database and apply unsupervised approaches to learn to set optimal angles for unseen instances. Our approach is clearly optimization method specific but can be applied to other state-of-the-art optimizers. Different optimizers would give different data but they can be combined and one would select the best set of angles found among all considered.

3.2 Exploratory Data Analysis

Having obtained the optimal angles, we apply EDA to observe concentration effects. We look at their corresponding performance ratios using the average cost yielded by QAOA for angles γ, β denoted with $E_{\gamma,\beta}(C)$. For MaxCut on unweighted Erdős-Rényi graphs, we

compute approximation ratios as $\frac{E_{\gamma,\beta}(C)}{C_{opt}}$. This value is upper bounded by 1, which is the optimal value. For QUBOs, we compute optimality gaps $\frac{C_{opt}-E_{\gamma,\beta}(C)}{C_{opt}}$ as the optima were all negative and the closer to 0, the better. We show boxplots in Fig 1 the ratios wrt depth. Increasing depth results in better ratios.

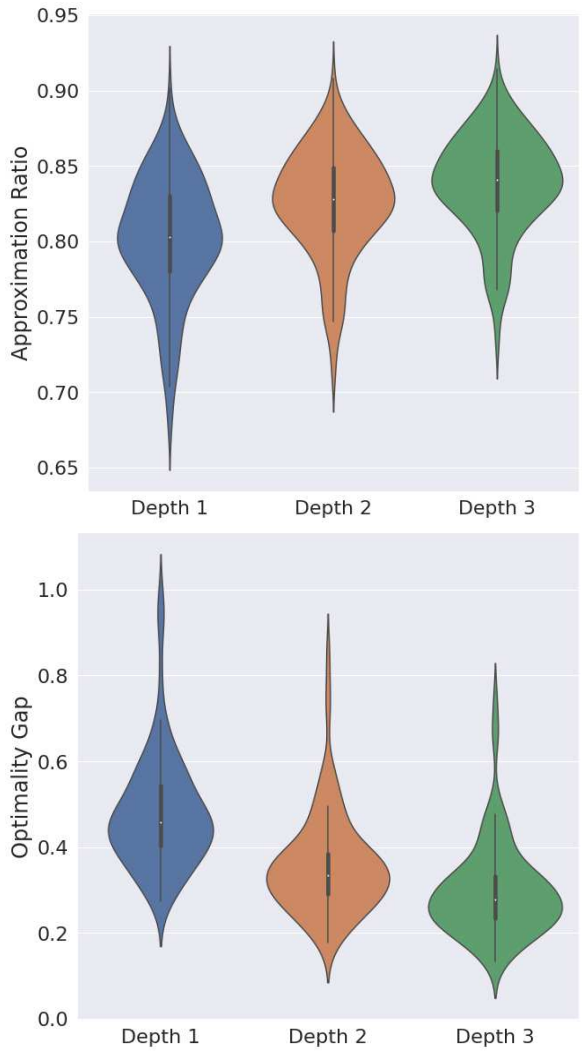


Figure 1: Violin plots of ratios on MaxCut and optimality gaps over QUBOs (bottom plot) for $p = 1, 2, 3$. The respective median by depth is $0.802954, 0.827901, 0.840478$ for MaxCuts and $0.457434, 0.335144, 0.278984$ for QUBOs, illustrating improved performances with increased depth.

Next, we looked at the distribution of γ_i, β_i values. Fig 2 shows that the concentration per each parameter is significant since their corresponding density function are quite peaky. Also, we also observed multiple clusters of angles as the density functions are multimodal. Finally, we applied t-SNE with two components to visualize the angle values in $2D$ for $p = 2, 3$. This highlights potentially a number of clusters for each depth and problem. Note that it may be the possible that we may not obtain global optima with these angles, or know if they are unique.

We notice that the probability of edge creation, represented by a different color, does not seem to influence the clusters. For dense QUBOs, we observe one important cluster and a few instances that start to form another. Finally, in the dense instances case, we witness more important spread in angle values at depth 1. This can be explained by differences between instances. Although the concentration effect is present, such order of magnitude will impact the performances of parameter setting strategies, and make an interesting playground to benchmark them.

Using clustering techniques can then reveal potential areas of QAOA angles values where good angles can be found to try on new instances. The angle values related to clusters can be used as recommendations for new instances. This becomes interesting as this enables lowering runtime and allow comparing based on function evaluations, or on the number of quantum circuit calls, in algorithms were QAOA would be used as a subroutine.

4 Clustering-based (unsupervised) learning for angles

As the EDA highlights a clustering effect, we propose different clustering approaches that use different data for angle recommendations. Namely, we describe first using the angle values directly for building clusters serving as angles to try. Then, we switch to using instance-related features. Finally, for the unweighted case, we use graph auto-encoders whose outputs can be used for clustering instead of computing graph features. In the following, we detail each clustering approach for flexible angle recommendation.

4.1 Identifying clusters of angles or problem instances

We first considered clustering using angle values. Given a database of optimal angles for Q problem instances $\{I_1, \cdots, I_Q\}, \{(\gamma^*, \beta^*)_1, \cdots, (\gamma^*, \beta^*)_Q\}$, this can be seen as computing or selecting a good set of angle values the database to apply on new instances. In this case, we do not use the problem instances during clustering. Given a user-specified number of angles to be tested K, this set of angle values are then applied on new QAOA circuits. To specify them, we can use a clustering algorithm on the database $\{(\gamma^*, \beta^*)_1, \cdots, (\gamma^*, \beta^*)_Q\}$. For instance, K-means will output centroids to use directly as angle recommendations for QAOA on new instances.

To incorporate knowledge from instances when recommending angles, we change the data fed to the clustering algorithm. We distinguish computing instance

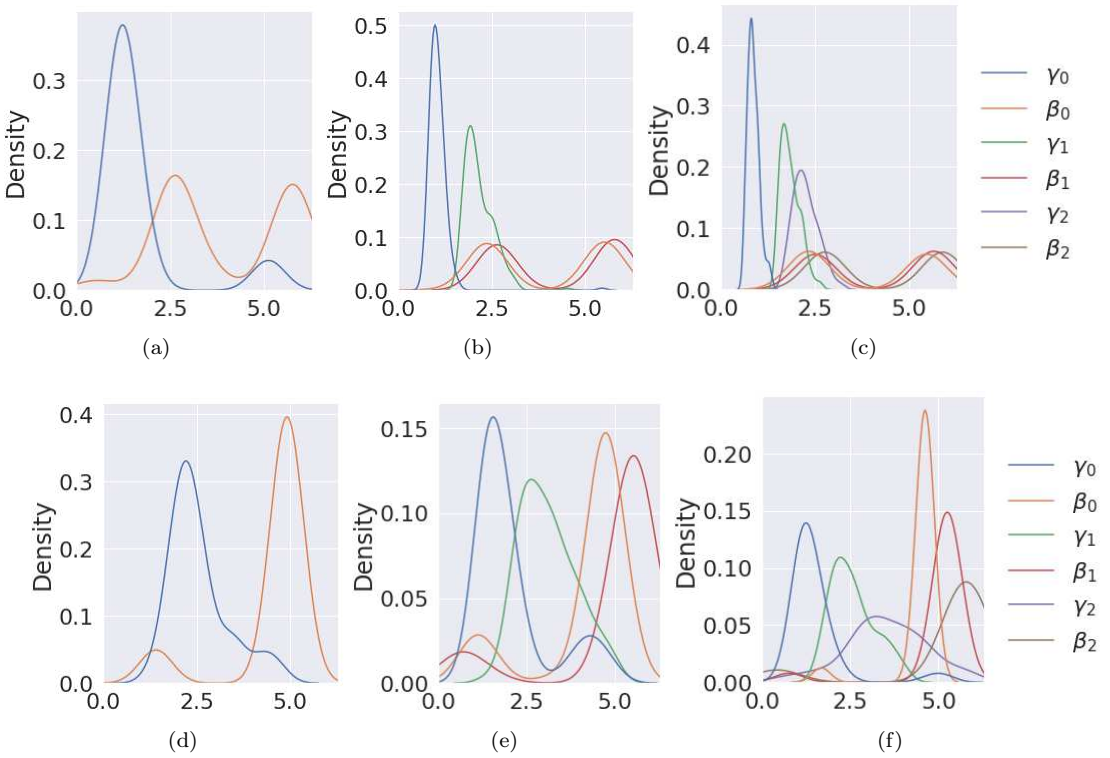


Figure 2: Distribution of angle values γ_i, β_i for each depth. Plots a), b) and c) concern MaxCut problems while the others refer to the dense QUBO matrices. We witness concentration effects of the angle values, suggesting the suitability of clustering as an angle setting strategy.

features from learning an embedding, that is a user-defined $F-$dimensional representation or encoding of the instances as data. We denote an encoding of an instance I_t as $f(I_t)$. The angle recommendation framework using a clustering algorithm for such instance representation is presented in Alg. 1. First, clusters are learnt from the encodings extracted on training data. Then, we find the instances in the database that are the closest in distance to the clusters, and their corresponding optimal angles. The latter are then used for QAOA circuits on new instances, from which we keep the best QAOA output.

Algorithm 1 K-angle recommendation framework for QAOA.

Input: Clustering algorithm, number of clusters K,
Training Data: $\{I_1, \cdots, I_Q\}; \{(\gamma^*, \beta^*)_1, \cdots, (\gamma^*, \beta^*)_Q\}$,
Testing Data: $\{I_1^{'}, \cdots, I_R^{'}\}$,
Initialize $anglesToRecommend = [], encodings = []$
for $t = 1$ **to** Q **do**
  Compute $f(I_t)$ and append to $encodings$.
end for
Apply Clustering algorithm on $encodings$
for $c = 1$ **to** K **do**
  Get cluster $f^c(I_t)$
  Get closest point in $encodings$ to $f^c(I_t)$ and extract index i_c
  Append $(\gamma^*, \beta^*)_{i_c}$ to $anglesToRecommend$
end for
for $t = 1$ **to** R **do**
  $bestOutput_t = INF$
  for $c = 1$ **to** K **do**
    Apply QAOA on $I_t^{'}$ with $anglesToRecommend[c]$
    if $E_{anglesToRecommend[c]}(C_{I_t^{'}}) < bestOutput_t$ **then**
      $bestOutput_t = E_{anglesToRecommend[c]}(C_{I_t^{'}})$
    end if
  end for
end for

4.2 Instance encodings

In this work, we show two main approaches to encode the instances for clustering. First, we computed a set of features following [25, 11]. *For Erdős-Rényi graphs, we took the graph density, the logarithm of number of nodes and edges, logarithm of the first and second largest eigenvalues of the laplacian matrix normalized by the average node degree and the logarithm of the ratio of the two largest eigenvalues. For QUBOs, we reduced them to the MaxCut formulation and used the logarithm of number of nodes, and the weighted laplacian matrix eigenvalues-based features.*

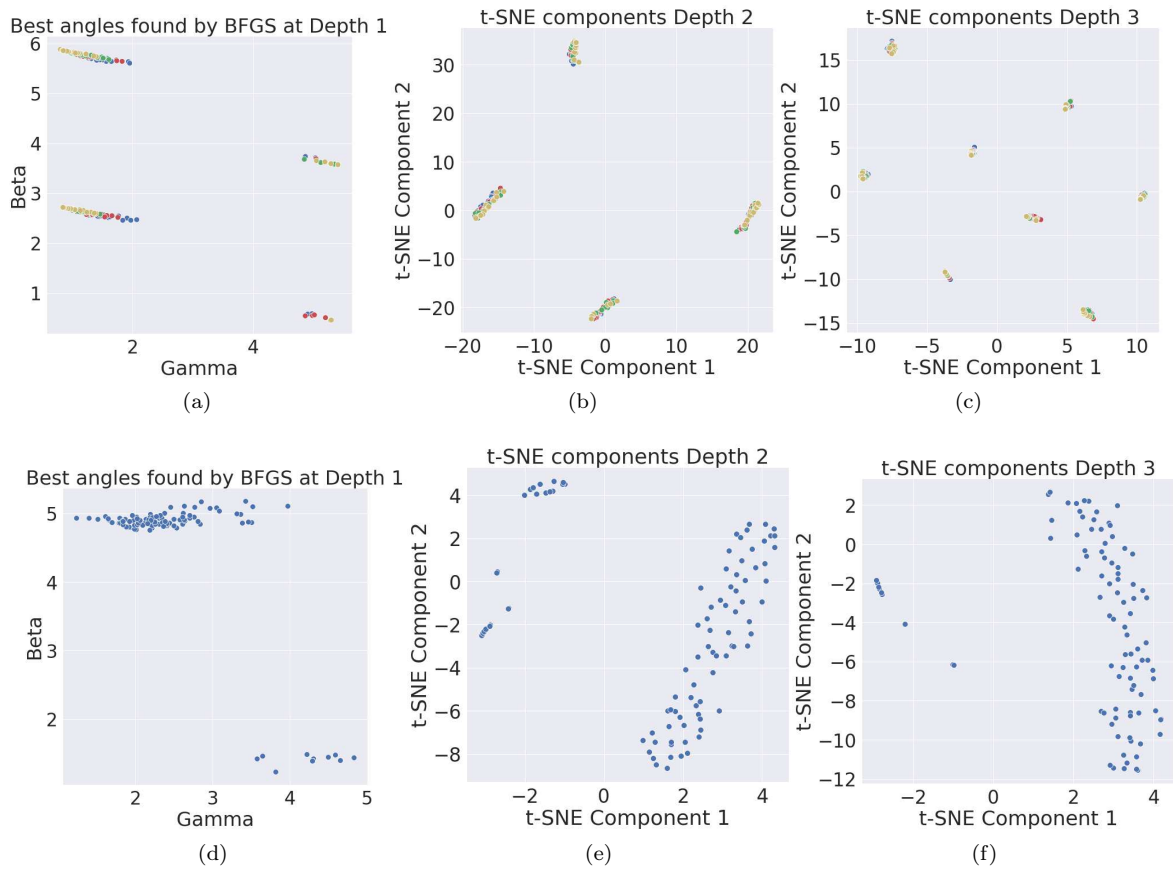


Figure 3: 2D angles visualization γ_i, β_i for each depth. Plots a), b) and c) concern MaxCut problems while the others refer to the dense QUBO matrices. For $p = 2, 3$, t-SNE is applied for projecting the angle values to 2D. Different areas of concentration are revealed again. We use different colors for differentiating the probability of edge creation of the Erdős-Rényi graphs, showing no correlation with clusters.

We also show how to use graph embeddings using Graph Neural Networks (GNNs) [26], avoiding the need for the user to have to compute the features. We employ the Variational Graph Auto-Encoders (VGAE) [27]. This technique only works on unweighted graphs by its design principle. Consequently, we only applied it to the MaxCut instances later in this work. a VGAE learns latent embeddings $\mathbf{Z} \in \mathbb{R}^{N \times F}$ where F is the dimension of the latent variables and N the number of nodes. Given the adjacency matrix A and nodes feature vector X, the model outputs the parameters of a Gaussian distribution μ, σ for the latent representation generation. We feed to the model the Erdős-Rényi graphs, and we add as node features the degree of the nodes. Once learning is completed, we compute the embeddings by a common average readout operation [26, 28]. This allows having a fixed dimension F for the encoding to be used by a clustering algorithm.

Having defined different strategies for clustering, we apply them to the data we generated and compare their performances. In the following section, we present our results obtained by taking a Machine Learning approach, starting from a simple baseline and cross-validating each method.

4.3 Results

In this Section, we apply the above-mentionned proposed strategies to the generated data where EDA revealed different areas of concentration. As a first baseline for angle setting strategy, we experiment with simple aggregation of angle values (median and average). Then we follow this up by K-means by varying the number of clusters from 3 to 10 as the underlying clustering algorithm. Finally, we change the K-means data to cluster based on instance encodings instead of angle values. We computed first a set of graph features that were used in a previous study [8]. Then we

investigate graph autoencoders to learn encodings of the Maxcut instances. We cross-validate each method using 5-fold cross-validation where we report the ratios $\frac{(C_{opt}-E_{\gamma,\beta}(C))}{(C_{opt}-E^{cluster}_{\gamma,\beta}(C))}$ on test instances. A value higher than 1 would mean that the average cost yielded by clustering has improved over the one found by optimization. We also consider the case where one trains on smaller instances to apply on the bigger ones.

4.3.1 From angle values

As simple baseline, we compute the average and the median of the optimal angles from the database $\{(\gamma^*,\beta^*)_1,\cdots,(\gamma^*,\beta^*)_Q\}$. From depth-aggregated results, averaging the angle values yielded a median ratio of 0.524307 for MaxCut and 0.671572 for QUBOs, while taking the median values increased it to respectively 0.949690 and 0.940555. This can be explained by the fact that the median value is statistically more robust than the mean when handling data sets with large variability.

As expected with K-means, increasing the number of clusters yielded better median ratios. With $K = 10$, the median ratios are 0.998052 and 0.984787 on each dataset, a less than $1-2\%$ reduction in performances w.r.t. the optimal angles. Fig. 4 shows the improvement with increased number of clusters. We observe also that with increased depth, median ratio performances are reduced. We conjecture that, when the dimension of the parameter space increases, more clusters are naturally needed to ensure a sensible recommendation.

Also, such a deterioration of performance w.r.t. circuit depth is more substantial on the QUBO instances than on the MaxCut ones, which can be explained by the clustering patterns in the MaxCut scenario being more significant and regular (Fig. 3). In addition, this observation suggests that for future work, for dense QUBO instances where the cluster center is not representative for all points pertaining to it, it is more reasonable to take a supervised learning method, which takes the problem instance as input as predicts the optimal angle values.

Overall, increasing the number of angles attempted will improve the quality of the QAOA output. Clearly, the results with less than 4 clusters present examples were the ratio is low, worsen the median performances. For instance, with 3 clusters on each problem, the median ratios are respectively 0.618275 and 0.915402. In the context where the budget of quantum circuit calls is very limited, this could be problematic and call for more robust approaches. To this end, we consider using instance features for clustering.

4.3.2 From instance encodings

To witness whether using instance features can improve the quality of clustering, we divided the ratios obtained with instance features by the ones using angle values. We show these results in Fig. 5 and Fig. 6 where we can clearly see better ratios with less than 4 clusters, and similar results on average otherwise.

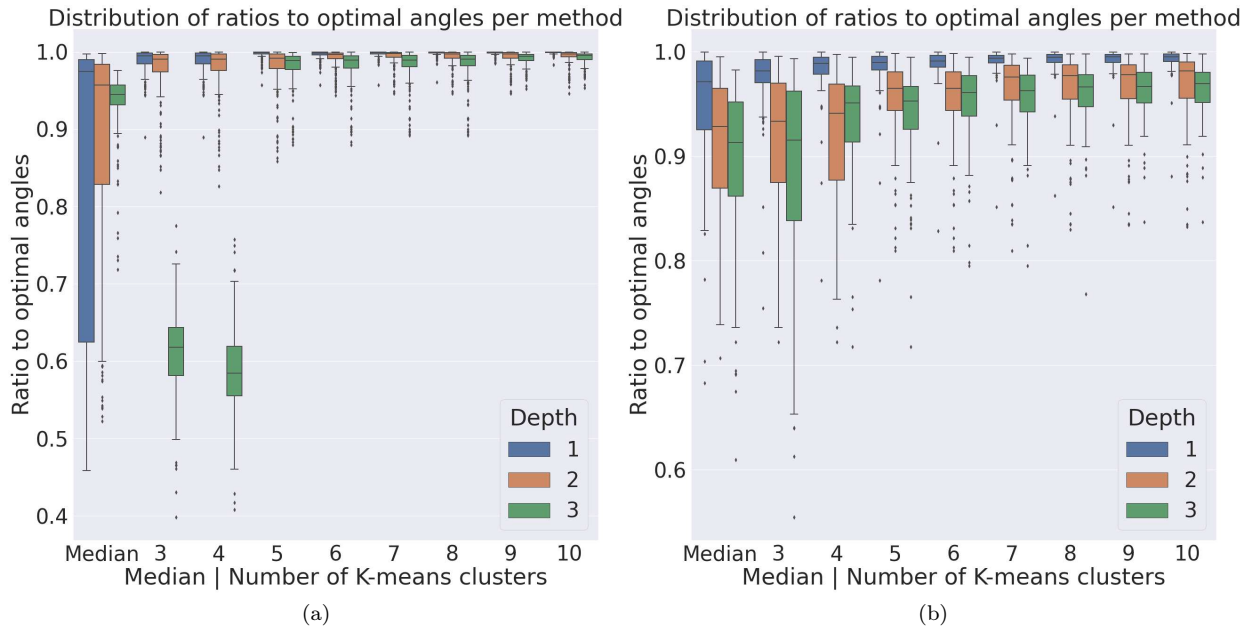
As for learned encodings or embeddings with autoencoders, the GNN model configuration we use is the same two-layer graph convolutional layer as [27]. Namely, the first one has 32 output-dimension using the ReLU activation function. This is followed by two 16-dimensional output layers for the latent variables generation. We train using Adam with learning rate 0.01 for 100 epochs and batch size set to the dataset size. Our implementation uses the Deep Graph Library (DGL) [28]. The embeddings obtained by averaging are of dimension $F = 16$. This allows having a fixed dimension for the encoding as input of the same K-means strategy described above. We observe in Fig. 7 that the results are similar to the ones obtained using instance features. Yet, on some instances we see better results. Hence, many clustering results can be combined to improve the performances in ratios cancelling each other weaknesses at the cost of trying more angles to find the best ones.

4.4 Aggregating results

Following the presentation of the different clustering approaches, we compare their performances to determine which approach works best. We propose to take the Empirical cumulative distribution functions (ECDF) of the ratios as the performance measure to compare those different approaches. They enable us to aggregate results of different number of clusters and depth. A better method will have more proportion of higher ratios, resulting in an ECDF curve located more to the right. From Fig. 8, we observe that *using instance encodings is more successful in yielding better angles than using the angle values*. This is also witnessed in Fig. 9 with increased depth and low number of clusters. Also, VGAE seems to be slightly better than instance features on the MaxCut problems. However, these methods can complement each other, especially as we do not need to increase dataset size. Hence, combining them at the cost of circuit calls becomes an option for running QAOA, as we showcase withrqaoa in the next section.

4.5 Case when test instances are bigger than training instances

One important consideration of these methods is to analyze scaling. This is relevant in settings where one is interested in solving larger instances given small ones. In our case, we apply these approaches in the case $K = 3$ by a $60-40\%$ train-test split. From Fig. 10 and 11, we find similar conclusions with respectively VGAE on MaxCut and instance features on the QUBO



(a) (b)

Figure 4: Boxplot visualization of ratios to optimal angles' expectation value per clustering method and depth on MaxCut (a) and dense QUBOs (b), when using the angle values. We show also the boxplots when computing the median of angle values, yielding a median ratio of 0.524307 for MaxCut and 0.671572 for QUBOs. The median ratios are respectively 0.980221 for 3 clusters, 0.994972 for 5 clusters and 0.998052 for 10. For QUBOs, we get 0.956099, 0.970842 and 0.984787 taking the same number of clusters. With reference to the optimal angles' expectation value, this corresponds on average to a less than $1-2\%$ reduction in performances when using 10 clusters.

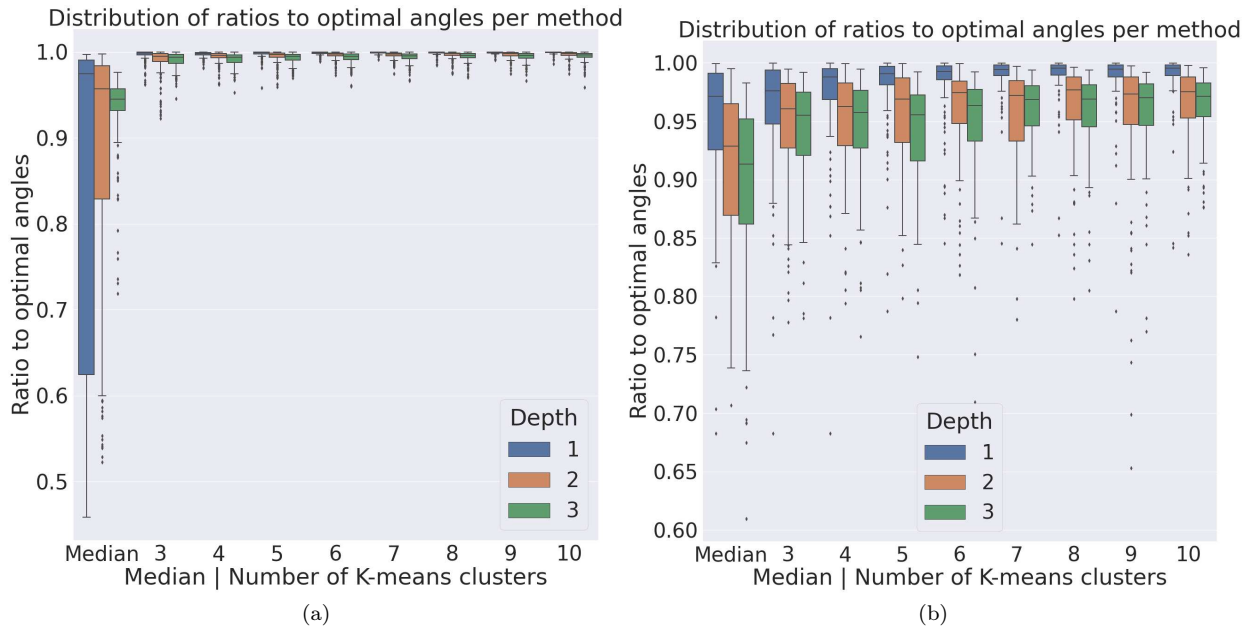
problems yielding better results. Note that we did not use the the logarithm of number of nodes and edges as feature when using instance features as the values between training and test are too different.

5 Demonstration with RQAOA

RQAOA [13] is a recursive algorithm where, given an Ising problem $\sum_{i,j} w_{ij} Z_i Z_j$, one starts by applying QAOA on the former. the quantum state output $|\gamma,\beta\rangle$ is then used to compute correlations $M_{ij} = \langle\gamma,\beta|Z_iZ_j|\gamma,\beta\rangle$. Then, variable elimination is carried out by selecting a pair of variables satisfying $(i_l, j_l) = \arg\max |M_{ij}|$, and substituting Z_{j_l} with $\text{sign}(M_{i_l,j_l})Z_{i_l}$ in the Ising formulation. This reduces the number of variables by 1. We then get a new reduced problem and we reiterate the procedure for a number of user-defined number of iterations. The choice of iteration fixes the size of the final instance which is then solved using a brute-force (or some other classical) approach, and the substitutions are used onto it to obtain a final solution.

Asrqaoa requires optimizing many QAOA instances that iteratively shrink in sizes, we demonstrate the application of our clustering approaches in this context. We do so for the MaxCut problems where we limit the number of iterations to half of the size of the Erdős-Rényi graphs. We do not consider the dense QUBOs asrqaoa would reduce an original dense graph to non-dense intermediate subproblems not part of the database. As per the number of QAOA parameters attempted per iteration, we limit it to 3 and apply the three clustering approaches: angle-value, instance features and VGAE-output based. We do so by using our previous database and train each method on all instances to get 3 QAOA parameter recommendations. The latter are then used for QAOA on therqaoa generated instances.

Fig. 12 shows that with the three approaches, we obtain a median 0.94117 approximation ratio withrqaoa. The minimal ratio obtained is 0.8367 and the optima were found on 33 instances. When looking at each method independently, we observe that the angle-value clustering performances at $p = 3$ are again lower than the others. Graph features and VGAE seem also similar in performances, with a small advantage at depth 2 for VGAE. Looking at the frequencies where the best ratio by instance was obtained, VGAE is more successful. Respectively, each method achieves the best found ratios over 88, 118 and 165 instances. Finally, we also tried using random angles, by sampling uniformly values in $[0, 2\pi]^p$,and optimizing further the angles from each



(a) (b)

Figure 5: Boxplot plot visualization of ratios to optimal angles' per clustering method and depth on MaxCut (a) and dense QUBOs (b), when using instance features. For Erdős-Rényi graphs, K-means yielded ratios $0.996214, 0.996368$ with $3, 4$ clusters and 0.998429 with 10. On dense QUBOs, we obtained respective median ratios of $0.963129, 0.971778$ and 0.982964. With reference to the optimal angles' expectation value, this corresponds on average to a less than $1 - 2\%$ reduction in performances when using 10 clusters.

approach with BFGS up to 100 iterations maximum. We clearly see better performances with clustering approaches compared to random angles. This is also the case when using BFGS (starting with random angles) limited to 3 circuit calls when optimizing, the same budget as our clustering-based approaches. Dividing the MaxCut ratios obtained with BFGS with the ones without further optimization yielded a median value of 1. Hence, the results were similar to the BFGS-optimized approaches, saving many circuit calls.

To conclude, our unsupervised approaches can be used to run quantum algorithms where QAOA is used as a subroutine. They are then considered as hyperparameters that can be tweaked to achieve better performances for QAOA-featured algorithms, depending on a user-defined budget definition. In ourrqaoa showcase, the maximal depth of QAOA as well as the number of parameters to try at each iteration was set to 3, and optimizing further did not improve. For MaxCut on Erdős-Rényi graphs, leveraging VGAE inrqaoa achieved the best ratios over 82.5% of the instances.

6 Discussion

In this work, we study different strategies for fixing the parameters of QAOA based on unsupervised learning. We focused on clustering given previous works highlighthing the concentration property and exploratory data analysis of the best angles found for MaxCut on Erdős-Rényi graphs and dense QUBOs. We however use a methodology closer to machine learning by cross-validating compared to related work.

Furthermore, we demonstrated that these techniques can be leveraged to restrict the number of QAOA circuit calls to small numbers (less than 10) with a less than $1 - 2\%$ reduction in approximation ratio on average from the best angles found when cross-validating. We also showed how to compare different clustering strategies and that leveraging instance encodings (by computing features or computing them with a model, in our case a VGAE) for angle setting strategies yields better results than using angle values only. Although the VGAE embedding-based is quite competitive, we recommend using the simpler instance features in practice since the VGAE brings extra computation overhead. For generalization, in regard to the problem scale, both instance features- and VGAE-based clustering approaches manage to retain the performance for unseen problem instances larger than the training set. For dense QUBOs, increasing the clusters is less impactful compared to MaxCut, which we conjecture that the clusters in QUBO are of large spread and less separable,

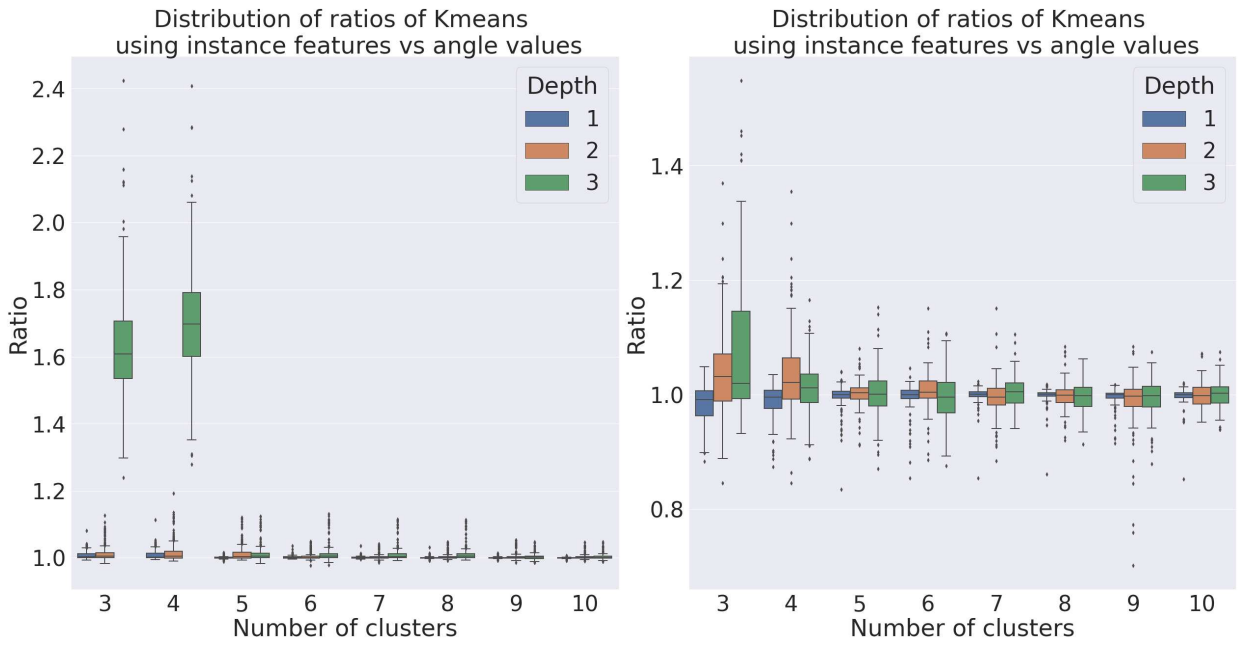


Figure 6: Boxplot of ratios comparing K-means with instance features against angle values on MaxCut (a) and QUBOs (b). A value higher than 1 means using instance features results in better QAOA objective. We see an overall improvement with $3, 4$ clusters mainly at $p = 3$.

hindering the performance of the clustering approach in higher dimensions. For both problems, it is necessary to increase the cluster number to retain a good performance when the circuit becomes deeper.

From an application perspective, we envision these techniques to be employed in algorithms were QAOA is run on a small part of the problem to solve such as divide-and-conquer [9, 10] and iterative algorithms [11, 12, 13]. Restricting to a few number of circuit calls will help decrease the runtime of quantum-featured or quantum-enhanced algorithms, making them closer to compete with classical heuristics. We showcased our approach in the context of Recursive QAOA as hyperparameters under limited budget (QAOA depth and number of QAOA parameters per iteration limited to 3), where we were able to achieve 0.94 median approximation ratio. With our approaches, we obtain a quite comparable performance to the case where we extensively optimize the angles, hence saving numerous circuit calls.

For future work, other clustering techniques can be studied and extended to predicting the angle values by instance in a semi-supervised approach, and for different problem instances. We also did not apply GNN to the dense QUBOs as graph autoencoders are mostly applied to unweighted graphs. Using VGAE that can reconstruct graph adjacency and node features is then another research direction. Since we use unsupervised methods, we expect the same methodology to be used on noisy hardware. Studying different approaches resilience under different noisy settings would be also considered of main interest. Finally, these approaches can be studied within different QAOA-featured algorithms and under different settings (depth of QAOA, number of clusters, Ising instances properties to name a few).

Acknowledgements
CM and VD acknowledge support from TotalEnergies.

Funding
This work was supported by the Dutch Research Council (NWO/OCW), as part of the Quantum Software Consortium programme (project number 024.003.037). This research is also supported by the project NEASQC funded from the European Union's Horizon 2020 research and innovation programme (grant agreement No 951821).

Abbreviations

- CO: Combinatorial Optimization
- QAOA: Quantum Approximate Optimization Algorithm
- ECDF: Empirical cumulative distribution functions
- VGAE: Variational Graph Auto-Encoders
- QUBO: Quadratic Unconstrained Binary Optimization
- EDA: Exploratory Data Analysis

Availability of data and materials
The datasets used and/or analysed during the current study are available from the corresponding authors on reasonable request.

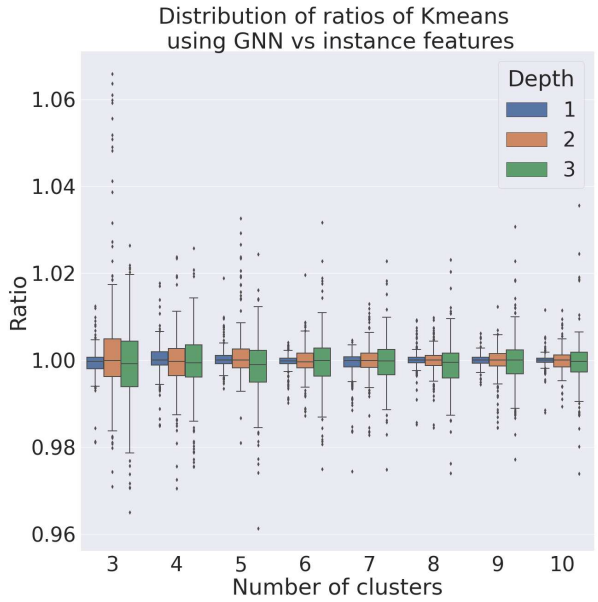


Figure 7: Boxplot visualization of ratios on MaxCut obtained using Variational Graph Auto-Encoders compared to using instance features. A value higher than 1 means using VGAE results in better QAOA objective. Overall, performances are similar as the ratios are close to 1 on average.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

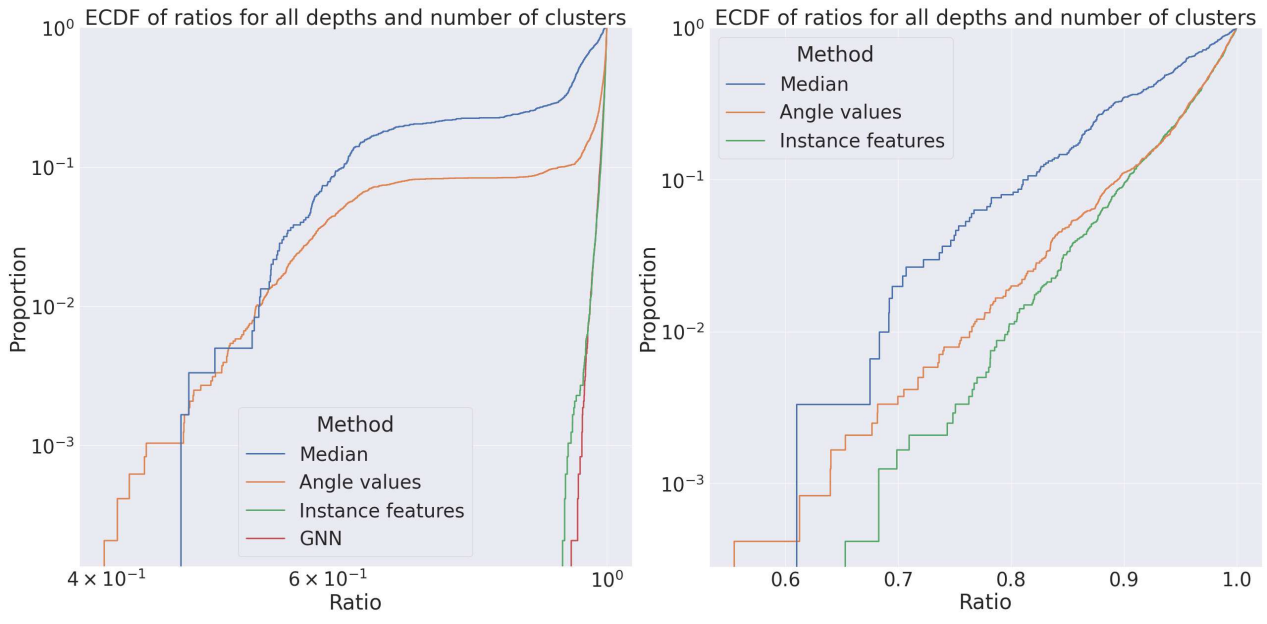
CM designed all the experiments. The manuscript was written with contribution from all authors. All authors read and approved the final manuscript.

Author details

LIACS, Leiden University, Leiden, Netherlands.

References

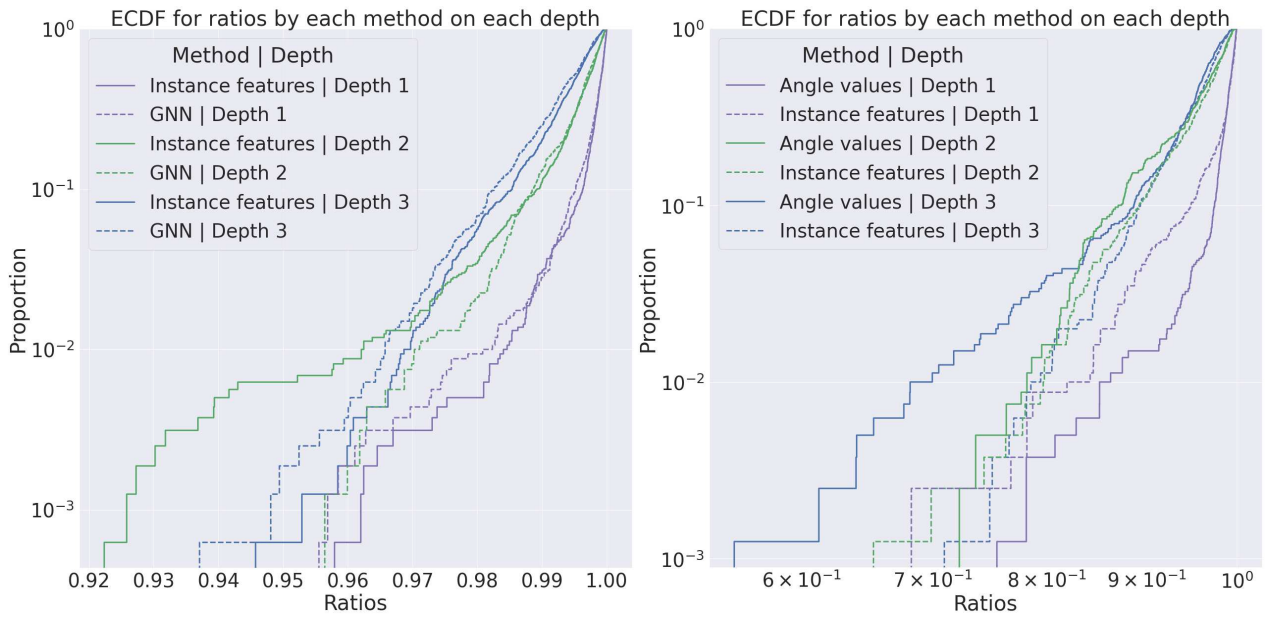
1. Preskill, J.: Quantum Computing in the NISQ era and beyond. Quantum **2**, 79 (2018). doi:10.22331/q-2018-08-06-79
2. Moll, N., Barkoutsos, P., Bishop, L.S., Chow, J.M., Cross, A., Egger, D.J., Filipp, S., Fuhrer, A., Gambetta, J.M., Ganzhorn, M., Kandala, A., Mezzacapo, A., Müller, P., Riess, W., Salis, G., Smolin, J., Tavernelli, I., Temme, K.: Quantum optimization using variational algorithms on near-term quantum devices. Quantum Science and Technology **3**(3), 030503 (2018). doi:10.1088/2058-9565/aab822
3. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. Quantum Science and Technology **4**(4), 043001 (2019). doi:10.1088/2058-9565/ab4eb5
4. Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm (2014). arXiv:1411.4028
5. Farhi, E., Harrow, A.W.: Quantum Supremacy through the Quantum Approximate Optimization Algorithm (2016). arXiv:1602.07674
6. Brandão, F.G.S.L., Broughton, M., Farhi, E., Gutmann, S., Neven, H.: For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances. arXiv:1812.04170 (2018)
7. Zhou, L., Wang, S.-T., Choi, S., Pichler, H., Lukin, M.D.: Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. arXiv:1812.01041 (2018)
8. Moussa, C., Calandra, H., Dunjko, V.: To quantum or not to quantum: towards algorithm selection in near-term quantum optimization. Quantum Science and Technology **5**(4), 044009 (2020). doi:10.1088/2058-9565/abb8e5
9. Li, J., Alam, M., Ghosh, S.: Large-scale Quantum Approximate Optimization via Divide-and-Conquer (2021). arXiv:2102.13288
10. Guerreschi, G.G.: Solving Quadratic Unconstrained Binary Optimization with divide-and-conquer and quantum algorithms (2021). arXiv:2101.07813
11. Moussa, C., Wang, H., Calandra, H., Bäck, T., Dunjko, V.: Tabu-driven quantum neighborhood samplers. In: Evolutionary Computation in Combinatorial Optimization, pp. 100–119. Springer, ??? (2021). doi:10.1007/978-3-030-72904-2_7. https://doi.org/10.1007/978-3-030-72904-2_7
12. Shaydulin, R., Ushijima-Mwesigwa, H., Safro, I., Mniszewski, S., Alexeev, Y.: Quantum Local Search for Graph Community Detection. In: APS March Meeting Abstracts. APS Meeting Abstracts, vol. 2019, pp. 42–009 (2019)
13. Bravyi, S., Kliesch, A., Koenig, R., Tang, E.: Obstacles to State Preparation and Variational Optimization from Symmetry Protection (2019). arXiv:1910.08980
14. Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y., Balaprakash, P.: Learning to optimize variational quantum circuits to solve combinatorial problems. Proceedings of the AAAI Conference on Artificial Intelligence **34**(03), 2367–2375 (2020). doi:10.1609/aaai.v34i03.5616
15. Lee, X., Saito, Y., Cai, D., Asai, N.: Parameters fixing strategy for quantum approximate optimization algorithm. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 10–16 (2021). doi:10.1109/QCE52317.2021.00016
16. Galda, A., Liu, X., Lykov, D., Alexeev, Y., Safro, I.: Transferability of optimal QAOA parameters between random graphs (2021). arXiv:2106.07531
17. Sauvage, F., Sim, S., Kunitsa, A.A., Simon, W.A., Mauri, M., Perdomo-Ortiz, A.: FLIP: A flexible initializer for arbitrarily-sized parametrized quantum circuits (2021). arXiv:2103.08572
18. Akshay, V., Rabinovich, D., Campos, E., Biamonte, J.: Parameter concentrations in quantum approximate optimization. Phys. Rev. A **104**, 010401 (2021). doi:10.1103/PhysRevA.104.L010401
19. Streif, M., Leib, M.: Training the Quantum Approximate Optimization Algorithm without access to a Quantum Processing Unit (2019). arXiv:1908.08862
20. Crooks, G.E.: Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem (2018). arXiv:1811.08419
21. Kochenberger, G.A., Glover, F.: A unified framework for modeling and solving combinatorial optimization problems: A tutorial. In: Multiscale Optimization Methods and Applications, pp. 101–124. Springer, Boston, MA (2006). doi:10.1007/0-387-29550-X_4. https://doi.org/10.1007/0-387-29550-X_4
22. Hinterberger, H.: Exploratory data analysis. In: Encyclopedia of Database Systems, pp. 1080–1080. Springer, Boston, MA (2009). doi:10.1007/978-0-387-39940-9_1384. https://doi.org/10.1007/978-0-387-39940-9_1384
23. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(86), 2579–2605 (2008)
24. BROYDEN, C.G.: The convergence of a class of double-rank minimization algorithms 1. general considerations. IMA Journal of Applied Mathematics **6**(1), 76–90 (1970). doi:10.1093/imamat/6.1.76
25. Dunning, I., Gupta, S., Silberholz, J.: What works best when? a systematic evaluation of heuristics for max-cut and qubo. INFORMS J. on Computing **30**(3), 608–624 (2018). doi:10.1287/ijoc.2017.0798
26. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph Neural Networks: A Review of Methods and Applications (2018). arXiv:1812.08434
27. Kipf, T.N., Welling, M.: Variational graph auto-encoders. NIPS Workshop on Bayesian Deep Learning (2016)



(a) (b)

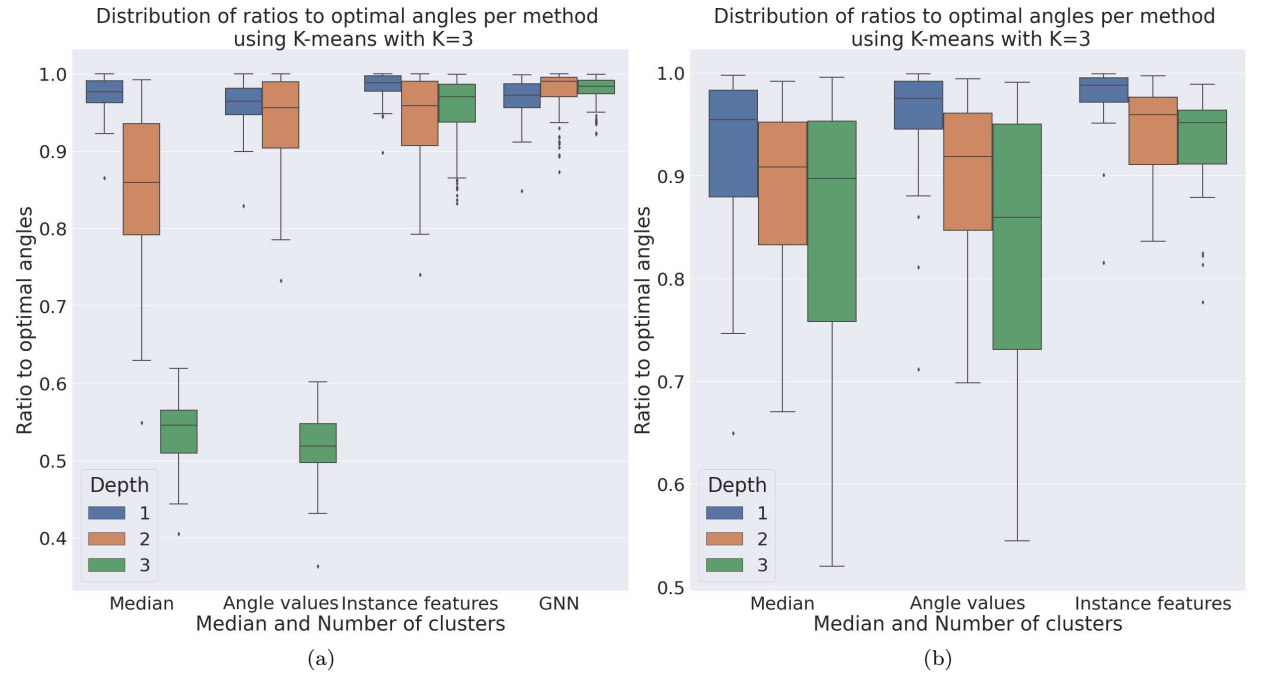
Figure 8: Empirical cumulative distribution functions of ratios to optimal angles' for all depths and number of clusters. A lower curve for an approach means better results when using it aggregating depths and number of clusters. We see for MaxCut (a) and QUBO problems (b) that instance features achieve better results, VGAE being competitive with instance features. When using 3 clusters, using VGAE on MaxCut instance and instance features for QUBOs leads to better ratios.

28. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., Zhang, Z.: Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315 (2019)



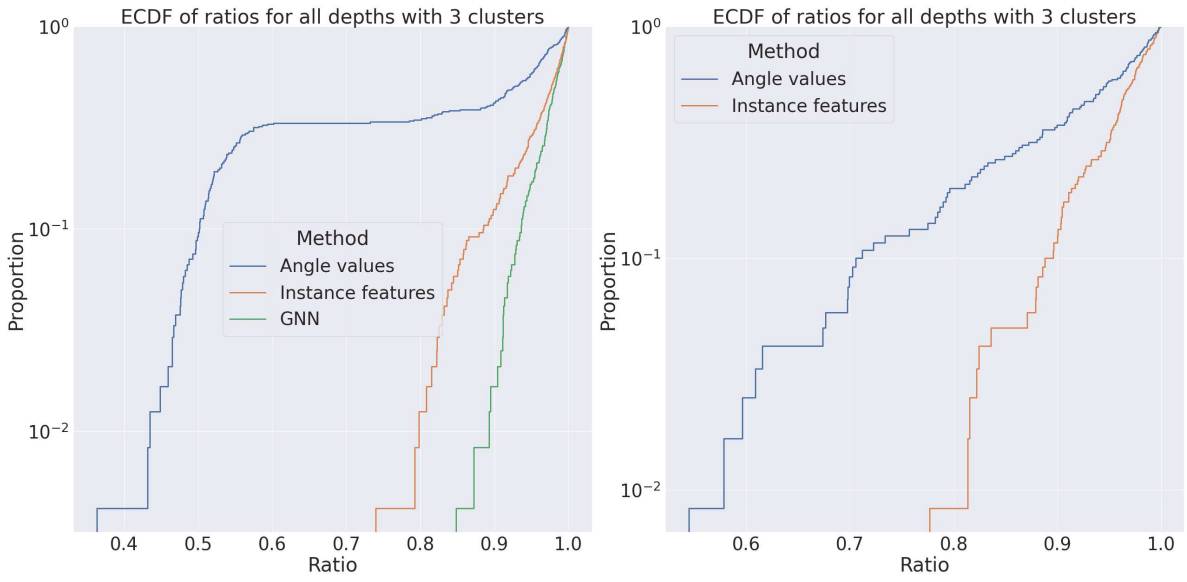
(a) (b)

Figure 9: Empirical cumulative distribution functions of ratios to optimal angles' per method and depth. The lower the curve, the better the method. In most cases, the curve instance features was lower (except for QUBOs (b) at $p = 1$, and VGAE's curve was more competitive at $p = 2$ for MaxCut (a)). This was also the case when using 3 clusters.



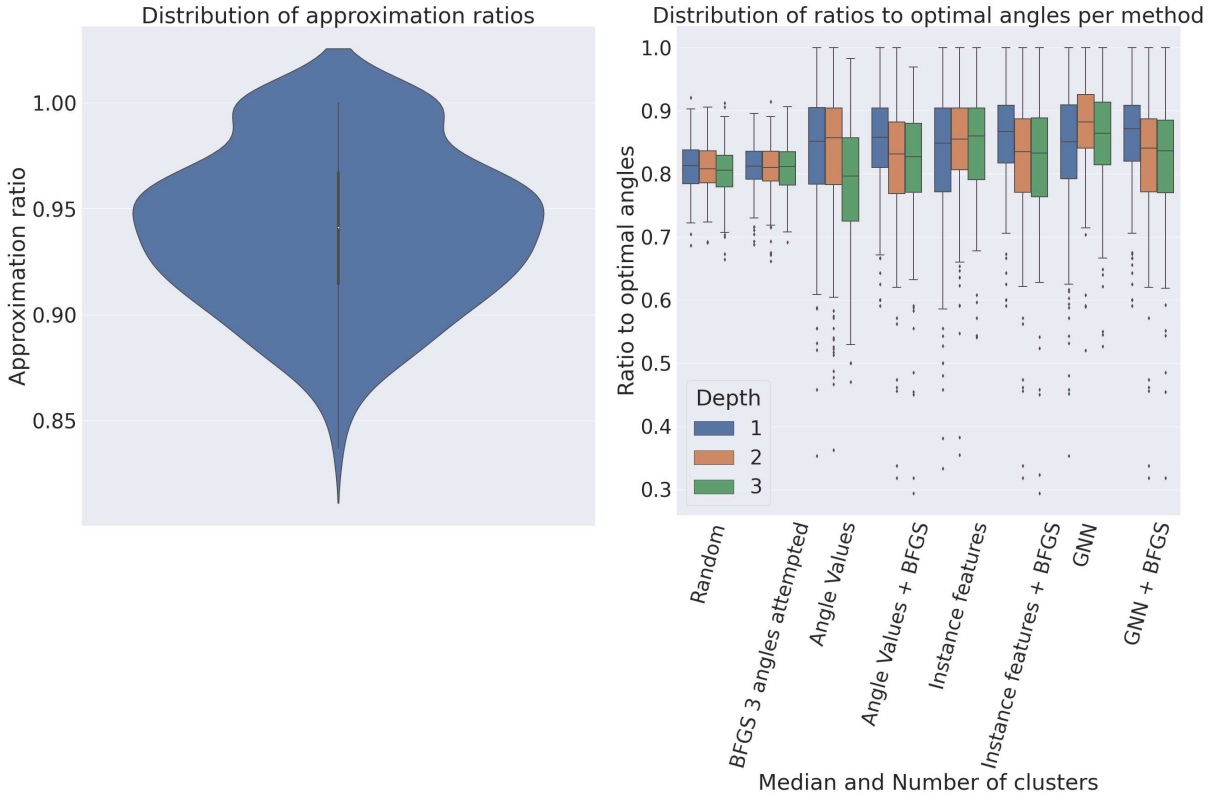
(a) (b)

Figure 10: Boxplot of ratios comparing K-means $K = 3$ using instance features against angle values on MaxCut (a) and QUBOs (b). The ratios are obtained from 40% of the instances with highest number of nodes. From depth-aggregated results, on MaxCut, using the median values gives a median ratio of 0.859316, 0.925564 with angle values, 0.976959 with instance features and 0.981618 using VGAE. On QUBOs, we obtained respectively 0.926136 for the median of angle values, 0.936679 clustering with angle value and 0.963677 with instance features.



(a) (b)

Figure 11: Empirical cumulative distribution functions of ratios to optimal angles'. The ratios are obtained from 40% of the instances with highest number of nodes. Similar results to Fig. 8 and Fig. 9 are obtained.



(a) (b)

Figure 12: The violin plot visualization represent ratios obtained on MaxCut using all three unsupervised approaches, using just 3 circuit calls perrqaoa iteration, without optimizing further with BFGS. A median ratio of 0.94117 was obtained. The boxplots represent the MaxCut ratios obtained using each approach. We added using random angles per iteration as a baseline as well as using BFGS (starting with random angles) with a budget of 3 angle values attempted during optimization, and we witness clustering approaches yielded better results. When dividing the ratios of the methods with the ones obtained by adding BFGS, we obtain a median ratio of 1, meaning we saved many circuit calls for similar results with clustering.