Solving correlation clustering with QAOA and a Rydberg qudit system: a full-stack approach

Weggemans, J.R.; Urech, A.; Rausch, A.; Spreeuw, R.; Boucherie, R.; Schreck, F.; Schoutens, K.; Minář, J.; Speelman, F.

DOI
10.22331/Q-2022-04-13-687

Publication date
2022

Document Version
Final published version

Published in
Quantum

License
CC BY

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
We study the correlation clustering problem using the quantum approximate optimization algorithm (QAOA) and qudits, which constitute a natural platform for such non-binary problems. Specifically, we consider a neutral atom quantum computer and propose a full stack approach for correlation clustering, including Hamiltonian formulation of the algorithm, analysis of its performance, identification of a suitable level structure for \(^{87}\text{Sr}\) and specific gate design. We show the qudit implementation is superior to the qubit encoding as quantified by the gate count. For single layer QAOA, we also prove (conjecture) a lower bound of \(0.6367(0.6699)\) for the approximation ratio on 3-regular graphs. Our numerical studies evaluate the algorithm’s performance by considering complete and Erdős-Rényi graphs of up to 7 vertices and clusters. We find that in all cases the QAOA surpasses the Swamy bound 0.7666 for the approximation ratio for QAOA depths \(p \geq 2\). Finally, by analysing the effect of errors when solving complete graphs we find that their inclusion severely limits the algorithm’s performance.
tions being APX-hard [30], it is natural to ask whether quantum algorithms can do better.

For much of the earlier work, the QAOA blueprint entailed working with qubits: the +1/-1 eigenstates of the Pauli Z-operator are chosen in direct correspondence with the variables of the underlying combinatorial optimization problem to compute the cost function, and the mixer Hamiltonian is chosen as an independent Pauli X applied to each qubit. For instance, the MAX-CUT problem divides a given graph into two sets, which makes the two-qubit ZZ-operator suitable for encoding the objective function. Our situation is quite different as correlation clustering naturally encompasses many more different clusters. While it is always possible in principle to encode higher-dimensional information into multiple qubits, this does significantly complicate the interactions required.

In this paper we investigate a different route, namely encoding the cluster choice into a qudit. Qudits can be realized in a number of physical platforms including photons [12, 22, 70, 72, 80, 87, 107, 142, 152], ions [82, 106, 120], molecules [9, 68, 115], superconducting circuits [44, 58], nuclear magnetic resonance platforms [44, 58] or NV centres [94, 123] and can offer a more resource efficient approach to quantum computing as compared to qubits [144]. Importantly, for the present problem of correlation clustering, they offer a support with the Hilbert space which is native to the studied problem, namely the different qudit states can encode the labelling of nodes into their respective cluster.

In this respect, quantum computers based on trapped neutral atoms interacting via highly excited Rydberg states are of particular interest [109, 112]. These platforms underwent striking developments in recent years with advances such as the creation of arbitrary quantum processor geometries [18, 19, 48] containing hundreds of individual atoms [116, 119] or the efficient implementation of quantum gates [83, 88]. This in principle allows for efficient implementation of QAOA algorithms [38, 153] and led to intense efforts, both academic and industrial [1–4], in neutral atom based quantum computing [67, 93].

Even though QAOA was invented as a NISQ-suitable algorithm, it can be far from trivial to connect the abstract Hamiltonian formulation to a physical system. Problems include how to relate the intended Hamiltonians to manipulations on the system, how to manage the spatial structure and non-uniformity of the pairwise interactions of a system in a lattice, and careful analysis of the types of errors that might occur in these operations.

It has also become progressively clear that an efficient quantum algorithm has to be designed in a way which implicitly takes into account the relevant hardware constraints, a so-called full-stack approach [10]. In this work we describe precisely such a full-stack solution starting from a combinatorial optimization problem (the correlation clustering problem), through a QAOA Hamiltonian formulation, all the way to describing how to control and analyse a Rydberg qudit system. This entails:

- We encode the correlation clustering problem into the QAOA paradigm, specifically tailored for qudit quantum systems. We give several improvements to the vanilla QAOA (see Sec. 2 for the definition), guided by simulations, including experiments with various meta-optimization strategies.

- Next, we go from the abstract Hamiltonian formulation to the operations available for an actual abstract Rydberg system. This entails showing how to drive the Rydberg system in a way that corresponds to the cost and mixer Hamiltonians, where we also have to take care of the spatial aspect of the interactions. Here, we focus on the example of fermionic strontium $^{87}\text{Sr}$, but a similar derivation would apply to any related system.

- We analyse the algorithm performance in the presence of noise corresponding to the ‘random Pauli’ method and link this error model to actual errors in Rydberg qudit systems.

- As an extra theoretical result, we apply the techniques of Wurtz and Love [150] to the case of correlation clustering on 3-regular graphs, showing that for a slightly modified QAOA at $p = 1$ the approximation ratio is at least 0.6367. From numerical results obtained through local optimization, we conjecture that a tighter bound of this approximation ratio would be 0.6699.
Related work  Independently of the current work, other studies have considered applying QAOA to multi-cut versions of MAX-$k$-CUT [54], and on MAX $k$-VERTEX COVER [28, 33], which are problems with related properties. A key difference between correlation clustering and the other studied problems is the explicit presence of positive / negative edges in correlation clustering, and the idea that for correlation clustering, and the idea that for correlation clustering, the number of clusters is not determined yet as part of the input. Additionally, we optimize the creation of the QAOA formulation, ending up with a native qudit implementation, as well as taking a full stack approach: we study all the steps from the problem towards the implementation on a realistic near-term quantum device.

The paper is organized as follows. In Sec. 2 we briefly recap the QAOA. In Sec. 3 we introduce the problem of correlation clustering and its implementation as QAOA. We then describe various strategies to improve the algorithm’s performance in Sec. 4, which we study in Sec. 5. Next, in Sec. 6 we discuss the experimental building blocks of the qudit Rydberg quantum computer. We proceed with the processor design and the associated gate count and comparison to qubits in Sec. 7. Finally, we discuss how the errors affect the algorithm in Sec. 8, and we conclude and discuss open questions in Sec. 9.

2 Recap of QAOA

In this section we briefly review the quantum approximate optimization algorithm (QAOA) [50]. Consider some combinatorial optimization problem with objective function $C : x \to \mathbb{R}$ acting on $n$-bit strings $x \in \{0, 1\}^n$, domain $\mathcal{D} \subseteq \{0, 1\}^n$, and objective

$$\max_{x \in \mathcal{D}} C(x).$$

(1)

In maximization, an approximate optimization algorithm aims to find a string $x'$ that achieves a desired approximation ratio $\alpha$, i.e.

$$\frac{C(x')}{C^*} \geq \alpha,$$

(2)

where $C^* = \max_{x \in \mathcal{D}} C(x)$. In QAOA, such combinatorial optimization problems are encoded into a cost Hamiltonian $H_C$, a mixing Hamiltonian $H_M$ and some initial quantum state $|\psi_0\rangle$. The cost Hamiltonian is diagonal in the computational basis by design, and represents $C$ if its eigenvalues satisfy

$$H_C|x\rangle = C(x)|x\rangle \text{ for all } x \in \{0, 1\}^n. \quad (3)$$

The mixing Hamiltonian $H_M$ depends on $\mathcal{D}$ and its structure [63], and is in the unconstrained case (i.e. when $\mathcal{D} = \{0, 1\}^n$) usually taken to be the transverse field Hamiltonian $H_M = \sum_j X_j$. Constraints (i.e. when $\mathcal{D} \subset \{0, 1\}^n$) can be incorporated directly into the mixing Hamiltonian or are added as a penalty function in the cost Hamiltonian. The initial quantum state $|\psi_0\rangle$ is usually taken as the uniform superposition over all possible states in the domain. QAOA, parametrized in $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_{p-1}), \beta = (\beta_0, \beta_1, \ldots, \beta_{p-1})$, refers to a level-$p$ QAOA circuit that applies $p$ steps of alternating time evolutions of the cost and mixing Hamiltonians on the initial state. At step $k$, the unitaries of the time evolutions are given by

$$U_C(\gamma_k) = e^{-i\gamma_k H_C},$$

$$U_M(\beta_k) = e^{-i\beta_k H_M}. \quad (5)$$

So the final state $|\gamma, \beta\rangle$ of QAOA$_p$ is given by

$$|\gamma, \beta\rangle = \prod_{k=0}^{p-1} U_M(\beta_k)U_C(\gamma_k)|\psi_0\rangle. \quad (6)$$

The expectation value $F_p(\gamma, \beta)$ of the cost Hamiltonian for state $|\gamma, \beta\rangle$ is given by

$$F_p(\gamma, \beta) = \langle \gamma, \beta \vert H_C \vert \gamma, \beta \rangle, \quad (7)$$

and can be statistically estimated by taking samples of $|\gamma, \beta\rangle$. The achieved approximation ratio (in expectation) of QAOA$_p$ is then

$$\alpha = \frac{F_p(\gamma, \beta)}{C^*}. \quad (8)$$

The parameter combinations of $\gamma, \beta$ are usually found through a classical optimization procedure that uses (7) as a black-box function to be maximized. The QAOA framework has thus been described so far, with randomized initial points and without any other improvement strategies, will be referred to as the vanilla QAOA in the rest of this work.

Accepted in Quantum 2022-03-21, click title to verify. Published under CC-BY 4.0.
3 Correlation clustering

Generally, the objective of clustering problems is to group elements into a family of subsets, named clusters, such that the elements within a cluster are more similar to one another than elements in different clusters. In case of the correlation clustering problem, we would like to cluster without specifying the number of clusters in advance based only on pairwise relations. The problem was introduced by Bansal et al. [16] to the theoretical computer science community and has applications amongst others in social psychology, statistical mechanics and biological networks.

Instances of correlation clustering problems are commonly represented as a graph problem, where the nodes are the elements to be grouped in clusters and edge weights represent similarities between these elements. Correlation clustering is then formally defined in the following way: let \( G(V,E) \) be an undirected graph, where \( V,E \), denotes the sets of nodes and edges, respectively. Let \( N \) be the total amount of nodes, i.e., \( N = |V| \). Every edge \( (u,v) \in E \) is labelled either ‘+’ or ‘-’, depending on whether the elements are similar or dissimilar, respectively. This is the unweighted variant of correlation clustering. Additionally, one can also consider the weighted variant: edges \( (u,v) \) carry weights \( w_{(u,v)} \in \mathbb{R}^+ \) describing an additional measurement of similarity or dissimilarity. There are two complementary problems to correlation clustering. MAXAGREE aims to maximize the number of agreements, defined as the number of ‘+’ edges inside clusters plus the number of ‘-’ edges across clusters. In MINDISAGREE one wants to minimize the number of disagreements: the number of ‘+’ edges across different clusters plus the number of ‘-’ edges inside clusters. The decision versions of MAXAGREE and MINDISAGREE are identical and known to be NP-complete [16]. However, they differ in the approximation setting. MAXAGREE on general graphs is APX-hard: to be precise, it has been shown that for every \( \epsilon > 0 \), it is NP-hard to approximate both the weighted [30] and unweighted [136] versions of MAXAGREE within a factor of \( 79/80 + \epsilon \), and the best classical algorithm has approximation ratio \( \alpha = 0.7666 \) [134] via semi-definite programming (SDP) with rounding techniques. In the remainder of the text this specific value of \( \alpha \) will be referred to as the Swamy bound, named after the author of Ref [134]. If the graph is complete and unweighted the problem becomes significantly easier although still NP-hard. For complete graphs, Bansal et al. provided a polynomial time approximation scheme (PTAS) [16]. MINDISAGREE is APX-hard on both complete [16] and general graphs [30], and the best approximation algorithm achieves only an approximation logarithmic in the input size [47]. When the correlation clustering problem is restricted to a fixed number of clusters \( k \), it also remains NP-hard when \( k \geq 2 \), albeit with existing PTASs for both the maximization as well as the minimization variant [60].

Other algorithmic methods that have been proposed to solve correlation clustering include integer linear programming (ILP) and heuristics: examples are greedy methods [6, 100, 125], local search methods [59, 61], and large move making algorithms [14]. In practice, the ILP-approach can solve to about 200 nodes due to a scaling of \( O(N^5) \) in the amount of constraints in its formulation. The SDP relaxation method has better scaling in its constraints \( (O(N^2)) \), with the best SDP solvers handling up to about several thousands of nodes [46]. Heuristics have been shown to be able to handle problems consisting over 100k nodes [14]. In this work, we will focus on unweighted correlation clustering in complete and Erdős–Rényi graphs.

A specific real-world application of the correlation clustering problem can be found in the sub-task of distinguishing between persons within multi-person problem. For example, decomposing different persons is necessary in human-robot collaborations where humans fulfil tasks collaboratively with robots and a robot has to differentiate between the position of different human collaborators. In the approach presented in [71], a Convolutional Neural Network (CNN) computes candidates for the location of different body parts of different persons within a given picture and also pairwise terms how these candidates relate to each other. Leaving out the specific labelling of body parts of each person, the detected body part candidates from the CNN can be represented as nodes and the computed pairwise terms as edge weights of a graph and thus solving the correlation clustering problem of this graph corresponds to a decomposition of the persons in a picture (cf. illustration in Fig. 1).
would require more than performance in practice, even when the optimal solution by a single qudit state of $N$ to a qudit quantum system of $d$ qudits consisting of $d$ levels, such that every node is described by a single qudit state $|c_u\rangle$ meaning that node $u$ is put in (a superposition of) cluster(s) $c$.

From industrial practice we know that the objective function within the correlation clustering problem can be very sensitive to the assignments of parameters and sometimes even a sufficiently good enough local minimum cannot be found within reasonable time by using classical algorithms.

3.1 Hamiltonian formulation for qudit systems

Let $G = (V, E)$ be a graph with $N = |V|$ nodes that specifies the input to some correlation clustering problem instance. Every edge $(u, v) \in E$ has a weight $w_{(u,v)} \in \{-1, +1\}$, representing the ‘+’ and ‘−’ relationships in unweighted correlation clustering. We assume that we have access to a qudit quantum system of $N$ qudits consisting of $d$ levels, such that every node is described by a single qudit state $|c_u\rangle$ meaning that node $u$ is put in (a superposition of) cluster(s) $c$.

As shown for the Swamy algorithm, which always produces a solution that uses at most 6 clusters and is still able to achieve a high expected approximation ratio [134]. Consider for example the all-negative weights graph, which would be an instance that requires the maximal amount of clusters to be used, which equals the number of the graph vertices, $d = N$. In this case, one would still be able to satisfy at least $d - 1$ out of $d$ edges and thus is still able to achieve an approximation ratio larger than $(d - 1)/d$. For the experimental example discussed in Sec. 6, we have $d = 10$ and thus the minimum approximation ratio would be $9/10 = 0.9$.

Figure 1: Overlay of a correlation clustering instance on a real-world picture to illustrate the occurrence of correlation clustering within the problem of decomposing persons in images. The graph contains $N = 7$ nodes with weights $w_{(u,v)} = \pm 1$ while the colours indicate an optimal clustering with objective function value of 10 in the MAXAGREE setting. Note how in general the graph does not need to have a perfect (unfrustrated) clustering, i.e. where all allocated clusters match the ‘+’ and ‘−’ relationships in unweighted correlation clustering, i.e. where all allocated clusters match the colours indicate an edge.

4Here, we would like to stress that even though $d$ might be limited, this will not be too detrimental to the QAOA’s performance in practice, even when the optimal solution would require more than $d$ clusters. This was already shown for the Swamy algorithm, which always produces a solution that uses at most 6 clusters and is still able to achieve a high expected approximation ratio [134].
such that every level is connected to its nearest neighbours, including periodic boundary conditions. The full mixing Hamiltonian is then

\[ H_M = \sum_{u \in V} h_M. \] (14)

We can pick any value of \( r \in \{1, \ldots, d-1\} \), where the special cases at the boundary are called the single-qudit ring mixer for \( r = 1 \) and the fully-connected mixer for \( r = d-1 \). We take the superposition of all qudit computational basis states as our initial state, i.e.

\[ |\psi_0\rangle = \frac{1}{\sqrt{d^N}} \sum_{z \in [d]^N} |z\rangle. \] (15)

Note how the cost Hamiltonian formulation (10) is not equivalent to correlation clustering in the MAXAGREE setting: instead of counting just the agreements we count the number of agreements minus the disagreements. However, since every clustering of two nodes (for which an edge exists), needs to be in either agreement or disagreement with the corresponding weight, the sum of the agreements and disagreements is equal to the number of edges in the unweighted setting. Therefore, for a correlation clustering problem with optimum value \( C^* \) in the MAXAGREE setting the approximation ratio of this QAOA formulation is equal to

\[ \frac{F_p(\gamma, \beta) + |E|}{2C^*}. \] (16)

For the numerical simulations in the algorithmic sections of this work, we implement the initial state by generalized Hadamard operations and assume that the cost and mixing unitaries are elementary operations native to our system. We adopt the Cirq framework [42], since it supports qudit systems, with custom gate operations to match our established formulation.

4 Improvement strategies

We will be interested in performance at low-depth and hence we only consider \( r = 1 \) in Eq. (11) for the simulations in sections 4 and 5. Through numerical evaluation of the vanilla QAOA, we found that the following strategies considerably improved the QAOA’s performance for our problem:

**Choice of the classical optimizer** There is no such thing as a one-size-fits-all classical optimizer that performs well for all QAOA problems: performance varies amongst different problems and can greatly differ for different hyper-parameter settings [81]. We decided to compare different classical optimizers, found in the scikit-quant [81] and SciPy Python packages [5], using their off-the-shelf hyper-parameter settings. In a small study, the best performance was obtained by using BOBYQA. The results of the full comparison can be found in Appendix A.

**Restarts.** Local optimizers can greatly benefit from restarts (i.e. re-running the same algorithmic procedure with the same parameter settings) since they are sensitive to the quality of initial points. But even with fixed initial points, due to stochastic elements in the optimizers’ procedures as well as being introduced from sampling (or gate errors), the algorithms can be made more robust by incorporating restarts. The work of Shaydulin et al. shows how multi-start methods can improve QAOA [121]. In our numerical simulations we set the number of restarts to 5.

**Optimised initial points.** Numerical and analytical works have indicated that QAOA’s parameters \( \gamma, \beta \) are concentrated in parameter space for instances that belong to ‘similar classes’ [8, 24, 130]. In practice, this means that we can select a single instance from a class, work very hard to find optimal parameters through variational optimization or from analytical arguments, and store these values to be used as initial points when solving other instances. For every considered \( N, d \), we use the all-negative-weights instance to obtain the parameter values to be used as initial points. Numerical results of a study at \( p = 1 \) can be found in Appendix B.

**Looping over clusters.** Initial experiments showed that the vanilla QAOA performed well on instances for which the optimal solution required many clusters – this means that the optimal cluster number was close to the available number of clusters \( d \) – but not so when the number of clusters was low. Since our Hamiltonian
The formulation allows for \( d \) to be varied, we can iterate the algorithm over all possible maximum number of clusters, i.e. \( d \in \{1, \ldots, N\} \), and keep the best result obtained over all iterations.

Fig. 2 shows the numerical results of the performance on a data set of 50 correlation clustering instances on complete graphs with weights randomly picked out of \( \{-1, +1\} \). In the creation of the data set, we swept the probability for giving an edge weight ‘+1’ from 0 to 1 uniformly in order to have a good representation of all possible weight configurations.

We observe that looping over cluster numbers has the largest contribution to the improvement, followed by the optimized initial points, which work particularly well for intermediate values of 2 ≤ \( p \) ≤ 4. This could be due to the fact that for \( p = 1 \) the optimizer is able to find good points even with bad initial points, whilst for \( p = 5 \) the distance from the initial point and the optimal point might be larger in parameter space, which results in the initial point becoming effectively random again. The restarts are expected to have less of an impact when used in conjunction with good initial points. Examples of other variants and strategies to QAOA that one could consider, but were not used in this work, are RQAOA [25], ADAPT-QAOA [154].

### 5 Performance

We consider correlation clustering data sets consisting of complete graphs as well as Erdős–Rényi graphs: these are random graphs with a fixed amount of nodes \( N \) but for which the edges are created according to some edge creation probability. In the creation of our Erdős–Rényi data sets we use an edge creation probability of 0.5, and additionally have the criteria that every instance needs to have at least one edge. For both graph types we again sweep the edge weight probability of giving an edge weight ‘+1’ from 0 to 1 to create 50 random instances.

Fig. 3 shows the average values, standard deviation and total range (defined as the worst and best performance over all instances in a data set of graphs of fixed size \( N \)) of the achieved approximations.
Figure 4: Scatter plot of the achieved approximation ratios $\alpha$ on the $N = 7$ complete graph data set as a function of the optimal number of clusters, which was obtained through brute force search. In the case when multiple different cluster numbers were optimal, the data points are plotted for all of these values.

Figure 5: Average (filled circles) and worst (triangles) approximation ratios $\alpha$ as a function of $N$ for $p = 2$ for the complete graph (labelled ‘C’) and Erdős–Rényi (labelled ‘ER’) data sets. The dashed line is again the Swamy bound. The shaded area represents the error in the mean.

...approximation ratios on instances of our data sets as functions of $N$ and QAOA depth $p$, achieved by the QAOA that adopts all strategies as listed in section 4. We observe that the worst performance is slightly better for complete graphs than it is for Erdős–Rényi graphs, which is to be expected as the complete graph problem instances have more structure and are also more easily solved classically [16]. Furthermore, we find a slightly more profound $p$-dependence, as reflected for instance in the amplitude of the standard deviation with increasing $p$, for Erdős–Rényi graphs as compared to the complete graphs. This is not surprising, as for the complete graphs the QAOA at $p = 2$ is already able to ‘see the whole graph’; for every edge any other edge is at most $p$ edges away. The average performance is comparable amongst different graph types. For $p = 1$ we observe that the worst performance on instances from our bench-marking data sets becomes comparable to the Swamy bound of 0.7666 for $N \geq 5$, but for $p = 2$ we have that the algorithm performs better than this bound on all instances in the data set.

In order to investigate instance-dependence of the performance, Fig. 4 shows a scatter plot of the performance on individual instances in the complete graph data set of $N = 7$ for different values of $p$. We find that the algorithm at low $p$ has the most difficulty with instances that require a low number of clusters (except the singleton cluster case, which is trivial for $d = 1$), and as $p$ increases the most difficult instances seem to have optimal cluster numbers in the middle between the singleton and all-different clusters.

Next, Fig. 5 shows the average and worst approximation ratios a function of $N$ giving an indication of the scalability. We observe that the decrease of the worst approximation ratio in $N$ seems to slow down considerably for $N \geq 5$ (for complete graphs it even improves). It should be noted though that it is difficult to make too definitive statements here: we were limited to studying graphs up to $N = 7$ and have no guarantee that we used optimal parameters. However, we can show that the approximation ratio in the limit of large $N$ can in fact become independent of $N$. We propose (conjecture) that in this limit QAOA$_1$, looping over $d = 1, 2, 3, 4$, has a performance guarantee of 0.6367 (0.6699) on 3-regular graphs. The full derivation of this bound can be found in Appendix C, and follows a similar method as the one proposed by Wurtz and Love [150]. However, it is not yet clear how this bound changes as a function of the graph’s degree. For MAXCUT [145] and Maximum Independent Set (MIS) [53] it has already been shown that the approximation ratio at $p = 1$ decreases as the degree of the graph increases. Overall, our results indicate that QAOA, albeit with some added heuristic strategies, might achieve a competitive approximation ratios with respect to the best classical approximation algorithm in solving correlation clustering problems of low-degree...
graphs at low circuit depth.

6 Experimental building blocks

In this section we describe the proposed implementation of the qudits. While most current experimental efforts focus on the use of qubits, neutral atoms are a natural platform for qudits and we specifically focus on the example of fermionic strontium, Sr. The reason is that it possesses a nuclear spin \( I = \frac{9}{2} \) and we specifically focus on the example of fermionic Sr\[40, 127, 128\]. The actual addressing relies on the Zeeman splitting of the \( \mathbb{P} \)-manifold and has been experimentally demonstrated using \( ^{173}\text{Yb} \)[135]. Applying moderate values of the magnetic field \( B \) results in linear Zeeman splitting with an energy shift between the adjacent \( m_F \) states of \( \mu_B g B / h \), where \( \mu_B \) and \( h \) is the Bohr magneton and the Planck constant respectively and \( g \) is the Landé g-factor. For the \( \mathbb{P} \)-manifold, \( g \approx 0.36 \) and \( \mu_B g / h \approx 0.5 \text{MHz/G} \)[23]. This provides a splitting of \( \approx 50 \text{MHz} \) between adjacent \( m_F \) states for a magnetic field amplitude of 100 G, allowing for both resonant and off-resonant addressing as we now discuss.

Rydberg states. The 2-gates are implemented via the Rydberg blockade provided by the interaction energy \( V \), which for the density-density interaction between a pair of atoms separated by a distance \( R \) typically corresponds to a Van der Waals type \( V(R) = C_6 / R^6 \), where \( C_6 \) is so-called “\( C_6 \)” (or Van der Waals) coefficient [27, 56, 109, 112].

The atom can be excited from the \( S \)-manifold through a two-photon transition via the \( \mathbb{P} \)-manifold to a Rydberg state \( |Ry⟩ = |n^3S_1⟩ \) (red and green arrows in Fig 6a,b). In principle one could also use a direct one-photon transition to the \( |n^1P_1⟩ \) Rydberg state (dashed purple line in Fig 6a). However, due to current technology limitations, such as lack of lasers of sufficient power (and the associated optical elements) for the |S⟩ – |n^1P_1⟩ transition wavelength of \( \approx 220 \text{nm} \), we solely focus on the |S⟩ – |P⟩ – |Ry⟩ scheme, cf. Fig. 6b. This allows for \( m_F \) specific addressability of the Rydberg states as well. The two-photon transition is typically operated off-resonantly but given the extremely long lifetime of the \( \mathbb{P} \)-manifold (\( \tau_P = 156 \text{s} \) in the absence of the tweezer light) and to further reduce the timescales needed for operation we consider a resonant two-step process: the chosen \( m_F \) state is transferred as \( |S, m_F⟩ \rightarrow |P, m_F⟩ \rightarrow |Ry⟩ \) using two consecutive \( \pi \)-pulses or using stimulated rapid adiabatic passage. The properties of the relevant manifolds are summarized in Fig. 6b.

In the following, we refer to |S⟩ as the qudit manifold. We note that owing to the long lifetime of the \( |P⟩ \)-manifold, one could in principle

Qudit manifold. The relevant level scheme of Sr is sketched in Fig. 6a. As stated above, the ground state manifold, which we denote with a slight abuse of notation as \( |S⟩ \equiv |1S_0, F = \frac{3}{2}⟩ \), consists of \( d = 2F + 1 = 10 \) \( m_F \)-states, \( m_F \in \{ -9/2, \ldots, 9/2 \} \). We also denote \( |P⟩ \equiv |3P_2, F = \frac{11}{2}⟩ \) the excited state manifold, which we will use to implement the 1- and 2-gates [102] (here we choose the \( 3P_2 \) manifold in particular due to its long lifetime, which allows for a resonant excitation to the Rydberg state, cf. below). The optical tweezers providing the trapping potentials are typically realized with light of wavelength \( \lambda_{\text{tweezer}} \) that is red-detuned from the dominant \( |S⟩ – |1P_1⟩ \) trapping transition (not shown in the Fig. 6a). The choice of the \( \mathbb{P} \)-manifold is motivated by the fact that, unlike the other possible choices such as \( |3P_2, F = \frac{7}{2}⟩ \) or \( |3P_2, F = \frac{9}{2}⟩ \), it possesses a so-called magic wavelength \( \lambda_{\text{tweezer}} \approx 900 \text{nm} \) for which the transition frequencies \( |S, m_F⟩ - |P, m_F⟩ \) are approximately independent of the intensity of the tweezer light for all \( m_F, m_F' \), which ensures a position independent addressing frequency of the individual \( m_F \) states [41, 113, 138]. The actual addressing relies on the Zeeman splitting of the \( \mathbb{P} \)-manifold and has been experimentally demonstrated using \( ^{173}\text{Yb} \)[135]. Applying moderate values of the magnetic field \( B \) results in linear Zeeman splitting with an energy shift between the adjacent \( m_F \) states of \( \mu_B g B / h \), where \( \mu_B \) and \( h \) is the Bohr magneton and the Planck constant respectively and \( g \) is the Landé g-factor. For the \( \mathbb{P} \)-manifold, \( g \approx 0.36 \) and \( \mu_B g / h \approx 0.5 \text{MHz/G} \)[23]. This provides a splitting of \( \approx 50 \text{MHz} \) between adjacent \( m_F \) states for a magnetic field amplitude of 100 G, allowing for both resonant and off-resonant addressing as we now discuss.

In the following, we refer to |S⟩ as the qudit manifold. We note that owing to the long lifetime of the |P⟩-manifold, one could in principle
choose it as the basis for the qudit states and the subsequent analysis can be readily adapted to this alternative choice.

Having identified the suitable level structure we now proceed with the design of the 1- and 2-qudit gates.

### 6.1 1-gates

We consider the implementation of a qudit 1-gate by coupling the qudit level to level by means of laser fields of Rabi frequency . The values of and are taken from [105]. The -manifold states are used to realize the single qudit gates such as the mixing unitaries, which are shown in (c) for and , see text for details.

#### 6.1.1 Implementation of hardware mixers

Let us first discuss the implementation of the mixing unitary , cf. Eqs. (5) and (14). Specifically, we will consider two specific cases, namely and , and we further comment on .

---

$r = 1$. We propose to implement the case as shown in Fig. 6c. Starting with all qudit levels in the manifold, we apply the following sequence of pulses:

1. Apply \( \left\lfloor \frac{d}{2} \right\rfloor \) \( \pi \)-polarized \( \pi \)-pulses on the levels \( \{1, 3, \ldots, l\} \) with \( l = \frac{d}{2} - 1 \), on the \( S \rightarrow P \) transition, which brings them from \( S \) to \( P \).

2. Apply the Rabi frequencies connecting nearest-neighbour qudit levels for time \( \tau_k \).

3. Repeat the step 1. to bring the \( \left\lfloor \frac{d}{2} \right\rfloor \) levels from \( P \) back to \( S \).

---

*We note that analogous approach has been adopted in Ref. [88], where the \( ^3P_0 \) manifold has been used for a realization of a qubit state.*
When the individual Rabi frequencies $\Omega_{\ell,\ell'}$ are adjusted such that $c_{\ell,\ell'=\ell+1} = 1 \forall \ell$, $h_{\Omega} \to h_M$ and the sequence 1.-3. corresponds to the mixing unitary $U_M$, Eq. (5), with the mixing parameter $\beta_k = \Omega_{\ell_k}$.

One important remark is in order: the mixing Hamiltonian (13) implements a “periodic boundary condition” in that it couples the levels 0 and $d - 1$ (and similarly for higher $r$). Such coupling is typically not native to a physical qudit, which instead corresponds to “open boundary condition” as is apparent from (17). We discuss the difference between using (13) or (17) below, cf. Sec. 6.1.2 and Fig. 7.

$r = 2$. For $r = 2$ we consider an off-resonant coupling exploiting a detuning $\Delta$ from the $\mathbb{P}$-states as shown in Fig. 6c. This allows to extend the structure of the coupling Hamiltonian Eq. (17) to include also a second diagonal. The coupling Hamiltonian (17) becomes

$$\tilde{\Omega} h_{\Omega} = \tilde{\Omega} \begin{pmatrix} 0 & \tilde{c}_{0,1} & \tilde{c}_{0,2} & \cdots & \tilde{c}_{0,d-1} \\ \tilde{c}_{0,1} & 0 & \cdots & \cdots & \cdots \\ \tilde{c}_{0,2} & \cdots & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

(18)

Here the tilde denotes the effective Rabi frequency which in the far-detuned limit is given by $\tilde{\Omega}_{\ell,\ell'} = \Omega_{\ell,\ell'} / |\Delta_{\ell}|$ where $|\ell\rangle$ is the detuned state from the $\mathbb{P}$-manifold to which $|\ell\rangle$ and $|\ell'\rangle$ are coupled. As for $r = 1$, here the Rabi frequencies $\tilde{\Omega}_{\ell,\ell'}$ have to be adjusted such that $\tilde{c}_{\ell,\ell+1} = \tilde{c}_{\ell,\ell+2} = 1 \forall \ell$.

$r > 2$. Going beyond $r = 2$ becomes non-trivial due to the connectivity of the coupling Hamiltonian (18), here limited to next-to-nearest qudit levels. A general strategy, exploiting the decomposition of an arbitrary single qudit unitary into a sequence of parallelized Givens rotations under the finite connectivity constraint and invoking a greedy optimization algorithm has been described in [101]. As in this work we are concerned only with $r \leq 2$ we do not analyse this situation further.

6.1.2 Performance of hardware mixers

To investigate how the hardware-specific mixers given by Eqs. (17) and (18) compare to the mixer (13) we run simulations similar to the ones performed in Sec. 5. To this end we consider the $N = 4$ complete graph data set. This choice is motivated by the fact that for $N = 3$ (i.e. $d = 3$), Eq. (18) becomes (13) and as $N$ increases (and $d = N$ for complete graphs considered here), the two boundary states $|0\rangle, |d - 1\rangle$ that have different mixing constitute an increasingly small fraction over all states. It is thus plausible to assume that the difference between the mixers Eqs. (13),(17) and (18) is most relevant for smallest $N > 3$, i.e. $N = 4$.

All non-zero elements in (17) and (18) are set to unity, and we again generate initial points starting from the all-negative-weights graph. The numerical results in Fig. 7 show that the performance is very similar amongst the different mixers – the largest observed percentage performance difference for a single instance is about 4%. From the data we can conclude that the hardware mixer with $r = 2$ performs better on average than the normal mixer with $r = 1$, which outperforms the hardware mixer with $r = 1$. Overall, we can conclude from this data that replacing the standard mixer (13) with the hardware-specific mixers (17) and (18) is not expected to result in a significant decrease in performance.

6.1.3 Unitaries beyond mixers

Next, we specify single qudit unitaries beyond mixers, which we will exploit in the construction of the 2-gates in Sec. 6.2 below. For a two-level system with levels $|\ell\rangle, |\ell'\rangle$ driven on resonance with Rabi frequency $\Omega_{\ell,\ell'}$, the unitary evolution operator expressed in the $\{|\ell\rangle, |\ell'\rangle\}$ basis reads...
Figure 7: Results for the different mixing Hamiltonians given by Equations (13), (17) and (18), labelled as ‘N: r = 1’, ‘HW: r = 1’ and ‘HW: r = 2’, respectively. A dot is the average approximation ratio over all 50 instances and the shaded area represents one standard deviation. The results are obtained using a complete graph data set in the same way as described in Sec. 5.

(cf. Appendix D)

\[ U_{\ell,\ell'}^{2\text{-level}}(\theta, \varphi) = \begin{pmatrix} \cos \frac{\theta}{2} & -ie^{i\varphi} \sin \frac{\theta}{2} \\ -ie^{-i\varphi} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \]

(19)

where \( \theta = 2t |\Omega_{\ell,\ell'}| \), \( t \) is the duration of the laser pulse and \( \Omega_{\ell,\ell'} = |\Omega_{\ell,\ell'}| e^{i\varphi} \).

Similarly, we can write down a general unitary evolution operator for a three-level system (so-called \( \Lambda \)-scheme). As we will be particularly interested in performing controlled operations through driving the qudit states to a Rydberg level, we shall consider a \( \Lambda \)-scheme where levels \( |\ell\rangle, |\ell'\rangle \) are coupled to a common Rydberg state \( |r\rangle \). We will further require that at the end of the operation, there is no population in the Rydberg state. In this case the unitary for resonant driving and expressed in the \( \{|\ell\rangle, |\ell'\rangle, |r\rangle\} \) basis reads (cf. Appendix D)

\[ U_{\ell,\ell'}^{3\text{-level}}(\theta, \varphi) = \begin{pmatrix} \cos \frac{\theta}{2} & e^{i\varphi} \sin \frac{\theta}{2} & 0 \\ -e^{-i\varphi} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \]

(20)

and \( \Omega = |\Omega_0| e^{i\alpha}, \alpha = 0, 1 \) such that \( \varphi = \phi_0 - \phi_1 \). Here, we have denoted \( \Omega_0 = \Omega_{\ell,r}, \Omega_1 = \Omega'_{\ell',r} \) for simplicity. The unitary takes the form (20) for pulses of duration \( t = \pi(2m - 1)/\Omega, m \in \mathbb{N} \). In what follows we refer to the unitaries (19) and (20) simply as \( U \) and we shall specify which one is considered where appropriate. For future convenience we denote the usual Pauli X as

\[ X_{\ell,\ell'} = U_{\ell,\ell'}(\pi, 0) \]

(22)

and the phase gate (defined up to a global phase)

\[ P_{\ell,\ell'}(\varphi) = U_{\ell,\ell'}(\pi, \varphi) U_{\ell,\ell'}(\pi, \varphi_1) = \begin{pmatrix} e^{i\varphi} & 0 \\ 0 & e^{-i\varphi} \end{pmatrix} \]

(23)

where \( \varphi = \varphi_2 - \varphi_1 \).

6.2 2-gates

\( \text{CP gate} \). We proceed with the construction of the cost unitary \( U_C \), Eq. (4). Noting that the the cost Hamiltonian (10) is given by a sum of commuting operators acting on the graph edges, we consider an action of the cost unitary on a single pair of qudits. It corresponds to a controlled-phase gate of the form

\[ \text{CP}(\gamma)|\ell\rangle |\ell\rangle = \left[ \delta_{\ell,\ell'} e^{-i\gamma} + (1 - \delta_{\ell,\ell'}) \right] |\ell\rangle |\ell\rangle \]

(24)

It is defined up to a global phase and \( \delta_{\ell,\ell'} \) is the Kronecker delta \(^4\).

The key element in the construction of the CP gate Eq. (24) is a controlled-phase unitary \( U \) acting on a single level \( |\ell\rangle \) from the qudit manifold by coupling it to the corresponding Rydberg state \( |r\rangle \), namely

\[ U_{\ell}^{1\text{-level}}(\gamma) \equiv \begin{pmatrix} P_{\ell,\ell'}(\gamma) \\ 1 \end{pmatrix} \]

(25)

where the first and second qudit is the target and control respectively (labelled as t and c) and P

\(^4\)We note that according to the definitions Eqs. (4), (9) and (10), the global phase omitted in the Eq. (24) equals \( \mp \gamma_k \) for the edge weight \( w_{(u,v)} = \pm 1 \) such that the relative phase \( \gamma \) between the two-qudit states \( |\ell\rangle |\ell\rangle \) equals \( |\ell\rangle |\ell\rangle \). It is given by \( \gamma = \pm 2\gamma_k \). We also remark that while we consider the graph edges weights \( w_{(u,v)} \in \{-1, 1\} \), one could implement arbitrary weights through edge-specific phases \( \gamma \rightarrow \gamma_{(u,v)} = 2w_{(u,v)} \gamma_k \).
is given in Eq. (23). Here, we have introduced the notation $U_q(q_t/q_c)$, where $q_t, q_c$ label the target and control qudits, respectively. We can then construct the controlled phase gate Eq. (24) either in a manifestly symmetric way

$$\text{CP}(\gamma) = \prod_{\ell=0}^{d-1} U_{\ell}^{(12)} U_{\ell}^{(21)},$$

(26)

or alternatively as

$$\text{CP}(\gamma) = \left[ \prod_{\ell=0}^{d-2} \text{CX}^{(12)}_{\ell,\ell+1|\ell} \right]^{\dagger} \prod_{\ell=0}^{d-2} \text{CX}^{(12)}_{\ell,\ell+1|\ell} \prod_{\ell=0}^{d-2} \text{CX}^{(12)}_{\ell,\ell+1|\ell},$$

(27)

cf. Appendix D for derivation. The gate (27) has the advantage of reducing the cost of the gate as counted by the number of hard-core operations (laser pulses), cf. Eq. (29) and Sec. 7. Here, $P_{\text{aux},0}^{(1)}$ is the phase gate (23) applied to the first (target) qudit and driving the level $|0\rangle$ through an auxiliary state, which here is not a Rydberg state. Similarly, we have introduced the controlled-X gates $\text{CX}^{(q_t|q_c)}_{\ell_t,\ell_t'|\ell_c}$, where $q_c$ denotes the control, $q_t$ the target, $\ell_c$ the control qudit level and $\ell_t, \ell_t'$ the pair of target levels being acted upon. Importantly, here the target levels $\ell_t, \ell_t'$ are swapped when the control qubit is not in the state $\ell_c$ which is highlighted using the negation sign in the subscript, $\ell_c$. We denote that for a typical hardware implementation, the choice of $\ell_t, \ell_t'$ is not arbitrary but upper bounded. In the present case we shall assume $|\ell_t - \ell_t'| \leq 2$. The controlled-X are implemented in the standard fashion as

$$\text{CX}^{(12)}_{\ell_t,\ell_t'|\ell_c} = \begin{array}{c}
\begin{array}{c}
\text{X}_{\ell_t,\ell_t'}
\end{array}
\end{array},$$

(28)

where the X-gate acting on the target is driven through a Rydberg level (a $\Lambda$-scheme) as described by the Eq. (20). We also remark that the CP gate Eq. (27) is invariant under the exchange of the control and target qudits, i.e. $(12) \rightarrow (21)$. In Sec. 7 we will be concerned with assessing the cost of the algorithm as counted by the number of elementary 2-gate operations. To this end we introduce the notation $[G]$ to denote the cost of a gate $G$ as counted by the number of the CX gates (28) or its equivalents in the decomposition of $G$. The corresponding cost of the CP gates is thus

$$\text{Eq. (26)} : [\text{CP}] = 2d$$

(29a)

$$\text{Eq. (27)} : [\text{CP}] = 2d - 2$$

(29b)

\textbf{SWAP gate.} In order to perform the CP gate on a pair of distant graph vertices, it is in general necessary to bring them together by means of a swap gate SWAP$_d$. Several possible implementations of the SWAP gate for qudits have been proposed [15, 57]. Here we shall consider the implementation of Ref. [57], which parallels the qubit SWAP construction consisting of three consecutive CX gates: \text{SWAP}_2 = \text{CX}^{(q_t|q_t)} \text{CX}^{(q_c|q_t)} \text{CX}^{(q_t|q_c)}$

where we have dropped the level indices taking $|1\rangle$ to be the control level for the qubit as customary. The qudit version of the SWAP is given by [57]

$$\text{SWAP}_d = \text{CX}^{(q_t|q_t)} \text{CX}^{(q_c|q_t)} \text{CX}^{(q_t|q_c)},$$

(30)

which is the direct generalization of the SWAP for qubits. The qudit controlled-X $\text{CX}_d$ is defined as

$$\text{CX}_d|\ell\rangle|\tilde{\ell}\rangle = |\ell\rangle|\bar{\ell} - \ell\rangle,$$

(31)

and mod $d$ is understood in evaluating $\bar{\ell} - \ell$ in the last expression. It can be implemented as

$$\text{CX}_d^{(q_t|q_t)} = \text{QFT}_d^{(q_t)} \text{CZ}_d^{(q_t|q_t)} \text{QFT}_d^{(q_t)},$$

(32)

where

$$\text{QFT}_d^{(\ell)} = \frac{1}{\sqrt{d}} \sum_{\ell' = 0}^{d-1} e^{i \frac{2\pi}{d} \ell \ell'} |\ell'\rangle,$$

(33)

is the quantum Fourier transform in the single qudit space (i.e. a unitary operation with matrix elements $(\text{QFT}_d)^{\ell,\ell'} = \exp[i(2\pi/d)\ell \ell']/\sqrt{d}$) and $\text{CZ}_d$ is defined as

$$\text{CZ}_d|\ell\rangle|\bar{\ell}\rangle = e^{i \frac{2\pi}{d} \ell \bar{\ell}} |\ell\rangle|\bar{\ell}\rangle.$$

(34)

We note that the QFT$_d$ gate is a 1-gate and can thus be synthesized using methods of [101], similarly to the mixing unitary for $r > 2$, cf. also [131] for implementation of QFT$_d$ in the context of

\textbf{For a qubit, the cost of the symmetric gate Eq. (26) can be further reduced to $[\text{CP}] = d$, cf. Appendix D}
multilevel atoms. To quantify the cost of the SWAP gate, we thus focus on the CZ$_d$ gate (34), which can be implemented as

$$\text{CZ}_d^{|q\rangle q\rangle} = \prod_{\ell=1}^{d-1} X_{\ell\ell_r}^{(q)} \prod_{\ell=1}^{d-1} P_{\ell\ell}^{(q)} \left( \frac{2\pi}{d} \ell \right) X_{\ell\ell_r}^{(q)} (35)$$

which contains $(d-1)^2$ applications of the phase gate P. Importantly, we note that the product in the brackets can be executed in parallel by simultaneous application of the laser pulses connecting each level $|\ell\rangle$, $\ell = 1, \ldots, d-1$, of the target to its respective Rydberg level $|r\rangle$. This is precisely an example of the parallelization offered by the qudit hardware. We thus get, in conjunction with Eq. (30), for the total cost of the qudit SWAP

$$[\text{SWAP}_d] = 3(d-1). \quad (36)$$

6.3 Initialisation and measurement

**Initialization.** The initial state (15) can be prepared by initializing all atoms in the $|0\rangle \equiv |\mathbb{S}, m_F = -9/2\rangle$ state by standard means of optical pumping and then applying a sequence of $d-1$ unitaries (19) such that

$$|+\rangle = \prod_{\ell=1}^{d-1} U_{\ell-1,\ell}^{\dagger} \left( \theta_{\ell}, \varphi = \frac{\pi}{2} \right) |0\rangle, \quad (37)$$

where $\cos(\theta_{\ell}/2) = 1/\sqrt{d-(\ell-1)}$. Here the unitary can be implemented either via resonant or off-resonant Raman coupling described in Sec. 6.1 for the range $r = 1$ and $r = 2$ of the mixing unitaries respectively.

**Measurement.** In order to projectively measure the quantum state after each iteration of the QAOA we consider imaging the atoms using resonance fluorescence by collecting the light scattered from the strongly driven $|0\rangle \rightarrow |0\rangle P_1, F' = 11/2, m_F = -11/2$ transition, cf. Fig 6a. This has the advantage of simultaneously cooling the atoms in the $|0\rangle$ state while imaging them [139], also [37, 89, 114] for related techniques. Exploiting the state-specific detection of individual $m_F$ states, first only the population in the $|0\rangle$ state ($m_F = -9/2$) is being detected. In the case of negative outcome (no population in the $|0\rangle$ state), the population from the adjacent $m_F$ state, i.e. from $|1\rangle$ ($m_F = -7/2$), is transferred to $|0\rangle$ by using optical pumping, or coherently via stimulated Raman adiabatic passage and the $|0\rangle$ is imaged again. This process is repeated until the positive detection of some qudit level $|\ell\rangle$. This allows for imaging of all the qudit states within the expected lifetime in optical tweezers of $\gtrsim 10$ seconds (as there is no active cooling of the $|1\rangle, \ldots, |d-1\rangle$ states during the imaging). This time is limited mainly by off-resonant scattering of the tweezer light and also by background gas collisions, where the latter can be further reduced by increasing the quality of the vacuum.

7 Processor design and gate count

Here we seek to evaluate the cost of the algorithm, cf. Sec. 2 and Sec. 3, as quantified by the gate count. As discussed in Appendix F, the dominant errors stem mainly from the 2-gates and we thus focus on the 2-gate count. To this end we consider the total count $C_{\text{tot}}$ of qubit primitive 2-gates and specifically we will use the qubit controlled-X [CX] as our cost unit (This choice of counting will be useful when comparing the qudit vs. qubit encodings in Sec. 7.1). $C_{\text{tot}}$ is determined by (i) the topology of the graph encoding the clustering problem, (ii) the topology of the quantum processor and, for qubits, (iii) the encoding scheme, which we discuss in Sec. 7.1.

In this section we will find that taking into account the hardware considerations (i)-(iii) yields the expected result, namely that the qudit encoding is superior to the qubit one in that it yields smaller $C_{\text{tot}}$. Readers not interested in the details of the gate count can consult the summary in the Table 2.

To proceed, let us first comment on 2-gates beyond nearest neighbours. The neutral atom and ion based systems possess a native long-range interaction, which typically decays as a power law $1/R^\alpha$ with distance ($\alpha = 6$ for a pair of Rydberg atoms interacting through a Van der Waals potential). Such potential gives rise to clusters of higher connectivity on the processor, which can lead to an improvement in performance over processors with only nearest-neighbour interactions [84]. The related scaling properties of the quantum gates for trapped ensembles rather than single neutral atoms have been analysed theoretically in [110] and such ensembles occupying hundreds of sites have been realized recently experimentally [143]. Furthermore, claims have been
made that up to 50 atoms can be connected without the need of a SWAP operation [2]. In the here considered implementation using Rydberg blockade and Van der Waals interaction, the price to pay for the higher connectivity is the longer duration of the gate. Taking the basic building block Eq. (25) of the CP gate as an example, the gate duration is mainly given by the duration of the P-gate applied to the target qudit. This is because while the X-gates acting on the control qudit can be performed in principle arbitrarily fast limited only by the available Rabi frequency, the Rabi frequency used to realize the P-gate has to satisfy the blockade constraint $V(R)/\Omega \gg 1$ for a given atom distance $R$. Since $V(R) \propto 1/R^6$, for the same quality of blockade (same $V(R)/\Omega$) the gate time thus scales as $\propto 1/\Omega \propto R^6$. For this reason and for the sake of clarity, in the following we consider only nearest-neighbour 2-gates and leave the algorithm hardware optimization exploiting longer-range connectivity for future work.

Another remark is that it is desirable to parallelize the gate operations, cf. e.g. [83] for recent realization with Rydberg atoms, to reduce the absolute time of the algorithm run. This in principle allows one to reduce the effect of noise such as the background gas collisions or off-resonant scattering from the optical traps, cf. Sec. 8. Such parallelization however does not change the 2-gate count and we don’t elaborate on it further.
Table 1: Left column: Considered processor geometries in 1D (upper two rows) and 2D (lower four rows) for both the qudits and the binary encoding. In 2D the shaded blue and red areas highlight the effective qudit (e-dit) in the binary encoding including the considered enumeration of the qubits. Right column: The construction of the CP gates for binary encoding. Construction for both $q = \log_2 d$ and $q \neq \log_2 d$ case is shown (based on Ref. [54]) together with the decomposition of the multi-controlled $C^{q-1}(U)$ gate for $q = 4$ and $q = 3$ (based on Ref. [17]). The unitaries $V$ satisfy $V^2 = U$ and $V^4 = U$ in the decomposition of $C^2(U)$ and $C^3(U)$ respectively [17].
| Geometry | Encoding | n_{\text{SWAP}}^{\text{intra}} | n_{\text{SWAP}}^{\text{inter}} | [\text{SWAP}] = q^2[\text{SWAP}_2] | n_{\text{SWAP}}^{\text{inter}}[\text{SWAP}_2] = 3q(q-1)[\text{SWAP}_2] | | [S^{i-1}(U)] | C_{\text{tot}} |
|----------|---------|-----------------|-----------------|-----------------|-------------------------------|---|---|
| 1D       | q = 2   | 3 x 4           | 3 x 6           | 1[\text{CX}] = 1 | $|E| \times 35 - O(N) \times 12$ | | |
|          | q = 3   | 3 x 9           | 3 x 18          | 5[\text{CX}] + 2[\text{SWAP}_2] = 11 | $|E| \times 98 - O(N) \times 27$ | | |
|          | q = 4   | 3 x 16          | 3 x 36          | 13[\text{CX}] + 2[\text{SWAP}_2] = 49 | $|E| \times 213 - O(N) \times 48$ | | |
| 2D       | O(N)    | $3 x 32^*$      | $3 x 4^*$       | 1[\text{CX}] = 1 | $|E| \times 17^* + O(N) \times 10^*$ | | |
|          |         | $3 x 8^*$       | $3 x 6^*$       | 5[\text{CX}] = 5 | $|E| \times 29^* + O(N) \times 24^*$ | | |
|          |         | $3 x 91^*$      | $3 x 124^*$     | 13[\text{CX}] + 2[\text{SWAP}_2] = 19 | $|E| \times 65^* + O(N) \times 28^*$ | | |

Table 2: Gate count as per Eqs. (39) for qudits and qubit binary encoding for 1D (blue shaded lines) and 2D (red shaded lines) processor geometries. The cost of the CP gate is evaluated using the circuits from Ref. [54] as well as the standard decomposition of multi-controlled qubit gates shown in the right column of Table 1. The cost of the SWAP gate is evaluated according to the qubit arrangements shown in the left column of Table 1. In 2D, it is obtained as a weighted average over the neighbours (for $q = 2, 3, 4$, each e-dit has four, three and two neighbours to which it is connected by one leg and two, three and four neighbours to which it is connected by three legs). This might result in a fractional value of [SWAP] and gate counts stemming from such weighted average are denoted by a star. The total gate count $C_{\text{tot}}$ of the algorithm is indicated in the last column in purple. For the qubit encoding, $C_{\text{tot}}$ also includes the contribution $2q[\text{CX}]$ from (40) not listed in the Table. For complete graphs considered here, the dominant contribution to $C_{\text{tot}}$ is coming from the number of edges $|E| = N(N-1)/2 = O(N^2)$. This is highlighted in bold for qudits in the last column. It is apparent from $C_{\text{tot}}$ that the qudit encoding is superior to the (best-case scenario $q = \log_2 d$) binary one in all cases.

### 7.1 Gate count and comparison to qubits

As stated in the introduction, the use of qudits in general offers a resource-efficient alternative to qubit encodings and for certain problems, such as correlation clustering, also a natural physical platform for their implementation. To proceed with the comparison between qudits and qubits, we first specify the qubit encoding. In what follows, we denote by tilde a qudit-like gate acting on the effective qudit encoded in qudits. We also term such an effective qudit an e-dit to distinguish it from the physical qudit.

Effective controlled-phase gates $\overline{\text{CP}}$ equivalent to (24) have been proposed in [54]. The Ref. [54] studied the MAX-$k$-cut using QAOA and considered two possible encodings, a one-hot and a binary one. The one-hot encoding seems to produce smaller 2-gate count when $q \neq \log_2 d$ and larger when $q = \log_2 d$, the precise numbers depending on the graph topology [54]. Here

$$q = \lceil \log_2 d \rceil,$$

is the number of qubits necessary to contain the qudit Hilbert space as subspace such that the two become equal when $q = \log_2 d$. On the other hand, the one-hot encoding is more resource extensive than the binary encoding and we therefore choose the binary one. In the scheme of Ref. [54] $q = \log_2 d$ and $q \neq \log_2 d$ correspond to two different realizations of CP, which we list in Table 1 for the reader’s convenience.

Next, we have to specify the processor topology. The versatility of neutral atom platforms allows in principle for arbitrary arrangements of the atoms, which can be exploited for efficient encoding of the graph instance at hand. For specificity, we perform the gate count on one of the examples of main interest, the complete graph, for which $|E| = N(N-1)/2$. Motivated by ongoing experiments [18, 19, 48, 83, 88, 116, 119], we consider a simple 1D chain (with open boundaries) and a 2D regular lattice. In 2D we consider a triangular lattice, which provides the densest
packing of atoms. The 2-gate count is given by

\[ C_{\text{tot}} = n_{\text{SWAP}}^{\text{inter}}[\text{SWAP}] + |E| |\text{CP}| \]  
(39a)

\[ C_{\text{tot}} = n_{\text{SWAP},d}^{\text{inter}}[\text{SWAP},d] + |E| |\text{CP}|, \]
(39b)

where \( n_{\text{SWAP}} \) and \( n_{\text{SWAP},d} \) count the number of SWAPs between the e-dits and qudits respectively such that each vertex has been a neighbour of every other vertex at least once. In principle we can perform the gate count for any \( d \). We note, that due to the \( \text{CP} \) gate structure for \( q \neq \log_2 d \), the total count is relatively much higher than for \( q = \log_2 d \) (Table 1 of [54] gives \( |\text{CP}| = 2, 70, 206, 14, 12, 78, 14 \) for \( d = 2, 3, 4, 5, 6, 7, 8 \), where the values with underline correspond to \( q = \log_2 d \)). By contrast, since the gate structure for qudits is the same for all \( d \) and since our main goal here is to compare the gate count for qudits and qubits, we focus only on the \( q = \log_2 d \) case. The reason for this is that for a given ratio \( C_{\text{qubits}} / C_{\text{qudits}} \) of gate counts for qudits and qubits for \( q = \log_2 d \), this ratio will be only smaller when \( q \neq \log_2 d \). In this case, the cost of \( \text{CP} \) can be further decomposed as

\[ |\text{CP}| = n_{\text{SWAP}}^{\text{intra}}[\text{SWAP}] + 2q(CX) + |C^{q-1}(U)|, \]
(40)

where \( n_{\text{SWAP}}^{\text{intra}} \) counts the number of SWAPs within the e-dit and \( C^{q-1}(U) \) is the multicontrolled unitary performed within the target e-dit, cf. Table 1. The unitary \( U \) in \( C^{q-1}(U) \) is a single qubit gate and its particular form is not relevant for the counting (cf. Ref. [54] for details).

1D. Let us first describe the situation in 1D. Here, \( n_{\text{SWAP}}^{\text{inter}} = n_{\text{SWAP},d}^{\text{inter}} \) indicates the number of SWAPs between qudits (or e-dits) such that each qudit (or e-dit) has been a neighbour of all the others. Here we shall use a rather natural choice of SWAP sequences described in [13, 78]. It consists of a repeated application of two layers of SWAPs, one performing SWAP operations on qudit (e-dit) pairs \( 1-2, 3-4, \ldots \) followed by the other one on pairs \( 2-3, 4-5, \ldots \). This yields \( n_{\text{SWAP}}^{\text{inter}} = n_{\text{SWAP},d}^{\text{inter}} = (N-1)(N-2)/2 \), see also Appendix E. Furthermore we also have \( |E| = N(N-1)/2 \) such that Eq. (39b) together with Eq. (29b) and Eq. (36) yield

\[ C_{\text{tot}} = \frac{(5N - 6)(N - 1)}{2} (d-1) = |E| \times 5(d-1) - O(N), \]
(41)

To analyse the qubit case, let us start with the analysis of the \( C^{q-1}(U) \) gate. Ref. [85] describes a systematic construction of \( C^q(U) \) for arbitrary \( q \) proposing two schemes for such a construction, one scaling exponentially and the other one polynomially with \( q \). While the polynomial scaling is clearly favourable for large \( q \), the actual gate count favours the exponential scheme for \( q \leq 4 \) considered here. We thus consider the decomposition for \( C^2(U) \) and \( C^3(U) \) shown in Table 1, proposed in the early works on quantum information [17, 122]. The considered ordering within the e-dit shown in Table 1 leads to the following counts as function of \( q \)

\[ [\text{SWAP}] = q^2[\text{SWAP}_2] \]
(42a)\n
\[ n_{\text{SWAP}}^{\text{intra}} = 3q(q - 1). \]
(42b)

The total gate count for qudits and e-dits and their break down as per (39) is summarized in Table 2.
geometries. The summary of the costs for 2D is given in Table 2.

As a result, we find the expected outcome, namely that in all considered cases the gate count is lower for qudit encoding than for qubit binary encoding, even for the best-case scenario $q = \log_2 d$ of the latter.

Here we have evaluated the gate count considering the realization of the SWAP gates via the Rydberg interactions. A remark is that the gate count of a sequence of consecutive SWAPs can be further reduced by considering so-called bridge gates, leading however only to a modest improvement by a factor $\approx 1.5$ [73]. In the context of neutral atoms in optical tweezers, it would be interesting to exploit a strength of these platforms and perform the SWAP by physically exchanging the atoms, which for distances $\sim 5 \mu m$ can be done on the timescale of $\sim 50 \mu s$ [117].

8 Errors

In this section we consider an error model used in Ref. [11] in the theoretical analysis of a Rydberg quantum computer and we discuss the implications of the errors for the algorithm performance. Importantly, the error model of Ref. [11] can be cast in the unitary evolution framework used in our work. We comment on the actual experimental errors and how they relate to the considered error model in Appendix F.

**Unitary error model.** Let us consider a set of $d^2$ single qudit unitaries $U \equiv \{ (\Sigma^X)^r(\Sigma^Z)^s \}$, where $r, s = 0, \ldots, d - 1$. Here $\Sigma^X = \Sigma^x + (\Sigma^x)^\dagger$, $\Sigma^x$ is given by Eq. (12) and

$$
\Sigma^Z = \begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & \lambda & & \\
\vdots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & \lambda^{d-1}
\end{bmatrix} \quad \quad \quad (43)
$$

with $\lambda = \exp(2\pi i/d)$ is the generalized Pauli Z. For a qubit, $d = 2$, this reduces to $U = \{ \mathbb{1}, \Sigma^X, \Sigma^Z, \Sigma^X \Sigma^Z \}$. Motivated by the experimental considerations, namely the fact that the errors are dominated by the 2-gate ones, cf. Appendix F, the model consists of applying an identity with probability $1 - p_2$ or a unitary $U \in \mathbb{U}^{\otimes 2} \setminus \{ \mathbb{1} \}$, with probability $p_2/((\mathbb{U}^{\otimes 2} \setminus \{ \mathbb{1} \})) = p_2/(|\mathbb{U}^{\otimes 2}| - 1)$ on each pair of qudits after each cost unitary $U_C = e^{-iH_C}$, cf. Eq. (10). Put formally, for $\rho = U_C \rho' U_C^\dagger$

$$
\rho \rightarrow \rho \quad \text{with prob.} \quad 1 - p_2 \\
\rho \rightarrow U \rho U^\dagger, \quad U \in \mathbb{U}^{\otimes 2} \setminus \{ \mathbb{1} \}, \quad \text{with prob.} \quad \frac{p_2}{|\mathbb{U}^{\otimes 2} \setminus \{ \mathbb{1} \}|} \quad \quad \quad (44)
$$

We consider the same data sets for complete graphs as have been used in Section 5 for $N \in \{3, 4, 5\}$, and use the optimized values for $\gamma, \beta$ and $d$ obtained in the noise-free setting. This way makes it possible to discard the classical optimization loop, saving computation time, and allows us to focus on the performance degradation as a result of the randomization of the state vector. It has been argued in Ref. [151] that noise generated by dephasing, bit flip, and depolarizing channels tends to flatten (on average) the parameter space energy landscape without changing its structure. Since the error model (44) is a qudit generalization of these types of channels, we expect the $\gamma, \beta$ obtained in the noise-free setting to be optimal also in the noisy setting.

The results are shown in Figure 8. For $p_2$ small ($\lesssim 10^{-3}$), we see that the performance is hardly affected by the noise. Once $p_2$ increases, we enter the regime where performance quickly degrades until we reach the performance of the completely randomized state. Whilst the performance at $p_2 = 1$ is considerably smaller than in the noise-free setting, the approximation ratio achieved on average is still relatively large. This is due to the fact that $d$ is already pre-determined: instances with $d = 1$ are not affected by the noise and still maintain a large approximation ratio. We define the threshold noise $p_{2, \text{Th}}$ (threshold amount of 2-gates $g_{\text{Th}}$) as the noise level (amount of 2-gates) for which the QAOA has lost half of its performance as compared to random guessing, on average for all instances.

By determining the values of $p_{2, \text{Th}}$ from the data in Figures 8a-c, and using that the amount of two-qubit gates is simply $pN(N - 1)/2$ (with $p$ the QAOA depth), we plot $g_{\text{Th}}$ as a function of $p_2$ in Figure 8d. We observe that our data are compatible with a linear dependence for $g_{\text{Th}}$ of the form

$$
g_{\text{Th}} = \frac{\kappa}{p_2}, \quad \kappa = 0.84. \quad \quad \quad (45)
$$

This naive model for the noise shows scaling similar to that of Ref. [129] – here the authors showed that a bound on the circuit depth scales inversely
proportional to the quantum gate error – but now in the amount of operations instead of circuit depth.

It is interesting to interpret the results of Fig. 8d in terms of the achievable system sizes and required hardware gate operations. Denoting the error probability of an elementary CX-like hardware gate as $p_{\text{CX}}$ we get for the total success probability (no error) after application of $|E| = g$ gates on the $|E|$ edges

$$p_{\text{success}} = (1 - p_2)^{|E|} = (1 - [\text{CP}]p_{\text{CX}})^{|E|}(1 - [\text{SWAP}]p_{\text{CX}})^{O(N)} \approx (1 - [\text{CP}]p_{\text{CX}})^{|E|},$$

where in the last relation we used that $p_2, p_{\text{CX}} \ll 1$, $|E| \gg N$ while $[\text{CP}] \approx [\text{SWAP}]$. This allows us to identify $p_2 \approx [\text{CP}]p_{\text{CX}}$. Using the result of Table 2, considering $d = 8$ for specificity, and using $|E| = N(N - 1)/2$, we find that for $p_2 = (10^{-2}, 10^{-3}, 10^{-4})$, $p_{\text{CX}} \approx (10^{-3}, 10^{-4}, 10^{-5})$ resulting in $N \approx 13, 41, 130$.\footnote{A technical remark is that strictly speaking the formula (45) was verified when the number of clusters $d$ saturates the number of vertices $N$. Keeping this in mind, the discussed achievable system sizes should thus be considered merely as an estimate.}

We thus see that the inclusion of the errors strongly limits the scaling of the algorithm to large problem instances. We note that similar limitations of the QAOA due to experimental errors have been discussed recently in [64] and [129]. In particular, it has been argued in Refs. [64, 129] that hardware-native graph instances, in case of Ref. [64] a simple square lattice with nearest-neighbour interactions, are less prone to errors and potentially allow to scale up to problem sizes large enough to achieve quantum advantage. This is due to significant simplification of the quantum circuit, which avoids a number of extra compilation steps (such as SWAP gates). The major downside is however that a quantum computer hardware is, typically, not application or instance specific.

In this respect, the neutral atom based platforms seem to be particularly interesting as they allow for implementing graph topologies beyond simple planar ones by exploiting the atom arrangements in three dimensions and the long-range interactions. Nevertheless, as is apparent from the results in this section, the robustness to

---

**Figure 8:** (a)-(c) Numerical results for the QAOA with a error channel given by (44) for $N = 3, 4, 5$, respectively. The plots are the average approximation ratios $\alpha$ over all instances in the complete graph data sets and the shaded areas correspond to one standard deviation. The filled circles data points correspond to the threshold noise for which the QAOA has lost half of its performance over random guessing. (d) Linear fit for $g_{\text{th}} \propto 1/p_2$, individual colours correspond to the respective data points where $p_2 = p_{2,\text{th}}$ in (a), (b) and (c).

---

**9 Conclusions and outlook**

In this work we have addressed the problem of solving the correlation clustering using QAOA and a qudit quantum computer. We have specifically considered a neutral atom based architecture, which has the potential to offer up to $\sim 1000$ qudits in the near future [93, 116, 119]. Here the gates are realized through the interaction of atoms in a highly excited electronic state, a so-called Rydberg state. Considering specifically the element $^{87}\text{Sr}$ we have identified a suitable level structure for the qudit, which in turn allowed us to design the gates for implementing the QAOA on the quantum processor. It is worth emphasizing that while we have considered correlation clustering, the discussed qudit gates can
be used for the closely related MAX-k-CUT and MAX k-VERTEX COVER problems, cf. [54] and [28, 33, 143] respectively.

We assessed the algorithm’s performance by numerical simulations including various optimization strategies, namely restarts, optimizing the initial points and looping over the cluster number. Focusing specifically on complete and Erdős-Rényi graphs of up to 7 vertices and clusters we found that in all studied cases the QAOA with depth \( p \geq 2 \) provides approximation ratios above the Swamy bound 0.7666 [134], corresponding to the best known classical strategy (based on SDP) with performance guarantees for MAXAGREE. Modifying and adopting recently developed classical simulation methods [91] might allow us to study larger graph instances, which we leave to future work.

While this result is encouraging, the inclusion of errors suggests that it is challenging for the QAOA to outperform classical algorithms, cf. Sec. 3, at least for complete graphs on near-term noisy quantum devices. This is in agreement with related results reported recently in Ref. [129] and Ref. [64], where various graph instances were considered to compare the performance of the QAOA using a superconducting quantum chip, including the effect of the errors.

In this respect, the neutral atom based platforms seem to be particularly interesting and our work indicate possible directions in the design of experiments for benchmarking the QAOA on correlation clustering instances. Not only do neutral atoms allow to assemble arbitrary structures in both 2D and 3D [18, 19, 48, 117] (a PTAS does exist for planar graphs [79], but one could still investigate the rate of convergence), but they also allow for native long-range interactions, known to yield an advantage over quantum processors featuring only nearest-neighbour ones [84]. Such long-range interactions in turn allow for implementation of non-planar graphs [64] as native geometry using even a planar arrangement of atoms and at the same time avoid the need for additional SWAP gates. One should however keep in mind that for complete graphs, the errors strongly limit the scaling even when this graph topology is native to the hardware, cf. Sec. 8. This suggests, together with the results of Refs. [64, 129], that the robustness of (low depth) QAOA increases with the decrease of the graph degree. Additional improvement in the performance might be also achieved when using multi-qudit gates, which can further reduce the circuit depth [76]. In this context, it would be also interesting to consider the (native) realizations of unit disk graph instances [103, 104]. Interestingly, and to the best of our knowledge, the NP-hardness of the correlation clustering problem on unit disk graphs remains an open question [43, 69].

It would be highly interesting to address systematically the above listed scenarios, which we leave for future work. It should be also emphasized that while achieving a practical quantum advantage using the QAOA remains a challenge, constructing a qudit quantum processor is a task worth pursuing - it constitutes an exquisite tool for applications beyond the QAOA, allowing for instance for the realization of a plethora of condensed matter models such as the \( d \)-state Potts and other SU(\(d\)) spin systems [20, 31, 34–36, 39, 45, 77, 95–98, 108, 124, 148, 149].

Acknowledgements

We thank Alexander Brinkman for his supervision of the master’s project [147] which led to this publication and Koen Groenland for useful discussions. This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative and supported by the Dutch Ministry of Economic Affairs and Climate Policy (EZK), as part of the Quantum Delta NL programme and by the Netherlands Organization for Scientific Research (NWO) under the Gravitation grant No. 024.003.037 and the Quantum Software Consortium. AR acknowledges the support of the German Federal Ministry of Education and Research in the funding program “quantum technologies – from basic research to market” (contract number 13N15585).

Author contributions. JW and FSp have performed the analysis and numerical simulations of the algorithm. AR has supported the analysis of the numerical simulations. RB encouraged JW to investigate performance guarantees and supervised the findings of this work. JM and KS have designed the gates and with AU, RS and FSc have proposed and analysed the hardware implementation. All authors have contributed to the writing of the manuscript.
References


[37] Jacob P. Covey, Iyavol. S. Madjarov, Alexandre Cooper, and Manuel Endres.


[54] Franz G Fuchs, Herman Oie Kolden, Niels Henrik Aase, and Giorgio Sartor. Efficient encoding of the weighted max k-cut


[69] Poolad Imany, Jose A. Jaramillo-Villegas, Ogaga D. Odele, Kyunghun Han, Daniel E. Leaird, Joseph M. Lukens, Pavel Lougovski, Minghao Qi, and Andrew M. Weiner. 50-ghz-spaced comb of high-dimensional frequency-bin entangled photons from an on-chip silicon nitride microresonator. Op-
tics Express, 26(2), 1 2018. DOI: 10.1364/OE.26.001825.


[84] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. Proceed-


[117] Kai-Niklas Schymik, Vincent Lienhard, Daniel Barredo, Pascal Scholl, Hannah Williams, Antoine Browaeys, and Thierry...


[135] Shintaro Taie, Yosuke Takasu, Seiji Sug-


[152] Jun-ichi Yoshikawa, Marcel Bergmann, Peter van Loock, Maria Fuwa, Masanori Okada, Kan Takase, Takeshi Toyama, Kenzo Makino, Shuntaro Takeda, and Akira Furusawa. Heralded creation of...


A Results optimizer study

We compare the performance of five different optimizers with off-the-shelf hyper-parameter settings using the vanilla QAOA: ImFil, SnobFit and BOBYQA taken from the scikit-quant package [81] and an implementation of COBYLA, used both as single optimizer and in conjunction with basin-hopping (BH), both from the SciPy optimization package [5]. We consider a single instance out of our correlation clustering data sets: the complete graph with $N = 4$ and all edge weights ‘−’ such that the optimal solution corresponds to all nodes being put in different clusters. For each $p \in \{1, \ldots, 5\}$, we generate 25 random initial points in the respective parameter space. The maximum number of iterations for each optimizer is set to 500. Since basin-hopping has two different budgets, one for the basin-hopping steps and one for its local optimizer, we set the maximum number of iterations by using the number of evaluations the local optimizer (COBYLA) used in its individual run: the number of basin-hopping steps is the rounded ratio of the budget over this number. The results for state vector sampling (1000 samples) are given in Figure 9.

![Figure 9: State vector sampling with 1000 samples. Left: found approximation ratios $\alpha$ for 25 random initial points using different optimizers. Right: total number of function evaluations ‘nfev’ as a function of $p$. The shaded area indicates the error in the mean, where the discrete points have been connected in order to improve the readability of the figure. ‘Best found’ indicates the best approximation ratio that was observed over all instances and optimizers.](image)

We find that BOBYQA outperforms all other optimizers in terms of the achieved approximation ratios, but does always use up the maximum available amount of iterations. Unfortunately, the scikit-quant optimization package does not allow us to change the tolerance, which would allow for fairer comparison. As far as the ratio of performance to number of function evaluations is concerned, ImFil performs well. Even though adopting BOBYQA already potentially results in a large performance increase compared to using COBYLA, which was used to obtain our initial results, we still observe that a large performance difference exists between the best value found and the average performance over different optimizer runs. In addition, note that the standard deviation is relatively large as we plot the error in the mean in Fig. 9. This means that there is still a lot to be gained in the classical optimization step, the most natural being the use of good initial points, followed by hyper-parameter optimization. Good initial points will have a larger effect on the local optimization methods (in particular COBYLA
and BOBYQA) compared to the global optimizers (e.g. IMFil). The concentration of initial points at $p = 1$ is studied in the next appendix.

B Initial points study for $p = 1$

In this part of the appendix we give numerical evidence that supports the conclusions of the work by Brandão et al. [24]: initial points for different instances, belonging to similar problem classes, are concentrated. To investigate this effect at $p = 1$, we run the following: we start with initial points we obtained from the all-negative-weights graphs, for some fixed $N$ and $d$, and then solve for all 50 instances in our correlation clustering data set, using COBYLA as the optimizer. The resulting optimal points are plotted in Fig. 10: Note how all points are in the neighbourhood of our initial points. In fact, the smallest rectangular area containing all points (indicated by the black rectangle) takes about 0.2% of the entire possible parameter space. The plot on the right zooms in on this rectangle, showing how the optimal points are located relative to each other. In this plot $x_0^*$ is the initial point, which is for both $N = 4$ and $N = 5$ positioned at the boundary of the collection of points. This makes sense due to the structure of the graph it belongs to—the all-negative-weights graph is itself an extreme case of the correlation clustering problem (requiring all clusters to be used). This also indicates that other graphs might be more suitable to generate initial points.

For the same problem instance, Figure 11 shows the location of the points obtained for different $N$ and $d$ in parameter space ($p = 1$). We observe that $d = 2$ is somewhat of an outsider, but again all points fall in a rectangular area encompassing about 0.2% of the entire possible parameter space.

C Approximation ratio bound on 3-regular graphs

A recent work by Wurtz and Love [150] shows a derivation of lower bounds for QAOA depths $p = 1$ and $p = 2$ (and a conjectured result at $p = 3$) on MAXCUT. In this appendix, we apply their techniques to the correlation clustering problem on 3-regular graphs: graphs for which every node has a fixed degree of 3. The goal is to find a lower bound for the approximation ratio $\alpha$ at $p = 1$ on 3-regular graphs, defined as

$$\alpha = \max_{d \in \{1, \ldots, d_{\text{max}}\}} \frac{F_d(\gamma, \beta)}{C_{\text{max}}},$$

(46)
where $F_d(\gamma, \beta)$ is the expectation value of the state produced by the QAOA algorithm using $d$ levels and $C_{\text{max}}$ denotes the optimal objective function value for a single correlation clustering instance. Since only graphs with degree 3 are considered, we need at most 4 clusters. Therefore, we only consider $d \in \{1, 2, 3, 4\}$. Based on local optimization, we will conjecture that for 3-regular graphs $G = (V, E)$ initial points $\gamma_d^*, \beta_d^*$ exist such that QAOA that loops over the clusters has an approximation ratio larger than 0.6699, even without the classical optimization loop. By relaxing the problem into a linear program (LP) we can prove that this bound is at least 0.6367.

C.1 Problem setup and a lower bound for the energy

Consider an arbitrary 3-regular graph $G$ of $N(G)$ nodes that is not the complete graph with $N = 4$. We identify for each edge $\langle i, j \rangle$ the sub-graph $G_{p,i,j}$ induced by all neighbouring edges at most $p$ steps away from $\langle i, j \rangle$. At $p = 1$ there are only three possible kinds of sub-graph structures as indicated in Fig. 12.

Since all sub-graph types have 5 edges, we have a total of $3 \cdot 2^5 = 96$ possible sub-graphs when we only consider weights $w_{u,v} \in \{-1, +1\}$. However, by symmetry arguments we can reduce the total number of weighted sub-graphs we have to consider: if reordering the edge labels results in the same graph under any rotation, it will look the same to the QAOA. We define three sets of sub-graphs $g_i, i \in \{1, 2, 3\}$, representing all 3-regular sub-graph structures with 6, 5 and 4 nodes respectively (see Figure 12), such that for every sub-graph $\lambda \in g_i$ there exists no other graph $\lambda' \in g_i$ that is equivalent.

Our derivation does not include the possibility that $G$ is the complete $N = 4$ graph, which is not neccessary as it cannot be extended.
in the QAOA setting. Our total set of possible sub-graphs is then $S = g_1 \cup g_2 \cup g_3$. We now decompose our graph $G$ into sub-graph environments $\lambda \in S$ for which the multiplicity of each $\lambda$ is $N(\lambda)$. Since we have such a sub-graph environment for every edge, we must have that the sum over all $N(\lambda)$ is equal to the total number of edges $3N(G)/2$. For a single edge $(u, v)$ with sub-graph environment $\lambda$, denoted as $(u, v) \rightarrow \lambda$, the contribution to the expectation value is given by

$$f_d, (u, v) \rightarrow \lambda \langle \beta_d, \gamma_d \rangle = \langle \beta_d, \gamma_d \mid w_{(u, v)} V_d \mid \beta_d, \gamma_d \rangle.$$ (47)

Note that Eq. (47) only contains terms that operate within $\lambda$. The expectation value of the algorithm with a maximum of $k$ clusters at $p = 1$ for some $\beta_d, \gamma_d$ is given by

$$F_d(\beta_d, \gamma_d) = \sum_{\lambda} N(\lambda) f_d, (u, v) \rightarrow \lambda \langle \beta_d, \gamma_d \rangle.$$ (48)

We must also have that this is always smaller than or equal to the expectation value using the best values of $\beta_d, \gamma_d$

$$F_{d, \text{max}} = \max_{\beta_d, \gamma_d} F_d(\beta_d, \gamma_d) \geq \sum_{\lambda} N(\lambda) f_d, (u, v) \rightarrow \lambda \langle \beta_d, \gamma_d \rangle.$$ (49)

providing us with a lower bound on the expectation value of the algorithm $F_{d, \text{max}}$ at a given $d$.

C.2 Upper bound on the number of agreements

We are now faced with the task of finding an upper bound on $C_{\text{max}}$. A naive bound would be the total number of edges, but we can do better by considering the same argument Wurtz and Love used to determine an upper bound for MAXCUT [150]. Consider the graph $G$, which is a collection of $N(\lambda)$ disconnected sub-graphs $G^\rho_{(u, v)}$ for each edge in $G$. Since the largest sub-graph has only $6^4$ feasible solutions, a brute-force method can be used to find the ratio between the optimal objective function value and the number of edges for every sub-graph $\lambda$, which we will call $c_\lambda$. Since all sub-graphs are isolated (i.e. not connected to each other), the global fraction of agreements to edges is equal to the average fraction over each sub-graph. However, we have several edges belonging to different disjoint sub-graphs in $G$ that are actually the same edge in $G$. In this case, we can have that for both sub-graphs a clustering exists for which the edge contributes to the objective value, but that the required clustering is different in both sub-graphs. As a result, we have that the objective function value of $G$ is bounded from above by

$$C_{\text{max}} \leq \sum_{\lambda} N(\lambda) c_\lambda.$$ (50)

C.3 Constructing the hardest graph

Since we have a lower bound for the algorithm’s expectation value and an upper bound for the optimal objective function value, the approximation ratio $\alpha$ is bounded from below by

$$\alpha(G) \geq \max_d \frac{\sum_{\lambda} N(\lambda) f_{d, \lambda} \langle \gamma_d, \beta_d \rangle}{\sum_{\lambda} N(\lambda) c_\lambda}.$$ (51)

The worst case approximation ratio is then given by the hardest graph $G = G^*$, which corresponds to some combinations of $N(\lambda) \in \mathbb{Z}_0$ for all sub-graphs $\lambda$. However, not all combinations of $N(\lambda)$ correspond to a valid graph, as was already noted by Farhi et al. [50]. For now, we will only consider the structure of the graph and not take the feasibility of certain weight combinations into account. First, we note that for every edge in sub-graph $g_3$ there must be at least 4 other edges that have the environment corresponding to sub-graph $g_2$. Also, we have that the ‘triangle’ of $g_2$ and ‘crossed square’ of $g_3$ cannot share the same vertex, which means that the number of triangular edges and crossed square edges must be smaller than the number of nodes $N(G)$. Our final constraints are that all $N(\lambda)$ are non-negative integers and must sum up to $3N(G)/2$. We define $n(\lambda) \equiv N(\lambda)/N(G)$.
such that we can relax the integer constraint in the limit of large $N(G)$, and obtain the following 

**Minimax Linear Fractional Program:**

$$\min_{n_\lambda(G): \lambda \in S} \max_d \frac{\sum \lambda n_\lambda(G)f_{d,\lambda}(\gamma_d, \beta_d)}{\sum \lambda n_\lambda(G)c_\lambda}$$

s.t. 

$$\sum_{n_\lambda(G): \lambda \in g_2} n_\lambda(G) - 4 \sum_{n_\lambda(G): \lambda \in g_3} n_\lambda(G) \geq 0$$

$$\sum_{n_\lambda(G): \lambda \in g_2} n_\lambda(G) \geq -1$$

$$\sum_{n_\lambda(G): \lambda \in g_2} n_\lambda(G) \geq -1$$

$$\sum_{n_\lambda(G): \lambda \in S} n_\lambda(G) = \frac{3}{2}$$

$$n_\lambda(G) \geq 0 \text{ for all } \lambda \in S$$

Equation (52) is in the form of a generalized fractional program, which is not reducible to a linear program (LP) which can be solved efficiently. However, there exist linear relaxation bounding techniques that do allow for global optimization up to some error $\epsilon$. We have performed initial experiments with one of those techniques [75], but were not able to achieve desirable results so far due to the difficulties in approximating our objective function. However, we can construct two related LP formulations that upper and lower bound the value of $\alpha$. Let us define the feasible region $C$ such that $x \in C$ when it satisfies the constraints of (52).

1. Take the number of edges as the upper bound instead of the fractional objectives (50), i.e. we let $c_\lambda \rightarrow 1$ for all $\lambda$. Under this relaxation we can write (52) as the following LP:

$$\min \alpha$$

s.t. 

$$\sum_{\lambda} n_\lambda(G)f_{d,\lambda}(\gamma_d, \beta_d) \leq \frac{3}{2} \alpha \text{ for all } d$$

$$n_\lambda(G) \in C, \alpha \in \mathbb{R}$$

The solution of (53) is a lower bound to the actual bound, as sub-graphs that originally had $c_\lambda = 0.8$ now contribute too much to the upper bound of the optimal objective function value (50).

2. Similarly, we can also assume that only sub-graphs for which a perfect clustering exists contribute to the construction of the most difficult graph. Define the set $S' = \{\lambda | \lambda \in S, c_\lambda = 1\}$ to be such a set. Since $S' \subset S$, the resulting solution provides an upper bound to the best value of $\alpha$ that we can find for $\lambda \in S'$.

$$\min \alpha$$

s.t. 

$$\sum_{\lambda} n_\lambda(G)f_{d,\lambda}(\gamma_d, \beta_d) \leq \frac{3}{2} \alpha \text{ for all } d$$

$$n_\lambda(G) \in C, \alpha \in \mathbb{R}, \lambda \in S'$$

All LPs will be solved by CVXOPT contained in the package lpsolvers [29] for Python.

C.4 Iterative procedure for determining the bound

The minimax optimization problem gives us a method to determine the hardest instance $G^*$, given that we fix the parameters $\gamma = (\gamma_1, \gamma_2, \gamma_3, \gamma_4), \beta = (\beta_1, \beta_2, \beta_3, \beta_4)$. But how do we choose the values of these parameters? As one would normally do with QAOA, a classical optimization loop can be adopted. First, we choose some initial values for $\gamma, \beta$ and calculate $f_{d,\lambda}(\gamma_d, \beta_d)$ for all sub-graph environments $\lambda$ and all $d \in \{1, 2, 3, 4\}$. Next, we construct the hardest graph $G^*$ by solving (52), (53)
or (54). Instead of minimizing over \( n_\lambda \) whilst keeping \( \gamma, \beta \) fixed, in the next step we now fix \( n_\lambda \) and try to maximize the objective over \( \gamma, \beta \), i.e., we want to

\[
\max_{\gamma_d, \beta_d} \frac{\sum_\lambda n_\lambda(G) f_{d, \lambda}(\beta_d, \gamma_d)}{\sum_\lambda n_\lambda(G) c_\lambda} \tag{55}
\]
s.t. \( \gamma_d, \beta_d \in [0, 2\pi) \),

for all \( d \in \{1, 2, 3, 4\} \). For these new values of \( \gamma^*, \beta^* \), there might exist some other graph \( G^{**} \) that is more difficult than the one we originally obtained. After we obtained \( G^{**} \) we can again try to find new parameters \( \beta^{**}, \gamma^{**} \). This suggests the use of an iterative procedure for establishing the bound. We do not know whether this procedure converges, but it doesn’t necessarily have to: any combination of \( \gamma, \beta \) has its own lower bound that holds specifically for these parameters, and can therefore be used as our result. Convergence would only suggest something about the tightness of this bound, guaranteed global convergence would mean that no better values of \( \gamma, \beta \) exist. In our iterative procedure, the best parameter combinations we found are listed in Table 3:

<table>
<thead>
<tr>
<th>( \gamma_d^* )</th>
<th>( \beta_d^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>2.857</td>
</tr>
<tr>
<td>-</td>
<td>2.773</td>
</tr>
<tr>
<td>-</td>
<td>2.682</td>
</tr>
<tr>
<td>-</td>
<td>0.4833</td>
</tr>
<tr>
<td>-</td>
<td>0.1310</td>
</tr>
<tr>
<td>-</td>
<td>0.1435</td>
</tr>
</tbody>
</table>

Table 3: Parameter values for different \( d \) at which we were able to obtain the results of Theorem C.1 and Conjecture C.1. At \( d = 1 \) the algorithm has only one state and hence no parameters.

For the parameter combinations in Table 3, (53) can be solved to an arbitrary precision, which establishes the proof of the following theorem:

**Theorem C.1.** For 3-regular graphs \( G = (V, E) \), where \( G \) is not the complete \( N = 4 \) graph, initial points \( \gamma_d^*, \beta_d^* \) exist such that at \( p = 1 \) the QAOA that loops over the clusters gives a 0.6367-approximation algorithm, even without the classical optimization loop.

However, as stated before, this bound is too strict as the objective function of \( G \) is overestimated. Using COBYLA as a local optimizer, we numerically observed that solving (52) never resulted in a bound lower than 0.6699. This is equal to the upper bound we find by solving (54) (also 0.6699). Since COBYLA does not guarantee a global minimum and (54) only considers a subset of all possible graphs, we can only conjecture that this is the actual lower bound:

**Conjecture C.1.** For 3-regular graphs \( G = (V, E) \), where \( G \) is not the complete \( N = 4 \) graph, initial points \( \gamma_d^*, \beta_d^* \) exist such that at \( p = 1 \) the QAOA that loops over the clusters gives a 0.6699-approximation algorithm, even without the classical optimization loop.

For this particular combination of sub-graphs belong to the hardest graph \( G^* \), the classical optimization step actually does not make much of a difference: our results show that solving (55) with \( G = G^* \) results in

\[
\max_{\gamma_d, \beta_d} \max_d \frac{\sum_\lambda n_\lambda(G^*) f_{d, \lambda}(\gamma_d, \beta_d)}{\sum_\lambda n_\lambda(G^*) c_\lambda} \approx 0.674,
\]

which is only a small improvement on the conjectured bound. This shows the quality of these values of \( \gamma^*, \beta^* \) (as well as the hardness of the graph \( G^* \)). Additionally, this also provides further indication that QAOA, when having access to good initial points, can also be used without the classical optimization step [24, 130].

### C.5 Performance bounds for \( p > 1 \)

Unfortunately, we do not observe evidence for the existence of a trivial graph hierarchy as Wurtz and Love proved (and conjectured) for MAXCUT at \( p \leq 2 \) \((p > 2) \) [150]. In fact, when we
consider their large loop conjecture [150] for our problem, we find that for our used $\alpha, \beta$ we obtain a worst-case approximation ratio of $\alpha = 0.693$, which is significantly larger than the bounds in Theorem C.1 and Conjecture C.1. Therefore, we do not conjecture the structure of the most difficult graph at any $p$, which would ease the determination of lower bounds at larger $p$. If we are to use the same method as we used for $p = 1$, we will have to consider of the order of $123 \cdot 2^{13} \approx 10^6$ different sub-graphs (not taking symmetries into account). We can reduce this number by exploiting symmetries, but since determining the energy of the largest sub-graph (14 nodes) is computationally very expensive we do not attempt to determine bounds for $p > 1$. In fact, a back-of-the-envelope estimation to the amount of computing hours needed to execute such a computation – again not utilizing symmetries – shows that we would need on the order of 10 million computing hours.

### D Details of the experimental building blocks

#### D.1 Dynamics of a driven three-level system

Here we briefly review the standard derivation of the unitaries Eqs. (19),(20). Let us consider the level scheme depicted in Fig. 13. Two qudit states $|\ell\rangle, |\ell'\rangle$ of energies $\omega_\ell, \omega_{\ell'}$ are coupled with lasers of frequencies $\omega_{L,\ell}, \omega_{L,\ell'}$ and Rabi frequencies $\Omega_{\ell,r}, \Omega_{\ell',r}$ to a Rydberg state $|r\rangle$ of energy $\omega_r$. Furthermore, we allow for the Rydberg state to be shifted by energy $V(R)$ if an atom at distance $R$ is excited to a Rydberg state (we recall we use $\hbar = 1$).

Written in the $\{|\ell\rangle, |\ell'\rangle, |r\rangle\}$ basis, the Hamiltonian (56) reads

$$H = \sum_{j=\ell,\ell'} \Omega_{j,r} |j\rangle \langle j| + \text{H.c.} + \Delta_j |j\rangle \langle j| + V |r\rangle \langle r|.$$  

The associated unitary operator $U = e^{-itH}$ can in principle be obtained analytically by diagonalizing (57), leading to a cubic equation for the eigenvalues. The situation simplifies at two-photon resonance.
\[ \Delta_\ell = \Delta_\ell' = 0. \] In the limit of large interaction, \( V \to \infty \), we get

\[ U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-iVt} \end{pmatrix}. \] (58)

Denoting \( \Omega_0 = \Omega_{\ell,r}, \Omega_1 = \Omega_{\ell',r} \) for the ease of notation, the \(|r\rangle\langle r|\) element of \( U \) for a general \( V \)
reads

\[ U_{rr} = e^{-1/2itV} \begin{pmatrix} \cos \left( \frac{1}{2} t \sqrt{V^2 + 4\Omega^2} \right) \\ -iV \sin \left( \frac{1}{2} t \sqrt{V^2 + 4\Omega^2} \right) \sqrt{V^2 + 4\Omega^2} \end{pmatrix}, \] (59)

where

\[ \Omega = \sqrt{\abs{\Omega_0}^2 + \abs{\Omega_1}^2}. \] (60)

Requiring that the Rydberg population returns to zero at the end of the evolution, if it was initially unpopulated, is equivalent to \( |U_{rr}| = 1 \). This leads to the condition

\[ t = \frac{2m\pi}{\sqrt{V^2 + 4\Omega^2}}, \quad m \in \mathbb{N}. \] (61)

Specifically, in the absence of interaction (\( V = 0 \)) and for \( m \) odd we get the expression Eq. (20)

\[ U_{3\text{-level}}^{\ell,\ell'} = \begin{pmatrix} \cos \frac{\theta}{2} & e^{i\varphi} \sin \frac{\theta}{2} & 0 \\ -iV \sin \frac{\theta}{2} & -\cos \frac{\theta}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \] (63)

with \( \theta \) and \( \varphi \) defined in Eqs. (21).

The unitary Eq. (19) is obtained in an analogous way when considering a two-level system with levels \(|\ell\rangle, |\ell'\rangle\) coupled by a laser with Rabi frequency \( \Omega_{\ell,\ell'} \) and described by the Hamiltonian

\[ H = \Omega_{\ell,\ell'} |\ell\rangle \langle \ell'| + \text{H.c.} + \Delta |\ell'\rangle \langle \ell'|, \] (62)

where \( U = e^{-itH} \) is again evaluated on resonance \( \Delta = 0 \).

### D.2 Derivation of the qudit controlled-phase gates

In this section we describe a systematic construction of the qudit controlled-phase gates Eqs. (26) and (27).

#### D.2.1 On qudit controlled-phase gate Eq. (26)

Let us start with the gate (26) acting on a qubit, \( d = 2 \). First we note that the action of the unitary \( U_0^{(21)} \) corresponds to

\[ U_0^{(21)} |\ell\rangle |\ell'\rangle = \begin{cases} e^{i\frac{\gamma}{2}} |\ell\rangle |\ell'\rangle \quad &\text{if } (\ell, \ell') = (1, 0) \\ |\ell\rangle |\ell'\rangle \quad &\text{otherwise.} \end{cases} \] (63)

It then follows that for a qubit, the controlled-phase gate expressed in the basis \{\(00\), \(01\), \(10\), \(11\)\} reads

\[ \text{CP} \left( \frac{\gamma}{2} \right) = U_0^{(21)} U_1^{(21)} \text{diag}(1, e^{i\gamma/2}, e^{i\gamma/2}, 1) \] (64)

(in this case, an alternative sequence \( U_0^{(12)} U_0^{(21)} \) or \( U_0^{(21)} U_0^{(12)} \) yields the same result). Furthermore, the cost of the controlled-phase gate (64) is \( [\text{CP}] = 2 = d \).

Motivated by the construction in Eq. (64), we next wish to extend it to qutrits. Here the situation becomes slightly more involved. Observing that the gate \( U_1^{(q|r)} \) is always diagonal, i.e. introducing
at most a phase for each element $|\ell\rangle|\ell'\rangle$ of the two-qutrit system, we introduce the following graphical notation for a representation of a general diagonal gate acting in the two-qutrit space

$$
U \rightarrow \begin{array}{|c|c|}
02 & 22 \\
00 & 21 \\
o1 & 12 \\
10 & 01 \\
11 & 11 \\
\end{array}
,$$

where the labels on the right-hand side denote the coordinates of the diagonal element $(\ell\ell'|U|\ell\ell')$ of $U$. Since $U$ is diagonal with elements, which are either 1 or a pure phase $e^{i\varphi}$, we shall use the notation • for 1 and $\varphi$ for the pure phase for easier readability. With the help of (65), we can list the action of the possible six unitaries $U_t^{(q|q_c)}$ acting on the two qutrits:

$$
\begin{align*}
U_2^{(1|2)} &= \begin{array}{|c|c|}
\cdot & \cdot \\
\cdot & \cdot \\
\gamma/2 & \cdot \\
\gamma/2 & \cdot \\
\gamma/2 & \cdot
\end{array} \\
U_2^{(2|1)} &= \begin{array}{|c|c|}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\gamma/2 & \cdot
\end{array} \\
U_1^{(1|2)} &= \begin{array}{|c|c|}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot
\end{array} \\
U_1^{(2|1)} &= \begin{array}{|c|c|}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\gamma/2 & \cdot
\end{array} \\
U_0^{(1|2)} &= \begin{array}{|c|c|}
\cdot & \cdot \\
\cdot & \cdot \\
\gamma/2 & \cdot \\
\gamma/2 & \cdot \\
\gamma/2 & \cdot
\end{array} \\
U_0^{(2|1)} &= \begin{array}{|c|c|}
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot
\end{array}.
\end{align*}
$$

(66)

It then follows that the controlled-phase gate can be obtained by concatenation of all the gates in (66)

$$
\text{CP}(\gamma) = U_2^{(1|2)}U_2^{(2|1)}U_1^{(1|2)}U_1^{(2|1)}U_0^{(1|2)}U_0^{(2|1)} = \begin{array}{|c|c|c|}
\gamma & \cdot \\
\cdot & \gamma \\
\cdot & \gamma \\
\cdot & \gamma
\end{array},
$$

(67)

which corresponds to Eq. (24) up to a global phase. It is then a straightforward exercise to verify that the manifestly symmetric form suggested by (67) generalizes to higher $d$ yielding the expression (26). In fact, it also applies to a qubit, yielding the cost [CP] = 4 = 2$d$, which is twice as large as that of (64). For the ease of exposition and compactness of notation, and motivated by the fact that we are mainly interested in applications with $d > 2$, we have kept the controlled-phase gate (26) also for qubits in the main text.

**D.2.2 On qudit controlled-phase gate Eq. (27)**

To understand the logic behind the construction of (27), it is instructive to work out a specific example and we shall consider the simplest case beyond qubit, the qutrit. To appreciate the role of the qudit CX gates in (27), we focus specifically on the equal-state elements $|\ell\rangle|\ell\rangle$ of the two-qutrit system. The evolution of these elements under the action of the controlled-phase gate (27) can be symbolically
Proposition E.2. Consider a quantum computer for which the graph of possible interactions is a
line graph in \( d \), \( d \geq 2 \). We can easily see that the previous strategy is close to optimal, and that no strategy can exist that
matches the topology of the quantum computer well. Our simulations considered complete graphs,
For first experiments, it might be prudent to focus on problem instances where the problem graph
matches the topology of the quantum computer well. Our simulations considered complete graphs,
for those it is natural to ask: How many swap operations are required to let a limited-interaction
quantum computer execute the algorithm on a complete graph? Or, if the hardware allows swaps on
distinct nearest-neighbour pairs of qudits to be performed in parallel, how many layers of swaps?

We can construct a simple sequence of swaps that achieves this task using not too many layers:

\[
\begin{align*}
|22\rangle & \xrightarrow{\text{CX}_{1,2}^{(12)}} |12\rangle \xrightarrow{\text{CX}_{1,2}^{(12)},\pi} e^{i\gamma}|02\rangle \leftrightarrow P_0^{(1)}(\gamma) \\
|11\rangle & \xrightarrow{\text{swap}} |11\rangle \xrightarrow{\text{swap}} e^{i\gamma}|01\rangle \leftrightarrow P_0^{(1)}(\gamma) \\
|00\rangle & \xrightarrow{\text{swap}} |00\rangle \xrightarrow{\text{swap}} e^{i\gamma}|00\rangle \leftrightarrow P_0^{(1)}(\gamma).
\end{align*}
\]

Here, the CX gates (listed only in the first line to avoid cluttering) act in the direction indicated by
the arrows. We thus see that the purpose of the CX gates is to bring the equal state elements to the
elements of the form \( |0\ell\rangle \), \( \forall \ell \), i.e. \( \text{CX}_{0,1}|0\ell\rangle \rightarrow |0\ell\rangle \), \( \ell = 0, 1, 2 \). Since there are at most
three states of the form \( |0\ell\rangle \), the states \( |02\rangle, |01\rangle, |00\rangle \) on the right of (68) exhaust all such states. The
phase gate \( P_0^{(1)} \) then imprints a phase \( \gamma \) to all these states, which is highlighted in red, and acting
with the CX gates backwards yields the controlled-phase gate (27). Similarly to the construction in
Sec. D.2.1 above, it is straightforward to verify that the scheme (68) holds also for higher \( d \).

E Simulating complete graphs using only nearest-neighbour interactions

For first experiments, it might be prudent to focus on problem instances where the problem graph
matches the topology of the quantum computer well. Our simulations considered complete graphs,
so for those it is natural to ask: How many swap operations are required to let a limited-interaction
quantum computer execute the algorithm on a complete graph? Or, if the hardware allows swaps on
distinct nearest-neighbour pairs of qudits to be performed in parallel, how many layers of swaps?

We can construct a simple sequence of swaps that achieves this task using not too many layers:

Proposition E.1. There exists a sequence of swaps such that the complete graph can be simulated on
a line graph in \( n - 2 \) layers of swaps. The total swap count for this protocol is \( \frac{(n-1)(n-2)}{2} \).

Proof. First assume \( n \) is even (the argument follows the same structure if \( n \) is odd). The sequence
will be generated by alternating the two following two sets of swaps: \( \pi = \{(1, 2), (3, 4), \ldots, (n-1, n)\} \),
\( \sigma = \{(2, 3), (4, 5), \ldots, (n-2, n-1)\} \). First observe what happens when applying \( \pi \) and \( \sigma \) to a qudit
starting in an odd position \( i \). If \( i \neq n - 1 \), this qudit will move to \( i + 1 \) because of application of \( \pi \),
and then to \( i + 2 \) by \( \sigma \). If \( i = n - 1 \), the qudit will move to position \( n \). Similarly, for a qudit starting
in even position \( j \) will move to position \( j - 2 \), except that the qudit at position 2 will move to position
1.

It is easily checked that after \( n - 2 \) layers of swaps, all pairs of qudits have been nearest-neighbours
at some stage of the process. \( \blacksquare \)

For a more thorough analysis of swapping sequences for executing all-to-all interactions, see also
Ref. [13] and Ref. [78], where variants of the previous construction are analysed in-depth.

We can easily see that the previous strategy is close to optimal, and that no strategy can exist that
is much better: by a simple counting argument the number of layers of swaps required to enable all
interactions of a complete graph on a line could not be much lower. This motivates us to look for the
Corresponding lower bound.

Proposition E.2. Consider a quantum computer for which the graph of possible interactions is a
line of \( n \) qudits. Then at least \( \frac{1}{7} \binom{n}{2} - o(n) \) SWAP gates are required to enable all-to-all interactions.
Additionally, at least \( \frac{n}{2} - 1 \) layers of SWAP gates are required.
Figure 14: Fidelities (72), (73) and (74) [solid orange, blue and green lines] as a function of the 2-gate error $p_2$, cf. Eqs. (44) and (75), for $|\psi_{in}\rangle = |+^2\rangle \otimes 2$ and CP gate phase $\gamma = \pi/2$. The approximate agreement between $F_{\text{open}}$ and $F_{\text{err}}$ for small values of $p_2$ was obtained by setting $\eta \approx 1.5$ in (75).

Proof. We can lower bound the number of necessary swaps by a simple counting argument: The complete graph has $\binom{n}{2}$ edges, so that number of interactions will be required. A line of $n$ qubits has $n-1$ edges, which is how many interactions are possible before the first SWAP gate. Next, every SWAP gate gives two qubits a new neighbour, enabling at most two new interactions. The required swap count therefore is at least $\frac{1}{2} \binom{n}{2} - (n-1)$.

For the number of layers, note again between a layer of SWAP gates at most $n-1$ nearest-neighbour interactions are possible. This implies a lower bound of $\frac{n}{n-1} = \frac{n}{2}$ of layers, hence $\frac{n}{2} - 1$ layers of SWAP gates to be able to have all interactions.

On experimental errors

The gate errors for Rydberg atoms have been analysed extensively, e.g. in [111], and verified experimentally, e.g. in [88]. The total error budget consists of various contributions including the collisions of the atoms with background gas, phase and intensity laser noise, light scattering from both the trapping and resonant lasers operating the processor, spurious level shifts from fluctuating electric and magnetic fields or atom motion in the tweezer traps. A detailed analysis of the individual contributions of these error sources goes beyond the scope of the present work. Given that the total 2-gate error dominates over the 1-gate one [111] and is one of the bottlenecks for scalability across different platforms, we consider only the 2-gate error. Specifically, we consider the error due to the spontaneous decay from the Rydberg levels.

Let us start with a brief overview of the situation and let us consider $P$ as the qudit manifold (we recall we consider the excitation to a Rydberg state as a resonant two step process through the long-lived $P$ manifold, cf. Sec. 6). For the corresponding Rydberg manifold $|Ry\rangle = |n^3S_1\rangle$, the Rydberg atom decays approximately with branching ratios $\approx 1:4$ to the $(np)^3P_J$ and $(5p)^3P_J$ manifolds respectively, where $n > 5$. Out of the decay to $(5p)^3P_J$, it branches further with ratios $\approx 1:3:5$ to the $J = 0, 1, 2$ manifolds and then further among the $F-$manifolds and the respective $m_F$ states (allowed by the selection rules). Consequently, the probability of the Rydberg state to decay back to the qudit manifold (here $|3\text{P}_2\rangle$) – but not necessarily to the same qudit state – is $\approx 0.4$ [140, 141].

The complete dynamics can be described by the standard means of a master equation for amplitude decay including all the decay channels. However, since the probability to decay outside the qudit manifold $\gtrsim 0.6$, we consider a simplified dynamics of a pair of interacting qudits $q = 1, 2$ given by

$$\dot{\rho} = i[\rho, H] + \frac{\Gamma}{2} \sum_{q=1,2} \sum_{\ell=0}^{d-1} 2\sigma_q(\ell)^- \rho \sigma_q(\ell)^+ - \{\sigma_q(\ell)^+, \sigma_q(\ell)^-, \rho\}$$

(69)
where $\sigma^{(q)}_\ell = |r^{(q)}_\ell\rangle\langle\text{aux}|$, $\{\cdot,\cdot\}$ is the anticommutator and a single auxiliary level $|\text{aux}\rangle$ is used to model the decay at rate $\Gamma$ outside the qudit manifold.

We seek to compare the effect of the realistic errors described by the master equation (69) to the unitary error model introduced above. Let us denote

$$\rho_{\text{id}} \equiv \text{CP} |\psi_{\text{in}}\rangle \langle \psi_{\text{in}}| \text{CP}^\dagger$$

the ideal state obtained upon an action of the qudit controlled-phase gate (24) on some pure state $|\psi_{\text{in}}\rangle$. We also define the fidelity between two quantum states with density matrices $\rho, \sigma$ as usual,

$$F(\rho, \sigma) = \text{Tr} \left[ \sqrt{\sqrt{\rho} \sigma \sqrt{\rho}} \right].$$

Next, we define the average fidelity of a state corresponding to the error model (44) as

$$F_{\text{err}}(p_2) = (1 - p_2) + \frac{p_2}{|U^{\otimes 2} \setminus \{I\}|} \sum_{U \in U^{\otimes 2} \setminus \{I\}} F(U \rho_{\text{id}} U^\dagger, \rho_{\text{id}}).$$

Similarly, denoting the density matrix resulting from the open dynamics given by the master Eq. (69) as $\rho_{\text{open}}$, we define the fidelity

$$F_{\text{open}} = F(\rho_{\text{open}}, \rho_{\text{id}})$$

and the average fidelity between $\rho_{\text{open}}$ and the states generated by (44)

$$F_{\text{err-open}} = (1 - p_2) F(\rho_{\text{id}}, \rho_{\text{open}}) + \frac{p_2}{|U^{\otimes 2} \setminus \{I\}|} \sum_{U \in U^{\otimes 2} \setminus \{I\}} F(U \rho_{\text{id}} U^\dagger, \rho_{\text{open}}).$$

The resulting fidelities depend on $|\psi_{\text{in}}\rangle$ and they in principle vary in the course of the algorithm. Furthermore, there is no direct unambiguous identification between the the error probability $p_2$ of the error model (44) and that of the open dynamics (69). Denoting the gate time of the cost unitary (27) as $t_{\text{CP}}$, the decay probability of an excited atom $\propto 1 - e^{-\Gamma t_{\text{CP}}}$. This motivates a parametrization

$$p_2 = 1 - e^{-\eta \Gamma t_{\text{CP}}},$$

where we have introduced a phenomenological factor $\eta$.

For illustration, in Fig. 14 we show the fidelities (72), (73) and (74) as a function of the error probability $p_2$, cf. Eq. (44), for the “isotropic” initial state $|\psi_{\text{in}}\rangle = |+\rangle^{\otimes 2}, |+d\rangle = 1/\sqrt{d} \sum_{\ell=0}^{d-1} |\ell\rangle$.

Denoting $\mathcal{P}_Q$ the projector on the qudit subspace, $Q = F$, we have also verified that the state evolution subject to the open dynamics (69) satisfies to a good accuracy $\mathcal{P}_Q \rho_{\text{open}} \mathcal{P}_Q \approx e^{-\Gamma t_{\text{CP}}} \rho_{\text{id}}$. This corresponds well to the naive guess that the resulting density matrix is just the rescaled ideal matrix $\rho_{\text{id}}$ due to the decay with rate $\Gamma$ outside the qudit subspace.

In summary, while the error model (44) seems to capture qualitatively the decrease of the state fidelity, it clearly cannot account for the dynamics outside the qudit subspace and more rigorous analysis of the errors in the context of the QAOA is desirable, see also [66] for related developments.