Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks

Michel Fabrice Serret, Bertrand Marchand, and Thomas Ayral

1 Atos Quantum Laboratory, Les Clayes-sous-Bois, France
2 Ecole Polytechnique, Palaiseau, France

Platforms of Rydberg atoms have been proposed as promising candidates to solve some combinatorial optimization problems. Here, we compute quantitative requirements on the system sizes and noise levels that these platforms must fulfill to reach quantum advantage in approximately solving the Unit-Disk Maximum Independent Set problem. Using noisy simulations of Rydberg platforms of up to 26 atoms interacting through realistic van der Waals interactions, we compute the average approximation ratio that can be attained with a simple quantum annealing-based heuristic within a fixed temporal computational budget. Based on estimates of the correlation lengths measured in the engineered quantum state, we extrapolate the results to large atom numbers and compare them to a simple classical approximation heuristic. We find that approximation ratios of at least \( \approx 0.84 \) are within reach for near-future noise levels. Not taking into account further classical and quantum algorithmic improvements, we estimate that quantum advantage could be reached by attaining a number of controlled atoms of \( \approx 8,000 \) for a time budget of 2 seconds, and \( \approx 1,000 - 1,200 \) for a time budget of 0.2 seconds, provided the coherence levels of the system can be improved by a factor 10 while maintaining a constant repetition rate.

In recent years, quantum hardware has witnessed very rapid technological progress \([1, 2]\), while still being characterized by a substantial level of imperfections \([3, 4]\). It is thus a very timely task to realistically assess the requirements for reaching a break-even point with classical algorithms, and hence guide future necessary developments. This task first requires the identification of problems that are difficult to solve classically, and for which some quantum platforms are expected to provide some form of speedup. Among the many classes of computational problems put forth as promising candidates, NP-hard combinatorial optimization problems stand out as prototypical hard problems. Many of them have industrial relevance, and exhibit a natural encoding onto quantum machines, through Ising Hamiltonians and quantum annealing (see, e.g., \([5]\)). Recently, for instance, the MaxCut and the Maximum Independent Set problems have received attention as candidates for quantum advantage \([6, 7, 8]\). These optimization problems are particularly well suited to noisy intermediate scale quantum (NISQ) computers, as they can be solved using various hybrid quantum-classical approaches \([9–12]\).

The predictive assessment of a break-even point calls for a realistic modeling of imperfections and a precise definition of the algorithmic success metric. While the above-mentioned quantum approaches (including quantum annealing) provide a general framework for tackling combinatorial optimization problems with quantum machines \([9, 10]\), very little is known about their performance under realistic hardware models, including a precise modeling of the microscopic Hamiltonian, decoherence effects, and readout errors. These imperfections give these quantum algorithms a stochastic and heuristic nature. In particular, approximate (i.e., valid, but sub-optimal) solutions to a given instance of a problem might be returned. These approaches should therefore be compared with classical algorithms designed for the similar task of approximately solving NP-hard problems, namely approximation algorithms \([13]\). For such a comparison, the most widespread success metric is known as the approximation ratio. This quality factor must be put in perspective with the execution time of both kinds of algorithms.

Due to the spatial structure of their interatomic interactions, platforms of Rydberg atoms \([14]\) have been recently proposed \([8]\) as candidate quantum processors to solve a subclass of the MIS problem called the Unit-Disk Maximum Independent Set problem (UD-MIS). For instance, Ref. \([8]\) compared the probability for finding the UD-MIS solution using the—analog—quantum annealing algorithm and the—digital (i.e gate-based)—Quantum Approximate Optimization Algorithm (QAOA \([12]\)) algorithm in the absence of quantum noise, showing the potential of the QAOA approach. Ref. \([15]\) investigated the generic effect of dephasing and relaxation noise on the QAOA algorithm with a simplified hard-sphere model of the Rydberg interaction.

Despite the promises of the digital (QAOA) approach and the steady progress to make Rydberg platforms suitable digital platforms \([16, 17]\), they are still lagging behind the two major digital quantum computing platforms, namely superconducting \([18]\) and trapped-ion \([19]\) processors, in terms of gate fidelities. Conversely, the number of controllable atoms in today’s most advanced analog Rydberg setups is close to a hundred \([20, 21]\) and far exceeds the number of qubits of digital platforms, thus severely challenging classical simulation capabilities. While recent experimental work \([22]\) demonstrated the implementation of adiabatic processes that could allow to tackle UD-MIS on grid graphs (square and triangular), other recent works \([20, 23, 24]\) demonstrated the possibility for arbitrarily positioning tens of neutral atoms, making it possible for the Rydberg platform to tackle almost any
of the graph instances coming up in UD-MIS problems. In this work, we therefore quantitatively compare the potential of Rydberg platforms, described in a realistic way and operated as analog quantum processors, to solve the UD-MIS optimization problem using a quantum annealing-based approximation method, with respect to a simple classical approximation approach for this problem.

We start by setting up a generic methodology for comparing quantum and classical randomized approximation algorithms via the comparison of the maximum approximation ratio achievable within a predefined computational time budget. In particular, we study how the system size and number of random samples (or repetitions) impact the approximation ratio.

On the classical side, we introduce a new (to our best knowledge), simple locality-based approximation heuristic, inspired from state-of-the-art approximation algorithms [25–27], and benchmark it on a class of experimentally-implementable random graphs. We qualify it with respect to our time-budget metric, and use it to estimate a classical boundary.

On the quantum side, we construct a hardware model with a realistic account of the van der Waals interactions, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.

We conduct, under this model and improved noise models mimicking near-term and future hardware improvements, numerical noisy simulations of Rydberg systems executing an annealing schedule (similar to the schedules used in [22] and [8]) aimed at heuristically tackling UD-MIS. We say this process constitutes a “quantum-annealing-based heuristic”, as it differs from standard quantum annealing by the fact that the final Hamiltonian (the experimental Rydberg “resource” Hamiltonian) is not the ideal “target” UD-MIS Hamiltonian. We reach the unprecedented number of 26 atoms (to our best knowledge), simple locality-based approximation, of decoherence and readout errors, and we validate it against published experimental results.
algorithm that would take as input any unit-disk graph, and return an independent set of maximum size, with certainty or good probability, in polynomial time.

However, while restricting the input of the MIS problem to unit-disk graphs preserves NP-hardness, the difficulty of approximately solving it drastically changes. UD-MIS indeed allows for efficient approximation algorithms, as we will now discuss.

An approximation algorithm to an NP-hard problem is a polynomial-time algorithm returning solutions that are guaranteed to be within a certain percentage of the optimum in terms of a cost function. Here, the cost function is the cardinality $|S|$ of an independent set. Formally, a polynomial-time algorithm $A$ may be called an approximation algorithm \cite{13, 29} to a given NP-hard maximization problem if, given an instance $I$ (in our case, a graph $G$), it returns a solution $S$ such that:

$$\alpha \equiv \frac{C(S, I)}{C(S^*, I)} \geq \phi(|I|),$$

where $\phi$ is a scalar function, sometimes called approximation factor, specifying a guarantee on the approximation ratio achieved achieved by the algorithm on inputs of size $|I|$. NP-hard optimization problems may then be classified with respect to the approximation factor achieved by their best approximation algorithm, on what could be called an “approximability scale” from least approximable to most approximable.

For instance, a maximization problem only allowing for factors $\phi(|I|)$ decreasing with $|I|$ is usually informally considered as hard to approximate. This is in fact the case for the general MIS problem, which cannot allow for anything better than a polynomially-decreasing factor under specific complexity-theoretic assumptions \cite{13}.

Some other NP-hard problems, such as MaxCut, allow for a constant-factor approximation algorithm, i.e. with $\phi(|I|) = \phi$ independent of the input size. A value close to 1, with a simple enough algorithm, may render the problem easy to approximately solve in practice.

UD-MIS falls in yet another category of problems, usually informally considered as the “most easily approximable problems”, namely those allowing for a PTAS, or Polynomial-Time Approximation Scheme.

Formally, a PTAS is an approximation algorithm that, given any $\epsilon > 0$ achieves a factor $(1 - \epsilon)$ in $O(\text{poly}(|I|))$. I.e., given any fixed $\epsilon$, the complexity of returning a solution within a factor $(1 - \epsilon)$ of the optimal is polynomial in the size of the input. A PTAS may therefore return solutions arbitrarily close to the optimal, at the expense of a longer computational time (the degree and coefficients of the polynomial may increase when $\epsilon$ decreases). In other words, a PTAS makes a problem “constant-factor approximable for any factor”.

Table I gives examples of problems with varying degrees of approximabilities. Note how the MIS problem goes from inapproximable (polynomial factor) to constant-factor and PTAS-approximable when one restricts its input to bounded-degree graphs and unit-disk graphs, respectively.

All PTAS’s for UD-MIS rely on the common idea of partitioning the graph into small sub-instances that are solved separately. The returned solution is then the union over the local solutions. Their practical computational complexity as $\epsilon$ increases is notably quite sharp \cite{27, 30}, which raises the question of how quantum approaches may fare compared to them.

Two main strategies can be distinguished for the UD-MIS approximation: 2-level shifting schemes \cite{25, 27, 30, 31} and Breadth-First-Search-sphere-based schemes \cite{26}. We implemented a locality-based heuristic inspired from the latter, as described in section II.B. For this approach, as we shall explain in the next section, we look at the best approximation ratio it can achieve within a certain

| Problem         | Best apx. factor. | \cite{13} |
|-----------------|-------------------|----------|
| general MIS     | polynomial        |          |
| bounded-degree MIS | constant         |          |
| MaxCut          | constant (\(\sim 0.878\)) |          |
| VertexCover     | PTAS              |          |
| UD-MIS          | PTAS              |          |
| Knapsack        | f-PTAS            |          |

Table I: Examples of NP-hard optimization problems with varying degrees of approximation hardness, sorted from least approximable (general MIS) to most approximable (Knapsack, which allows for a strong version of PTAS called f-PTAS \cite{13})
C. Methodology: comparing quantum and classical approximation algorithms

The basic idea behind our work is to compare the best classical and quantum approaches, given the same time budget, to the UD-MIS problem, with realistic assumptions as to the imperfections of the quantum hardware.

We view our work as a comparison of classical and quantum approximation algorithms [13] to this problem. We therefore focus on a standard metric to evaluate the quality of the approximate solutions, namely the approximation ratio:

\[ \alpha = \frac{C(S, G)}{C(S^*, G)}, \quad (3) \]

where \( S \) denotes an approximate solution to a particular graph instance \( G \), \( S^* \) the optimal solution, and \( C \) the cost function that one seeks to maximize. In this study, we typically look at expected approximation ratios, averaged over both random instances of graphs and independent runs of a given algorithm, whether classical or quantum.

Moreover, in order to take into account the fact that one may have the time to run an algorithm many times within a predefined time budget, either serially or in parallel, and is then likely to select the best outcome as output, we also look at what we call the “maximum approximation ratio” or “best-of-\( n_{\text{shots}} \) approximation ratio”.

Formally, we define it as:

\[ \alpha(n_{\text{shots}}) \equiv \max \frac{C(S_i, G)}{C(S^*_i, G)} \quad (4) \]

where \( S_1, \ldots, S_{n_{\text{shots}}} \) are approximate solutions resulting from \( n_{\text{shots}} \) independent runs of a given algorithm \( \mathcal{A} \) on the same instance \( G \). It may be regarded as the plain approximation ratio defined in Eq. (3), but achieved by a meta-algorithm \( \mathcal{A}' \) consisting of \( n_{\text{shots}} \) independent runs of \( \mathcal{A} \) and the selection of the best result as output.

As with the basic approximation ratio, we then look at that quantity averaged over random instances and/or several independent runs of \( \mathcal{A}' \) (each of them in turn consisting in \( n_{\text{shots}} \) independent runs of the algorithm \( \mathcal{A} \) under study).

The average maximum expectation ratio reflects the typical procedure one would follow with a quantum algorithm or randomized classical approach (perform many repetitions of the state-preparation/algorithm, either serially or in parallel, and keep the best solution among all the repetitions). However, we also keep track of the basic average expectation ratio, which we also find instructive as an assessment of the typical achievable intrinsic approximation ratio. Besides, it provides a lower bound for the maximum approximation ratio.

II. METHODS

A. Quantum approach: Solving UD-MIS using Rydberg atoms

In this section, we describe how the solution of the UD-MIS problem can be approximated using a hybrid quantum-classical algorithm whose quantum part makes use of a platform of Rydberg atoms.

1. Variational Quantum Simulation approach to the UD-MIS optimization problem

The UD-MIS problem can be reformulated as a minimization problem that consists in finding the ground-state energy of the following Hamiltonian:

\[ H_{\text{target}} \equiv - \sum_{i \in V} \hat{n}_i + u \sum_{i,j \in E} \hat{n}_i \hat{n}_j \quad (5) \]

with \( \hat{n}_i = (I - \sigma_i^z)/2 \). The \( u \) parameter is a fixed Lagrange multiplier, whose value can be optimized in an outer optimization loop. In this work, we will fix \( u \) to a value \( u > 1 \) (namely \( u = 1.35 \)), which guarantees that the ground state of \( H_{\text{target}} \) will necessarily be an independent set (IS). The full derivation of the cost function \( H_{\text{target}} \) is given in Appendix C.

To solve (at least approximately) the problem of finding the ground state of \( H_{\text{target}} \), we follow the Variational Quantum Simulation (VQS) framework, as introduced by [11]. An analog version of the Variational Quantum Eigensolver (VQE, [32]), VQS consists in minimizing the expectation value of \( H_{\text{target}} \) over a family of variational states \( |\Psi(\hat{\theta})\rangle \) constructed using an analog quantum computer:

\[ E_{\text{target}}(\hat{\theta}) \equiv \langle \Psi(\hat{\theta}) | H_{\text{target}} | \Psi(\hat{\theta}) \rangle \quad (6) \]

The ansatz states are prepared by a (generically time-dependent) “resource Hamiltonian” \( H_{\text{resource}}(\hat{\theta}) \) whose parameters \( \hat{\theta} \) are optimized with a classical computer to minimize \( E_{\text{target}}(\hat{\theta}) \). Typically, possible variational parameters are the coefficients of the various terms of the resource Hamiltonian (if they are tunable), the duration of application of each term, etc. In a nutshell, VQS consists in designing a Hamiltonian schedule \( H_{\text{resource}}(\hat{\theta}) \) to prepare a final state \( |\Psi(\hat{\theta})\rangle \) that is as close as possible to the ground state of \( H_{\text{target}} \).

2. Resource Hamiltonian for a Rydberg platform

We now turn to the available resource terms for \( H_{\text{resource}}(\hat{\theta}) \) when working with a Rydberg platform. One
of the most common Rydberg setups implements an Ising Hamiltonian [14]:

\[ H_{\text{Rydberg}}(t) = \frac{\omega(t)}{2} \sum_{i \in V} \hat{\sigma}_i^x - \delta(t) \sum_{i \in V} \hat{n}_i + \sum_{ij \in E} V_{ij} \hat{n}_i \hat{n}_j, \]

(7)

with three "resource terms": a Rabi term, a dephasing term and a van der Waals interaction term. While the third term is time-independent (in the following, we will take \( V/h = 2.7 \text{ MHz} \)), following [22], the first two terms are time-dependent and tunable, and as such \( \omega(t) \) and \( \delta(t) \) can be regarded as variational parameters \( \frac{\delta}{\delta \theta} \) (after a suitable discrete parametrization). Several strategies are then possible to design schedules over \( H_{\text{resource}} \) that prepare a (possibly approximate) ground state of \( H_{\text{target}} \). In this work, we will use the quantum annealing schedules described in [22] and [8] to build \( H_{\text{resource}} \) from Eq. (7).

3. Quantum annealing-inspired schedule construction

The goal of the quantum annealing (QA) algorithm is to prepare the ground state of a given \( H_{\text{target}} \). Starting from a \( H_{\text{target}} \), QA prescribes a generic form for the time-dependent Hamiltonian that is meant to prepare the target ground state. This Hamiltonian, which, in reference to the VQS notation, we will also refer to as a resource Hamiltonian, usually assumes the following form

\[ H_{\text{annealing}}(t) = \Omega(t) \sum_i \hat{\sigma}_i^x + \tilde{\Omega}(t) H_{\text{target}} \]

(8)

with the initial and final conditions \( \Omega(t=0)=1, \tilde{\Omega}(t=0)=0 \), \( \Omega(t=t_1)=0, \tilde{\Omega}(t=t_1)=1 \), and the assumption that the initial state is \( |\Psi(t=0)\rangle = |+\rangle^\otimes n \) (i.e., the ground state of \( H_{\text{resource}}(t=0) \)). We note that \( |+\rangle \) is defined as \( |+\rangle = (|0\rangle + |1\rangle)/\sqrt{2} \), where |0\rangle (resp. |1\rangle) denotes the ground (resp. excited, i.e. Rydberg) state of the atom.

A main advantage of QA is that the so-called adiabatic theorem (see, e.g., [10]) guarantees that this temporal evolution prepares the (exact) ground state \( |\Psi^0\rangle \) of \( H_{\text{target}} \) (provided the annealing time is long enough).

A drawback of QA is that the prescribed form for \( H_{\text{annealing}} \) may not match the resource Hamiltonian that is available on a given platform. In our case, indeed, the three resource terms available on the Rydberg platform cannot straightforwardly implement \( H_{\text{resource}} \). The most consequential difference is the interaction term: while QA would dictate a hard-sphere interaction between neighboring atoms \( u \sum_{ij \in E} \hat{n}_i \hat{n}_j \) term in (5)), Rydberg atoms interact with a \( 1/r^6 \) potential. Moreover, the value of this potential cannot be turned on and off at will, as would be required by a time-dependent \( \Omega(t) \) drive fulfilling the initial and final conditions.

Several advanced strategies could be investigated to make the Rydberg resource Hamiltonian better match the resource Hamiltonian prescribed by QA. For instance, a stroboscopic scheme was proposed in [8]. One could also optimize, under the constraint of keeping the graph identical, the locations of the atoms themselves, to minimize the effect of long-range tails. While those strategies could likely lead to a better resource Hamiltonian (insofar as it would be closer to the QA form, and thus be endowed with mathematical guarantees via the adiabatic theorem), they have not yet been implemented experimentally and therefore warrant a detailed separate analysis that is outside the scope of this paper.

Here, instead, we deliberately stick with already experimentally implemented terms [23] in an effort to quantify what can be achieved with today’s technology. We use the full experimental Rydberg Hamiltonian as our resource to produce states, and evaluate results through the ideal target Hamiltonian from Eq. (5). We merely take inspiration from the QA resource Hamiltonian to design a resource Hamiltonian that prepares a reasonable approximation to the target ground state while complying with the constraints of the Rydberg architecture:

**Initialization:** One cannot start from the superposition state \( |\Psi(t=0)\rangle = |+\rangle^\otimes n \), but only from the state \( |\Psi(t=0)\rangle = |0\rangle^\otimes n \) where no atoms are in the excited (Rydberg) state.

**Schedule:** As already mentioned, the van der Waals interaction of Eq. (7) differs from the hard-sphere interaction \( \sum_{ij \in E} \hat{n}_i \hat{n}_j \) required by our problem, Eq. (8). Moreover, we cannot turn it off at will as the initial conditions on \( \tilde{\Omega}(t) \) require. As a partial accommodation for these differences, we design the time evolution of \( H_{\text{Rydberg}} \) as follows, with three stages (illustrated in Fig. 2):

(i) for \( 0 < t < t_{\text{rise}} \), we are going to increase \( \omega \) from 0 to a maximum value \( \omega_0 \). The goal of this stage is to bring the system from state \( |0\rangle^\otimes n \) to state \( |+\rangle^\otimes n \) (indeed, the ground state at \( \omega = 0 \) (resp. \( \omega = \omega_0 \)) is \( |0\rangle^\otimes n \) (resp. \( |+\rangle^\otimes n \)).

(ii) for \( t_{\text{rise}} < t < t_{\text{rise}} + t_{\text{sweep}} \), we are going to increase the value of \( \delta \) from \( \delta_0 \) to \( \delta_{\text{max}} \) in order to favor sets with a large number of atoms, while keeping the tunneling (Rabi) term on;

(iii) for \( t_{\text{rise}} + t_{\text{sweep}} < t < t_{f} \), we are going to decrease \( \omega \) from \( \omega_0 \) to 0. The goal is for the state of the system at the end of this stage to be the ground-state \( |\Psi_{\text{resource}}\rangle \) of

\[ H_{\text{Rydberg}}(t = t_f) = -\delta_{\text{max}} \sum_{i \in V} \hat{n}_i + \sum_{ij \in E} V_{ij} \hat{n}_i \hat{n}_j \]

We note that the very same adiabatic process as presented above has already been implemented experimentally, albeit for grids and not general graphs [22]. Importantly, even if this schedule were applied on a noiseless system, and slowly enough to guarantee adiabaticity, the differences between the experimental Rydberg Hamiltonian and the ideal UD-MIS Hamiltonian may imply non-optimal output results. Ref. [33] gives examples of graph instances where the \( 1/r^6 \)-tail of the Rydberg Hamiltonian
We note that this form of optimization is nothing but a batic theorem would make this parameter useless, as the time dependence of the coefficients of the dephasing (top) and Rabi (bottom) terms, see Eq. (7).

The only remaining variational parameter is thus the total annealing time \( t_f \). In the absence of noise, the adiabatic theorem would make this parameter useless, as the success probability of QA gets larger as the annealing time \( t_f \) gets larger. In the presence of noise, however, long annealing times lead to an increased sensitivity to decoherence, and it thus becomes important to choose a \( t_f \) large enough that one can find a good approximation to the ground state, but short enough that the system has not undergone too much decoherence.

We note that this form of optimization is nothing but a very crude optimal control protocol, and that much more sophisticated schedules \( \tilde{\Omega}(t) \) could be designed using quantum optimal control (see, e.g., [34]).

The approximation ratio for our algorithm is defined, in its "one-shot version" (see discussion in Sec 1C), as

\[
\alpha \equiv \frac{E_{\text{target}}(t'_f)}{E_{\text{MIS}}(S^*)}
\]

with \( E_{\text{target}} \) defined in Eq. (6), and \( E_{\text{MIS}}(S^*) = -|S^*| \). We note that because of the definition of \( H_{\text{target}}, E_{\text{target}} \) is usually negative.

4. Hardware model

The Hamiltonian given in Eq. (7) describes the system of neutral atoms in the absence of errors. In this work, based on a validation with respect with previously published experimental data, we take into account two sources of errors: dephasing noise and readout errors.

**Decoherence by dephasing** Decoherence is taken into account via a single dephasing model that gives a reasonably accurate fit of the published experimental data (as checked in Appendix A1). Following Ref. [22], we describe this dephasing with a Lindblad master equation

\[
\frac{d\rho}{dt} = -i [H, \rho] - \frac{1}{2} \sum_{i \in V} \gamma_i \left( \left[ L_i^\dagger L_i, \rho \right] - 2 L_i \rho L_i^\dagger \right)
\]

where \( \rho \) is the density matrix describing the mixed state of the system, and \( H = H_{\text{Rydberg}}(t) \). The jump operators \( L_i \) corresponding to dephasing take the following form:

\[
L_i = \hat{n}_i
\]

We choose a uniform dephasing described by a single dephasing parameter \( \gamma_i = \gamma \).

**Readout assignment errors** Following Ref. [35], we model the readout errors by a simple assignment error model that accounts for the probability \( \epsilon \) of erroneously detecting an excited atom \(|1\rangle\) while it was in fact in its ground state \(|0\rangle\) (false positives), as well as the probability \( \epsilon' \) of not detecting an excited atom (false negatives). This simple error model is characterized by a so-called assignment probability matrix:

\[
A_1 = \begin{bmatrix}
p(0|0) & p(0|1) \\
p(1|0) & p(1|1)
\end{bmatrix},
\]

with \( P(1|0) = \epsilon \) and \( P(0|1) = \epsilon' \), that modifies single-qubit probabilities as follows

\[
\begin{bmatrix}
\tilde{p}(0) \\
\tilde{p}(1)
\end{bmatrix} = A_1 \begin{bmatrix}
p(0) \\
p(1)
\end{bmatrix}
\]

We will neglect any leakage out of the computational subspace, and thus assume \( P(0|0) = 1 - \epsilon \) and \( P(1|1) = 1 - \epsilon' \).

The probability distribution for all atoms is thus modified as \( \tilde{P} = \{ A_1 \otimes A_1 \cdots \otimes A_1 \} \cdot P \).
Repetition rate. The repetition rate in state-of-the-art experiments is in the 3-5 Hz range \cite{36, 37}. It is dominated by the time it takes to load atoms into the trap and by the final measurement stage. In particular, the duration of the annealing ($t_1$) is negligible compared to the duration of the other stages.

B. Classical approach: A locality-based heuristic

We now switch, in this section, to a description of a simple locality-based UD-MIS approximation heuristic for UD-MIS, inspired from \cite{26}. We use it to estimate the classical boundary, given a fixed time budget, of the UD-MIS approximation. Algorithm 1 and Fig. 3 describe its operation.

The design of this heuristic was motivated by the relative complexity of existing Polynomial-Time Approximation Schemes \cite{26, 27, 31, 35}. While they all consist in splitting a graph into sub-instances that are solved independently, several decompositions generally have to be considered, before making a choice guaranteeing a lower bound on the approximation ratio. We chose to implement a simpler, intuitive approach that only requires, during one run, to process each vertex exactly once.

We obtain this simplification at the price of renouncing to a provable guarantee on the approximation ratio. However, since we are able to compute numerically these approximation ratios (by solving instances optimally to obtain the optimal maximum independent set size) up to relatively large sizes (500-550 vertices), this is a relatively minor drawback.

Another motivation is that the local nature of solving subproblems in our method is reminiscent of the finite quantum correlation distances that have been estimated experimentally in Rydberg systems \cite{22}, and numerically in our simulations (see Appendix F). It makes it particularly relevant to a comparison with Rydberg adiabatic quantum computing systems.

Our algorithm (Algorithm 1) consists in an overall while loop that continues until the graph is empty. In an iteration of the while loop, UD-MIS is locally solved around a randomly selected vertex. We refer to this set of randomly selected vertices as "seeds". For each seed $u$, we compute (using Breadth-First Search (BFS, \cite{29})) the set of vertices within a distance $d$ of $u$, that we call $S_d(u)$. $d$ is an integer that is given as an execution parameter to the algorithm (by distance we mean shortest-path distance within the graph and not geometric distance between points). UD-MIS is then solved optimally for the subgraph induced by $S_d(u)$, which is removed from the graph. This process is repeated until the graph is empty. Figure 3 illustrates an iteration of the algorithm on a specific seed. The final solution returned by the algorithm is then the union of all the local maximum independent sets obtained throughout the execution by locally solving UD-MIS around the seeds. To ensure that the returned solution is indeed an independent set, in addition to removing $S_d(u)$ from the set of vertices after its processing, we further remove any vertex connected to the computed Maximum Independent Set of $S_d(u)$. Such vertices are indeed constrained to not be part of the final solution. In Figure 3, one can see an instance where vertices not belonging to $S_d(u)$ are nonetheless removed from the graph, because they are connected to the Maximum Independent Set of $S_d(u)$.

Note that the iterations of the algorithm cannot be easily parallelized as the results from solving $S_d(u)$ condition which vertices are removed, and therefore the rest of the execution of the algorithm.

Contrary to \cite{26} or \cite{27}, we do not formally call this simple locality-based randomized approach an "approximation algorithm", as we do not provide a provable guarantee, regardless of the graph, on the achieved approximation ratio given a certain value of $d$. However, in practice, it qualitatively behaves like a PTAS, as one can see on Fig. 4 (right), in the sense that the average approximation ratio achieved by this locality-based heuristic gets closer to 1 as $d$ increases.

Much like the parameters used in \cite{26} and \cite{27}, $d$ therefore acts as a tunable parameter that can be used to
refine the quality of an approximation at the expense of a potentially longer computation time.

Convergence to an approximation ratio of 1 for any graph as $d$ increases is guaranteed by the fact that if $d$ exceeds the diameter (the length of the longest shortest path between any pair of vertices) of the graph, then no matter the choice of seed $u$, $S_d(u)$ will cover the graph entirely, and a global optimal Maximum Independent Set will be computed.

Algorithm 1: Simple randomized approximation heuristic for UD-MIS

\begin{algorithm}
\caption{Simple randomized approximation heuristic for UD-MIS}
\begin{algorithmic}
\Input: a unit disk graph $G = (V, E)$. \Output: a bit assignment for every vertex \{$b_v\}_{v \in V}$, coding for an independent set.

\Function{Execution parameter:}{An integer $d \geq 0$}
\State $B \leftarrow \emptyset$ \Comment{$B$ will contain the “border” between exactly-solved patches}
\State $W \leftarrow V$
\While{$W \neq \emptyset$}
\State $d \leftarrow d + 1$ \Comment{$d$ is the current neighborhood size}
\State $u \leftarrow$ Pick a vertex $u \in W$ randomly.
\State $S_d(u) \leftarrow \{v \in V : d(u, v) \leq d\}$
\State $C_{u,d} \leftarrow \{v \in W \setminus S_d(u) : \text{s.t. } b_v \text{ is constrained to be } 0\}$
\State $B \leftarrow B \cup C_{u,d}$
\State $W \leftarrow W \setminus S_d(u)$
\EndWhile
\Return $W \setminus C_{u,d}$
\EndFunction
\end{algorithmic}
\end{algorithm}

A particular but valid parametrization for Algorithm 1 consists in choosing $d = 0$. It appears for instance on Figure 4 or Figure 13. By definition, a BFS sphere $S_d(u)$ with $d = 0$ simply corresponds to $\{u\}$. The “MIS” of a graph with one vertex $u$ is simply $\{u\}$ and the set $C_{u,d}$ (see line 7 of Algorithm 1) is simply the neighborhood of $u$ in the graph. Running the heuristic with $d = 0$ therefore corresponds to randomly selecting available vertices and putting them into the returned solution, without really computing anything.

III. RESULTS

In this section, we summarize our main findings, namely quantum and classical approximability results, and then explain the practical implications of this comparison.

The main results are shown in Fig. 4, where we compare the approximation ratios as defined above as a function of the problem size for various external parameters, both in the quantum and the classical approach.

A. Results of the quantum approach

The quantum results are shown in the left panel of Fig. 4. We show the average approximation ratios obtained in three hardware scenarios for Rydberg platforms: (i) state-of-the-art hardware, corresponding to already published experimental results [22], with noise levels captured by one parameter, $\gamma = 3.0$; (ii) near-term hardware, corresponding to current noise levels on the most advanced platforms, with noise levels divided by a factor of 10 with respect to the first scenario [36], $\gamma = 0.3$; (iii) noiseless hardware, with a vanishing noise parameter $\gamma = 0.0$. The definition of, and justification for these models is given in section II.A.4 above, and in Appendix A below.

We show the average results obtained with a time budget of 0.2 seconds and 2 seconds. With a realistic experimental repetition rate of about 5 Hz [36, 37] (comprising initialization, evolution and readout), these two cases correspond to a single repetition (measurement) of the algorithm, and ten repetitions, respectively. They thus correspond to the two possible definitions for the approximation ratio that we discussed in section IC.

As expected, decreasing noise levels yield improved approximation ratios, as do longer computational time budgets. For small system sizes (here number of neutral atoms), the approximation ratio decreases as the number of atoms increases. The behavior then differs for different run times:

For one repetition (0.2 s) the approximation ratio (corresponding to the statistical expectation value of the approximation over the quantum state distribution) saturates for the largest values of noise ($\gamma = 3.0$, and to a lesser extent $\gamma = 0.3$), while it slowly decreases for smaller noise parameters ($\gamma = 0.0$).

For several (10) repetitions, the approximation ratio does not seem to saturate but instead to slowly decrease with the number of atoms.

Both behaviors—saturation for one shot and slow decrease for several shots—can be explained, and we will use this explanation to extrapolate the curves to much larger number of atoms.

The saturation one observes for the average one-shot approximation ratio and the larger noise levels can be traced back to the finite correlation lengths induced by an imperfect preparation of the sought-after MIS solution.

By analyzing the spin-spin correlation function, we compute rough estimates of the spin correlation length $\xi$ for various system sizes and noise levels (see details in Appendix A.4), and conclude that the correlation length is set primarily by the noise (decoherence) level, and is roughly independent of the system size (see Fig. 12). (The value we find for $\gamma = 3.0$, $\xi \approx 1.4$, is very similar to the correlation length extracted in previous experiments, albeit on a regular lattice [22].) This allows us to
compute the approximate system size $N^*$ above which finite size effects should become irrelevant in the numerical simulation. Such a size can be estimated by counting the number of atoms comprised in a $\xi \times \xi$ square. For the atom density we consider, $\nu = 2$, we find $N^* \approx 4$ for $\gamma = 3.0$ and $N^* \approx 30$ for $\gamma = 0.3$ (for $\gamma = 0$, we find much larger, although finite estimates, of $> 1000$: our resource Hamiltonian prepares only an approximate ground state of $H_{\text{target}}$, as expected). We argue that the average approximation ratio, which is a function of the average (local) spin value $\langle z_i \rangle$ (see Eq. (6)), should not depend on $N$ beyond $N^*$. (It also depends, a priori, on $(z_i \bar{z}_j)$, but this correlation can be regarded as weak because the dynamics are dominated, by design, by independent-set configurations, see Appendix E.1.)

We therefore extrapolate the value computed numerically for $N = N^*$ to larger values, provided we can reach a size of $N^*$ atoms with our numerical simulations. By reaching 26 atoms, we achieve this for $N^*(\gamma = 3.0)\approx 4$ and $N^*(\gamma = 0.3)\approx 30$. Since we can even go beyond $N^*$ (especially so for $\gamma = 3.0$), we can not only perform the extrapolation, but also check that the approximation ratio has already reached saturation for number of atoms close to $N^*$. The extrapolated values are shown as the dash-dotted lines in Fig. 4. We can regard these values as a lower bound for the quantum approximation ratio.

The shape of the slow decrease in the maximum approximation ratio after several runs (Fig. 4, left side, solid line and dotted extrapolation) can be rationalized by statistical arguments. In the simpler case of a uniform final state distribution and mild assumptions detailed in Appendix G, one can show that the maximum approximation ratio after $n_{\text{shots}}$ repetitions on a system of $N_{\text{atoms}}$ atoms is upper bounded by $\alpha + \sqrt{\frac{\log n_{\text{shots}}}{2N_{\text{atoms}}}}$, with $\alpha$ the one-shot expectation value ($\alpha = 1/2$ for a uniform state distribution). It turns out that the actual approximation ratio for a uniform state distribution obeys the same $\propto \sqrt{\frac{\log n_{\text{shots}}}{2N_{\text{atoms}}}}$ dependence as its upper bound.

Interestingly, at fixed number of shots, for the uniform distribution, the maximum approximation ratio decreases quite slowly ($\propto 1/\sqrt{N_{\text{atoms}}}$) with the number of atoms. Conversely, at a fixed number of atoms, the increase in approximation ratio is very slow ($\propto \sqrt{\log n_{\text{shots}}}$), and even more so with larger number of atoms. This has the practical implication that a very large number of repetitions (shots) may not be the most efficient way to improve the quality of the algorithm. In particular, the scaling of the approximation ratio gain limits the potential added value of running several quantum machines in parallel within the same time budget, and selecting the best result. For $\sim 1000$ atoms, using our simplified model, running a hundred machines in parallel would only amount to a gain of about $\sqrt{\frac{\log 100}{2\times 1000}} \approx 0.047$ in terms of approximation ratio.

A mathematically rigorous extension of this observation to the generically non uniform state distribution pro-
duced by the quantum algorithm is beyond the scope of this work. Nevertheless, not only is the approximation ratio obtained for a uniform distribution a lower bound to the approximation ratio we obtain via quantum annealing, but the \( \sqrt{\frac{\log n_{\text{shots}}}{2N_{\text{atoms}}}} \)-dependence is arguably a purely statistical property; it is therefore not a feature of the sole uniform distribution.

We thus assume that the \((n_{\text{shots}}, N_{\text{atoms}})\) dependence of the maximum approximation ratio obtained numerically can be fitted and extrapolated by the following function:

\[
\alpha(N_{\text{atoms}}, n_{\text{shots}}, \gamma) = \alpha(\gamma) + \beta(\gamma) \sqrt{\frac{\log n_{\text{shots}}}{2N_{\text{atoms}}}},
\]

with \(\alpha(\gamma)\) being given by the saturation value computed for the one-shot case using the \(N^*\) extrapolation explained in the previous paragraphs.

We then use the numerical data we obtained for \(N_{\text{atoms}} = 6\) to 26 to find the value of the remaining parameter, \(\beta(\gamma)\), using a least-squares minimization. With this parameter, we can extrapolate the approximation ratio after several shots to larger numbers of atoms (see the dotted lines in Fig. 4, left panel), and compute the corresponding 95% prediction interval.

B. Results of the classical approach

1. Identification of an efficient locality-based heuristic

In order to provide a meaningful classical reference benchmark, we reviewed several approximation algorithms \([26, 27, 30, 31, 38, 39]\) for UD-MIS, and more specifically Polynomial-Time Approximation Schemes (PTAS, see Section 1B above).

We experimented in particular with a simplified version of [26], namely a locality-based approach that solves the MIS problems exactly within a certain distance \(d\) around randomly chosen points. Whereas the algorithm presented in [26] chooses this distance adaptively to ensure a specified approximation ratio, our simplified method takes a specific value as input and uses it everywhere (see Section II B for more details)

We see this classical locality-based heuristic as loosely “quantum-similar”, the finite solving distance \(d\) playing a similar role to that of the quantum correlation length \(\xi\), as characterized experimentally in Rydberg systems in, e.g., [22], and estimated numerically from our simulations (see Appendix F).

2. Classically attainable approximation ratios

The approximation ratio achieved by our locality-based approach, as a function of the number of vertices, and for several values of the fixed “correlation distance” \(d\), is plotted on the right panel of Fig. 4. Curves stop when even one run of the algorithm exceeds the time limit. Note that, for the range of atoms considered in Fig. 4, this only happens with \(d = 15\).

For other values of \(d\), the approach may run within the specified time budget well up to a several thousand vertices, as one can see on the lower right inset of Fig. 4, showing the maximum attainable graph size as a function of \(d\). In all cases, the approximation ratio appears to stabilize to an asymptotic value for large number of vertices. These asymptotic values are therefore accessible up to very large graph sizes (\(\geq 5000\)).

The evolution of the approximation ratio’s asymptotic value with respect to \(d\) is rather intuitive: as the solving distance \(d\) increases, the quality of approximation to the optimal improves. This makes sense within the analogy to the quantum correlation distance that we drew above: as vertices are allowed to “interact” across larger distances, global behavior improves.

On all curves (whether corresponding to the average approximation ratio or average maximum approximation ratio, see Section III for details on these definitions), the approximation seems to start close to \(\sim 1\) for small sizes. This makes sense, as a small graph is quite likely to be covered entirely by a BFS sphere \(S_d(u)\), no matter the random choice of the initial vertex \(u\).

Note that, in order to compute the approximation ratio achieved by an approach on a given graph, one needs to determine the size of a maximum independent set for this graph, i.e. to solve the problem optimally. This was carried out using the algorithm presented in [40] for graphs containing up to 550 vertices to compute the results of the right panel of Fig. 4. For our class of random graphs (see Section D for details), instances with \(\leq 200\) vertices are optimally solved in \(\leq 0.05\)s, while the “exponential explosion” of the solver happens around \(\sim 300 - 400\) vertices (See Figure 16 for more precise benchmark results).

3. Execution time

The largest attainable graph size given a value of \(d\) within the specified time-budget displays a much more counter-intuitive behavior, namely the bell-shaped curve we see on the lower-right inset of Figure 4, and in Figure 5. This behavior comes from the fact that the execution time of the locality-based heuristic is determined by a subtle interplay between the number of sub-instances to solve and their sizes, and does not necessarily increase with \(d\) (See Appendix B, in particular Figs. 15 and 19 for more details).

This explains why, for instance, the heuristic may be run with \(d = 5\) up to larger graph sizes than \(d = 2\) within the prescribed time-budget (7000 compared to 3000), even though \(d = 5\) involves solving larger sub-instances and achieves better approximation ratios.
Asymptotic values of the approximation ratio, plotted against the highest number of vertices reachable within two seconds, are reported on Figure 5. These values can be found on Figure 4 for \( d = 0, 5, 10, 15 \) (and Figure 18 for other values of \( d \)).

A striking fact concerning our locality-based approximation heuristic is therefore that one can run it with \( d = 8 \) for up to \( \sim 8000 \) atoms (see Figure 13) while staying within our time-budget of 2 seconds, and achieving an asymptotic approximation ratio value of \( \sim 0.95 \).

Note that our benchmarks were carried out with a single-core implementation of our locality-based heuristic. If the heuristic itself is not parallelizable (the state of the graph at some point of the execution depends on the result of previous iterations, because of line 7 of Algorithm 1), one could of course launch several runs of the algorithms in parallel.

With a serial implementation, the value of the “maximum average approximation ratio” coincides with the basic average approximation ratio at the time limit (as only one run can be carried out, see Fig. 4). Launching several independent runs in parallel could allow keep on achieving approximation ratios higher than the average baseline at the time limit.

However, based on the statistical arguments developed in the previous subsection, and detailed in Appendix G, it is reasonable to believe that the gain on approximation ratio will scale as \( \sqrt{\log n_{\text{max}}/2N_{\text{atoms}}} \), and therefore be of limited magnitude.

In addition, importantly, parallelism does not affect the maximum attainable graph sizes within a given time budget. As the heuristic itself is not parallelizable, several independent runs will still need the incompressible time of one run (i.e the values reported on Figs 13 and 19) to execute.

As the run time of the heuristic is dominated by the optimal solving of subinstances (see above, Fig. 15 and Appendix B), and as we neglect the “input-output” overhead of the solver we use ([40]), it is reasonable to say that improvements upon the classical benchmark results we report here may only either come from pure algorithmic improvements on [40], or from running the algorithm on more modern CPU chips.

C. Discussion of the break-even point

The comparison of the quantum and classical approaches we just described allows us to discuss the conditions that must be satisfied for the quantum approach to become more competitive than the classical approach. The approximation ratio as a raw metric must be put in perspective with the time-to-solution and the system size, as we stressed in the beginning of this section.

The classical frontier can readily be mapped for the classical approximation heuristic we have investigated. It is reasonable to believe that the gain on approximation ratio as a raw metric must be put in perspective with the time-to-solution and the system size, as we stressed in the beginning of this section.

Figure 5: Break-even diagram. Black dots (resp. grey squares): average classical approximation ratio reached for the maximum graph size reachable in 2 seconds (resp. 0.2 seconds) (asymptotic approximation ratio extrapolation from Figure 4 and 18, and maximum reachable graph size estimation from Figure 4 and 19).

Dash-dotted lines: lower bound to quantum approximation ratio reached for decoherence levels of \( \gamma = 3.0 \) (red) and \( \gamma = 0.3 \) (blue), assuming constant repetition rate with system size. Black (resp. grey) dashed line: break-even line for a 2-second (resp. 0.2-second) time budget.

We superimposed, in Fig. 5, the extrapolated average approximation ratios obtained through the quantum approach (i.e. corresponding to one repetition, or a budget of 0.2 seconds; we did not show the average maximum approximation ratio that could be reached within 2 seconds, which would be slightly, yet not substantially, higher).

In extrapolating these two lines to very large atom numbers, we made the important assumption that the experimental repetition rate will remain constant up to these large numbers of atoms. While a naive individual rearrangement of atoms would lead to a repetition rate roughly scaling with the inverse atom count, more advanced techniques could lead to a mitigation of this dependence (see, e.g., [41, Supp. Mat. 3.3], for an example of a more collective rearrangement, with a \( O(1/\sqrt{N_{\text{atoms}}} \) scaling of the repetition rate). Further improvements at the level of image acquisition are also possible, so that increases of the repetition rate by one order of magnitude are claimed to be realistic [42].

Working at a fixed time budget, there are two non-
IV. CONCLUSIONS

In this work, we compared the approximation ratios reachable in a finite computational run time using state-of-the-art, yet realistically implementable, quantum and classical approximation algorithms to the UD-MIS combinatorial optimization problem.

We set up a precise comparison methodology based on the computation of the average maximum approximation ratio reachable within a fixed time budget, as opposed to the more usual average approximation ratio used for characterizing randomized algorithms. We studied the dependence of both metrics with respect to system size and run time, under a realistic hardware model of the Rydberg quantum platform where this quantum algorithm could be executed thanks to the specific form of its interparticle interactions.

Based on simulations with up to 26 atoms, and on estimations of the spin correlation lengths, we predicted the large-size limit of the quantum average approximation ratio. We also inferred, from statistical arguments, the asymptotics of the quantum maximal approximation ratio. We found average approximation ratios of ≈ 0.72 for the noise levels corresponding to recently published data, and of ≈ 0.84 for near-future noise levels. We found that two key aspects for reaching quantum advantage with respect to classical algorithms are the coherence level and the repetition rate. Keeping a fixed time budget of 2 seconds, if the coherence level is substantially improved, or if the repetition rate is maintained while scaling to larger atom counts, Rydberg platforms could reach quantum advantage if approximation ratios above ≈ 95%, or atom counts of about 8000, can be attained.

These quite drastic size requirements point to the importance of developing and implementing quantum algorithms with better success probabilities and better suited for actual hardware.

Such algorithms would effectively lower the quantum advantage bar. In this study, we deliberately simulated a rather simple quantum annealing-based algorithm that can be implemented on today’s Rydberg platforms. Including more sophisticated existing and forthcoming algorithmic improvements will likely bring the quantum advantage frontier closer, provided the hardware specificities are duly taken into account: we did not seek, as it is done in optimal control (see, e.g, [34]), to optimize the classical control parameters of the quantum algorithm; nor did we try to use improved optimization costs such as the Conditional Value at Risk (CVaR, [15, 43]). Other possible routes towards improved approximation ratios could include algorithmic refinements of the annealing procedure (see, e.g, [44]), attempts to better tailor the resource Hamiltonian to the target Hamiltonian (like optimizing the placement of atoms to mitigate unwanted tail interactions), or even a parallel implementation of the algorithm on multiple Rydberg platforms.

ACKNOWLEDGMENTS

We acknowledge numerous fruitful discussions with A. Browaeys and T. Lahaye, as well as V. Lienhard (IOGS) and L. Henriet (Pasqal). The computations were performed on the Atos Quantum Learning Machine (QLM). This work received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 817482 (PASQuanS).
Appendix A: Supporting data for the quantum approach

In this section, we address the question of the maximum approximation ratio one can reach using a Rydberg platform as a function of the run time and the number of atoms. In the context of the hybrid quantum-classical algorithm we described in the Methods section, the run time of the algorithm is dominated by the physical set-up of the Rydberg atom platform.

The results we show are averaged over a certain number of random unit-disk graphs (the detailed graph generation procedure is described in Appendix D). The main metric for describing such graphs is the vertex density $\nu$. Small ($\nu \ll 1$) and large ($\nu \gg 1$) densities result in easy optimization tasks (see e.g. [15]). We thus choose an intermediate density $\nu = 2$ to tackle a hard optimization regime.

The numerical methods used to obtain the results below are described in Appendix E.

1. Determination of the noise model

The experimental results of Ref. [22] can be reproduced by adjusting $\gamma = 3.0$ in the noise model described above (section II A 4)). (This corresponds to $\gamma/2\pi = 0.48$MHz when reinstating dimensions).

In Figure 6, we show the temporal evolution of the Néel structure factor measured experimentally and compared to noisy simulations with and without readout errors, for a square lattice geometry (as opposed to the rest of the text, where we consider unit-disk graphs).

A value of $\gamma = 3.0$ gives an accurate agreement between the noisy simulation and the experimental data. As for the readout error, we take $\epsilon = \epsilon' = 3\%$ to reflect typical readout errors: Ref. [35] estimates $\epsilon \approx 1 - 2\%$, $\epsilon' < 5\%$.

2. Annealing time optimization

Here, we focus on the optimization of the annealing time for reaching the lowest target energy (see Appendix C for more details about the construction of $H_{\text{target}}$).

We first show numerical evidence that in the presence of...
noise, there exists an optimal annealing time $t^*_f$. Fig. 7 shows the dependence of the target energy on the annealing time. We observe that in the absence of noise, the larger the annealing time, the lower the target energy, as expected from the theory of adiabatic computing. By contrast, in the presence of noise, there is an optimal annealing time. Indeed, an increased annealing time entails a larger susceptibility to decoherence, and thus an increase in the minimal energy reached by the system (we note that for large noise values and large annealing times one obtains a target energy that gets closer to the energy of the maximally mixed state, as expected). Contrary to the noiseless case, there is thus a balance to be found between increasing the success probability of annealing by slowing down the annealing process, and fighting against decoherence by completing the computation as quickly as possible.

In the following, we thus use a classical optimizer (COBYLA) to find the annealing time that accommodates both constraints. In Fig. 8, we show the dependence of the approximation ratio on the number of optimization steps for different numbers of atoms and various noise levels.

For small numbers of atoms $N_{\text{atoms}} \leq 12$, we use the full Hilbert space to carry out the simulation. For larger atom numbers ($N_{\text{atoms}} \geq 12$), we use a restriction of the Hilbert space to independent sets, as described in Appendix E1. We observe that convergence is reached after a couple of iterations.

In Fig. 9, we show the obtained optimized annealing times as a function of the number of atoms. We observe that in the presence of noise, the optimal time is roughly independent of the atom number. In the noiseless case, it shows a slight increase with respect to system size, possibly indicating a decreasing value of the minimum gap.

Because of this weak, or absence of, dependence of the annealing time on the system size, the computation of the optimal annealing time can essentially be neglected in the computation of the total run time. It can indeed be regarded as a heuristic parameter that can be determined once and for all by studying small (and therefore negligibly quick to simulate numerically) instances of the problem.

We thus do not take this optimization step into account in the total run time estimations shown in this work.

3. Impact of the number of repetitions

Once the annealing time is fixed, the system can be evolved in time until its final state $\rho(\tau_f)$ is projected by a fluorescence measurement. The outcome of the measurement is a string of bits $s \equiv (b_1 \ldots b_n)$ indicating whether the $k$th atom is in its Rydberg state ($b_k = 1$) or its ground state ($b_k = 0$). The probability
Figure 11: Approximation ratio as a function of the number of atoms in quantum annealing for various fixed run times (same data as Fig. 10). Solid lines: without readout noise. Dashed lines: with readout noise $\epsilon = \epsilon' = 3\%$. Dash-dotted lines: fit function Eq. (19). Black, blue and red curves: approximation ratio for the final state distribution obtained with noise parameters $\gamma = 0.0$, 0.3, and 3.0, respectively, averaged over 20 random graphs (except for $N_{\text{atoms}} = 24$, 26: 10 graphs). Green dash-dotted lines: average approximation ratio for a uniform probability over IS states.

For this particular bitstring to be measured is given by Born’s rule $p_i = \langle \langle i | \Psi^0 \rangle |^2 = \langle \rho(t_i) | i \rangle$. This bitstring can be mapped to a target energy $E_{\text{target}}(i)$ to evaluate its proximity to the optimal solution.

The value for the target energy that is obtained on average by performing one measurement is thus given by

$$E[E_{\text{target}}] = \sum_i p_i E_{\text{target}}(i). \quad (A1)$$

One can repeat this measurement process $n_{\text{shots}}$ times and keep the maximum value obtained over these $n_{\text{shots}}$ repetitions. The average value of this maximum over $n_{\text{shots}}$ repetitions can be readily computed using $p_i$. For this, we define the cumulative energy density function as:

$$F(x) \equiv \sum_{i=0}^{d-1} p_i \theta(x - E_{\text{target}}(i)/E_{\min}), \quad (A2)$$

with $d$ the dimension of the space (number of IS states if we restrict the dynamics to IS states), $E_{\text{target}}(i)$ the target energy of state $i$, and $\theta$ the Heaviside function. We note that due to the restriction to the IS states, the support of $F$ is $x \in [0, 1]$.

4. Extrapolation to larger number of atoms: the role of the coherence length

The numerical simulations presented in the previous subsections are limited to a number of atoms (26) that is small compared to the graph sizes that can be reached in reasonable compute times by classical algorithms.

Yet, the numerical data for the higher noise level ($\gamma = 3.0$, see Fig. 4) display an apparent saturation of the ap-
proximation ratio. This saturation occurs already for quite small atom counts. Such an observation can be accounted for by considering the correlation length associated with the MIS ordered state. By conducting an analysis of the spin-spin correlation function (see Appendix F for details), we extract a graph-averaged correlation length and study its dependence on the graph size and noise level. As shown in Fig. 12, this correlation length is roughly independent of the graph size, and gets shorter as the intensity of decoherence ($\gamma$) increases. This physically-expected behavior can explain the saturation observed for $\gamma = 3.0$ and the onset of a saturation for $\gamma = 0.3$. Indeed, the value of the approximation ratio can be intuitively related to the correlation distance observed in the system: on average, the first *defects* in the MIS state will occur when perfect correlation is lost between two atoms. This happens at a length scale $\xi$. Increasing the system size beyond the number $N^*(\xi)$ of atoms contained in a volume of size $\xi$ will therefore not increase the approximation ratio. Consequently, the numerically computed approximation ratio will saturate beyond system sizes of the order of $N^*(\xi)$.

We can estimate $N^*(\xi)$ based on our numerical estimates of $\xi$: by counting the number of atoms in a square of length $\xi$, we find $N^*(\gamma = 3.0) = 4$ and $N^*(\gamma = 0.3) = 30$. This explains the observed saturation for $\gamma = 3.0$. For $\gamma = 0.3$, it is likely that the saturation is not entirely reached, although the maximum atom number of 26 is in the same range as 30, allowing us to use, as a rough estimate of the approximation ratio at large atom numbers, the value that we obtained numerically for $N_{\text{atoms}} = 26$.

Interestingly, the ratio of correlation length to the optimal annealing time (see subsection A.2 above) seems to be independent of $\gamma$ (its value is close to 1). This behavior deserves further investigation.

**Appendix B: Supporting data for the classical approach**

### 1. Approximation ratio

Figure 4 (right-panel) shows the approximation ratio achieved by our locality-based heuristic (Algorithm 1) on a class of random unit-disk graphs with constant density (see Appendix D for more details on our graph generation procedure), for several value of the resolution distance $d$.

As the time to perform a single run of Algorithm 1 increases with graph size (see Figure 13, the number $k$ of allowed runs within the time budget of 2 seconds (Figure 13, bottom) decreases. Therefore, because the average maximum approximation ratio describes the average ratio obtained when running $k$ runs and selecting the best outcome, its value is bound to converge to the “single-run” approximation ratio, equalling it when the time to carry out a single run is $\simeq 2$ seconds.

### 2. Run time and classical limit

Figure 15 shows the execution time of our locality-based approach along with a numerical lower bound solely taking into account the time to exactly solve MIS on sub-instances. Their agreement show that Breadth-First Search and other graph manipulations in Algorithm 1 are negligible compared to MIS-solving, and that the run time of the heuristic can be expressed as:

$$\text{run time} \equiv \sum_{u \in \text{seeds}} \text{exact solving time}(S_d(u)) \quad (B1)$$

The exact-solving time of a unit-disk graph is, as one can expect from the NP-hard nature of UD-MIS, exponential in the graph size. Figure 16 shows the average exact-solving time of our random graph class (see Appendix D for details) using a state-of-the-art solver, freely available on the Internet [40]. One can see the exponential “taking off” around 300-400 vertices with graphs of up to $\simeq 200$ vertices routinely solved in $\leq 0.05$ seconds.

This can explain why, although providing a better approximation ratio and involving the solving of larger instances, the heuristic with $d = 5$ is not slower than $d = 2$, as one can see on Figure 15. Indeed, as one can see on Figure 14, reporting the 0.9-quantile of the sizes of subgraphs to solve, $d = 5$ involves solving graphs largely falling under 200 vertices, which is below the “exponential explosion” of the exact solving run time at 300 - 400. Therefore, it involves fewer instances than, say, $d = 2$ that are bigger but not significantly longer to solve.

Figure 13 shows the run time (top) and corresponding number of allowed shots (bottom) within the time-budget we chose (2 seconds). The $d = 15$ curve fairly resembles the optimal solving run-time curve of Figure 16, going over time-budget at around 300 – 400 vertices. This can be explained by the size of sub-instances to solve being roughly equal to the entire system size up to 300 – 400 vertices, as exemplified by Figure 14.

Other curves involve instances whose sizes largely fall below the “exponential explosion threshold” of optimal solving. Therefore, they are able to stay within time-budget for much longer, timing out at $\sim 4000$, $\sim 5000$ and $\sim 8500$ for $d = 10$, 0 and 5 respectively.

### 3. Benchmark and implementation

Algorithm 1 involves, given a graph, producing sub-instances that will be solved and removed from the graph, until it is empty.

These sub-instances are solved with [40] (see next subsection). It comes as an executable, which is called onto a graph-describing text file. In our implementation, such text files are created, filled and erased automatically when executing Algorithm 1.
In our reported execution times, on Figure 13 or 15, for instance, we neglect the time taken by these file manipulations. We consider that it does not represent actual “computation time”, see subsection below and Figure 16 for further discussion.

The parametrization integer $d$ governs the degree of approximation to the optimal solution. A larger value of $d$ involves solving larger instances, as one can see on Figure 14.

4. Run times of exact branch-and-bound solver

In Fig. 16, we show the average run times of the optimal branch-and-bound solver, as a function of the number of vertices, for our class of random graphs (see Appendix D).

We used a state-of-the-art MIS solver [40] freely available on the web. It is a generic solver, not restricted nor specifically optimized for unit-disk graphs.
It is a legitimate question to ask whether, given the specific geometric structure of unit-disk graphs, a specialized solver would not fare better in practice. We tested this hypothesis by implementing a dynamic programming technique directly derivable from [25]. In algorithmic terms, it exploits the fixed-parameter tractability [45] of UD-MIS, with the thickness of input graphs taken as a parameter. Figure 17 compares the execution times of the specialized dynamic-programming and the generic branch-and-bound approaches, largely in favour of the latter, namely [40], whose exponential explosion happens around 300-400 vertices (See Figure 16) and not $\sim 30$ as in Figure 17.

In estimating the run time ("reported" run time in Fig. 16), we remove the input-output overhead specific to the solver of [40] so as to measure only the actual computational time ([40] works as an executable that reads and writes graphs from files). We report both run times (with and without I/O) as "wall-clock" time and "reported" run time, respectively. We observe that the I/O does represent a large part of the run time for small graphs.

5. Supplementary data

Figure 18 shows the average approximation ratio achieved by locality-based approximation heuristic for different values of the parameter $d$. For all values, the approximation ratio is observed to stabilize to an asymptotic value, which increases with $d$.

Figure 19 documents the execution time of our locality-based approximation heuristic for values of $d$ not shown on Figure 13, and the corresponding number of allowed shots. This data is used in Figure 5 to estimate the largest graph size attainable within a time-budget of 2s.

Appendix C: Reformulation of the UDMIS problem as a minimization problem

To be amenable to a resolution using a quantum processor, the problem has to be reformulated as an unconstrained minimization problem, ultimately in the form of an Ising Hamiltonian.
1. Boolean reformulation

Let us reformulate the UD-MIS problem defined in Section I entirely in terms of Boolean functions by defining

\[
\begin{align*}
 f(S) &\equiv \sum_{i \in V} n_i(S) = |S| \\
h(S) &\equiv \sum_{i,j \in E} n_i(S) n_j(S)
\end{align*}
\]  

(C1) (C2)

with \( S = (n_1, \ldots n_{|V|}) \) a given bitstring.

Let us note that \( S \in IS \iff h(S) = 0 \), so that the problem described in Eq. (1) can be reformulated as:

\[
\begin{align*}
 \max_{S \in B} & f(S) \\
 \text{s.t.} & h(S) = 0
\end{align*}
\]  

(C3)

Let us then denote \( S^* \) the solution of this optimization problem, called (UD)MIS, and \( p^* \) its optimum:

\[
p^* \equiv f(S^*)
\]  

(C4)

2. Lagrangian relaxation of UD-MIS

We want to transform the above constrained optimization problem, Eq. (C3), into an unconstrained optimization problem, while requiring that the optimal solution of this new optimization problem is still a Maximum Independent Set.

One way to achieve this is to construct a Lagrangian relaxation (see, e.g., Ref. [46]) of the problem by introducing the Lagrangian

\[
\Lambda(S, u) \equiv f(S) - u \cdot h(S)
\]  

(C5)

with the Lagrange multiplier \( u \). Since \( h(S) \geq 0 \) for any \( S \), we see that whenever \( u > 0 \), the second term penalizes non-feasible solutions (i.e., non-IS states). Let us also define the Lagrangian dual function:

\[
g(u) \equiv \max_{S \in B} \Lambda(S, u)
\]  

(C6)

At this point, we cannot ensure yet that the maximum \( S_u^* \) of the maximization problem Eq. (C6) is an IS. To impose this condition, one can prove (see subsection C4 below) that it is enough to require \( u > 1 \). Thus, we are going to solve the following maximization problem:

\[
\max_{u \in \mathbb{R}} \max_{S \in B} \Lambda(S, u)
\]  

(C7)

\text{s.t.} \( u > 1 \)

Since we have proven that the solution \( S_u^* \) is IS, we have \( \Lambda(S_u^*, u) = f(S_u^*) \). As in addition, among all IS solutions, the ones that maximize \( f \) are the Maximum Independent Sets, we have that \( S_u^* \) is indeed a Maximum Independent Set of the graph, and that the optimal solutions to both optimization problems are the same.

3. Reformulation as a minimization problem

Our goal is to use a quantum algorithm to solve the inner optimization problem, defined by Eq. (C6). As described in the main text, our quantum algorithm is going to perform a minimization task. Therefore, we first reformulate the above maximization problem as a minimization problem. This is achieved by redefining the objective function as \(-f\) instead of \(f\). Thus, we want to solve

\[
p^* = \min_{S \in B} \{-f(S)\}
\]  

s.t \( h(S) = 0 \)

with the following Lagrangian relaxation:

\[
q^* = \min_{u \in \mathbb{R}} \min_{S \in B} \{-f(S) + u \cdot h(S)\}
\]  

(C8)

s.t \( u > 1 \)

Following the reasoning of the previous subsection, we have the guarantee that the optimal solution is indeed an optimal Maximum Independent Set. One can now solve the double minimization problem of Eq. (C8) in two steps:

(A) the inner minimization

\[
g(u) = \min_{S \in B} \{-f(S) + u \cdot h(S)\}, u > 1
\]  

(C9)

can be performed using a quantum algorithm; since this quantum algorithm itself comes with parameters that can be optimized (like the annealing time), the inner minimization will itself be a quantum-classical algorithm; this inner minimization is represented as the light gray box (box (b)) in Fig. 20

(B) the outer minimization

\[
\min_{u > 1} g(u)
\]

can be performed using a classical minimization algorithm. This outer minimization is represented by the outermost box (box (a)) in Fig. 20. In this work, we do not study the influence of this outer minimization, and instead consider the Lagrange multiplier to be set to a fixed value \( u = 1.35 \).

4. Lagrangian relaxation: Proof of bound on \( u \)

Here, we prove the condition \( u > 1 \) that ensures that the cost function is maximal for a feasible solution (independent set).

Let us suppose that \( u > 1 \), and that we have a solution (bitstring) \( \tilde{S}_u^* \) of the maximization problem Eq. (C6) that is not an IS.

Since it is a solution, we have:

\[
f(\tilde{S}_u^*) - u \cdot h(\tilde{S}_u^*) \geq f(S) - u \cdot h(S), \forall S \in \Omega
\]
In particular, for IS states,

\[ f(\tilde{S}_u^*) - u \cdot h(\tilde{S}_u^*) \geq f(S), \forall S \in \text{IS} \]

(C10)

Let us call \( S_u^* \) the IS one finds when removing (some) edge-sharing vertices from \( \tilde{S}_u^* \) (one can always construct such a state: a possible (but non optimal) procedure is, for instance, to consider every conflict, i.e every edge \((i,j)\) such that both ends are in the set \( \tilde{S}_u^* \), and remove one of the two vertices from \( S_u^* \)). Let us call \( k > 0 \) this number of vertices.

Then:

\[ f(S_u^*) = f(\tilde{S}_u^*) - k \]
\[ h(S_u^*) = 0 \]
\[ h(\tilde{S}_u^*) \geq k \]

The last inequality comes from the fact that a new occupied vertex can create more than one edge. Thus:

\[ f(\tilde{S}_u^*) - u \cdot h(\tilde{S}_u^*) \leq (f(S_u^*) + k) - uk \]
\[ = f(S_u^*) - (u - 1)k \]

Since \( u > 1 \),

\[ f(\tilde{S}_u^*) - u \cdot h(\tilde{S}_u^*) < f(S_u^*), \]

which contradicts Eq. (C10).

Appendix D: Random graph generation procedure

We use the following procedure to generate random graphs:

**Algorithm 2: Random point generation algorithm**

Input parameters: density \( \nu \), number of points \( N \), exclusion radius \( r \).

Initialize an empty list to be filled by the points: \( V \).

while \( |V| < N \):

| pick two Cartesian coordinates \( x, y \) uniformly at random in \([0, \sqrt{\frac{N}{\nu}}] \); |
| if if the point \((x, y)\) comes within less than \( r \) of any point in \( V \) then |
| reject point |
| else |
| add it to \( V \) |

Result: \( V \)

The parameter \( \nu \) is the vertex density, i.e. the average number of atoms per unit square, while \( r \) plays the role of an exclusion radius.

We choose a density \( \nu = 2 \) that corresponds to a hard computational regime, i.e it is above the percolation threshold of \( \nu_c \approx 1.4 \), so that the generated graphs are connected on average, and not too high to be compatible with our exclusion radius, \( r = 0.3 \). This exclusion radius is chosen to be above the minimum distance between two atoms, and below the blockade radius of 0.5 (our condition for two vertices to be connected is that their distance should be \( \leq 1 \), which means that disks of radius 0.5 around them should intersect).

Appendix E: Numerical methods

1. Restriction to independent sets subspace

The Rydberg interaction (Eq. (7)) favors configurations where neighboring atoms have a different internal state. These configurations correspond to the independent sets (IS) of the underlying graph.

In our simulations, we therefore assume that the temporal dynamics are limited to the vector space spanned by such configurations.

We check that this approximation, which becomes exact only in the hard-sphere limit with \( V \to \infty \), has little quantitative impact on the final results: see, e.g., Fig. 7 (the solid lines show a computation within the full Hilbert space, the dashed lines with our restriction to the IS subspace), or Fig. 8 (see the panel for \( N_{\text{atoms}} = 12 \)). As can be expected, the difference is largest for high noise levels and unoptimized anneal times, both of which may favor non-IS states.
2. Quantum trajectories approach

The Lindblad master equation is solved using the quantum trajectories or quantum jump approach [47] as implemented in Qutip [48]. We take 100 trajectories per run.

3. Classical optimization

For the classical optimization of the annealing time, we choose the COBYLA optimizer as implemented in scipy.optimize [49].

Appendix F: Dependence of the correlation length on the noise level and the number of atoms

In this section, we give data supporting the claim that the correlation length is roughly independent of the graph size, and depends on the noise level $\gamma$.

At the end of the optimization procedure for $t_l$, we compute the correlation functions

$$\langle z_iz_j \rangle_\rho \equiv \text{Tr} \left[ \rho \sigma_i^z \sigma_j^z \right], \quad (F1)$$

with $i,j = 1 \ldots N_{\text{atoms}}$. We do not choose connected correlation functions because the sought-after solution is a classical state in general, which would entail a vanishing connected correlation function.

Such a correlation function displays oscillations corresponding to the alternation of correlations (1) and anticorrelations (-1) between the occupancy of site $i$ and that of site $j$, as shown in Fig. 21. These oscillations appear to be damped as the (Euclidean) intersite distance $d_{ij} = |r_i - r_j|$ increases, with a damping that increases as the decoherence level $\gamma$ increases. Our goal is to extract the envelope of these curves to compute a corresponding spin-spin correlation length $\xi$.

To this aim, we then compute the following binned correlation function

$$g^{(2)}(r) = \max_{G \in \mathcal{G}} \left| \langle z_iz_j \rangle_{\rho_G} \right| \quad (F2)$$

We thus extract the envelope by finding, for each bin around $r$, the maximum over several (here 5) random graph instances, and over all pairs such that $|r_i - r_j|$ is in the vicinity $\delta r$ of $r$ (here, we choose $\delta r = 0.04$, but the results do not depend on this value provided each bin $\delta r$ contains enough points). Similar (although quantitatively slightly different) results can be obtained by averaging over the bin, instead of maximizing.

We plot the corresponding curves in Fig. 22. An exponential fit $\exp(-r/\xi(\gamma,N_{\text{atoms}}))$ turns out to be an accurate description of the data. We extract the corresponding correlation length $\xi$, which we plot in Fig. 12.

We observe that for $\gamma > 0.0$, the correlation length is reasonably constant with respect to the number of atoms, and decreases substantially with increasing noise level $\gamma$. For $\gamma = 0.0$, the extracted correlation lengths are larger than those obtained in the noiseless case, but they do not appear to be constant with respect to the graph size.

We henceforth consider the value of $\xi$ to be constant with respect to the number of atoms: taking the average over the largest three atom counts (12, 14, 16), we get the estimates $\xi(\gamma = 0.3) = 3.9$ and $\xi(\gamma = 3.0) = 1.4$ (and $\xi(\gamma = 0.0) \approx 33$ if we average despite the variations).
We can roughly estimate the size (in terms of number of atoms) of the system beyond which the approximation ratio is going to get stable because the spatial extent of the atoms gets larger than the correlation length. For this, we estimate the number $N^*$ of atoms contained in a square of length $\xi$. It is given by $N^* = \nu \xi^2$. For the density under consideration ($\nu = 2.0$), and the above estimates for the correlation length, we obtain $N^*(\gamma = 0.3) = 30$, $N^*(\gamma = 3.0) = 4$ (and $N^*(\gamma = 0.0) \approx 2000$).

In practice, $N^*$ also gives us the atom number beyond which it is legitimate to extrapolate the numerical values obtained for lower atom counts.

**Appendix G: Expectation value of the maximum over a series of runs**

In this section, we rationalize the asymptotic behavior of the maximum approximation ratio with respect to the number of repetitions (shots) and system size.

1. **Definitions**

Let us denote by $p_i = |\langle i | \Psi \rangle|^2$, $i = 0 \ldots d - 1$ ($d$ is dimension of the space, $d = 2^{N_{\text{atoms}}}$ if all states are kept), the probabilities of the computational basis states in the final state $\Psi$ at the end of the annealing and

$$E_i = - \langle \hat{H}_{\text{target}} \rangle_{ii}$$

the corresponding target energies (we put a minus sign because due to the definition of $\hat{H}_{\text{target}}$, the minimum target energy, which measures the size of the maximum independent set, is negative). Let us further define the expectation value

$$\langle E \rangle_p \equiv \langle \psi | - \hat{H}_{\text{target}} | \psi \rangle = \sum_{i=0}^{2^n-1} p_i E_i$$

We want to compute the expectation value $M(N)$ of the maximum of the value of $-\hat{H}_{\text{target}}$ over $N$ repetitions of the preparation and measurement of $\Psi$ ($N = n_{\text{shots}}$ in the main text). It is given by:

$$M(N) = \mathbb{E}_P \left[ \max (E_{i_1}, \ldots, E_{i_N}) \right]$$

$$= \sum_{i_1, \ldots, i_N} \max (E_{i_1}, \ldots, E_{i_N}) P(i_1, i_2, \ldots, i_N)$$

where $P(i_1, i_2, \ldots, i_N)$ is the probability of observing the states $i_1, \ldots, i_N$ as the $N$ outcomes of the readouts. Since the repetitions are independent, $P(i_1, i_2, \ldots, i_N) = p_{i_1} \cdots p_{i_N}$ and thus

$$M(N) = \sum_{i_1, \ldots, i_N} \max (E_{i_1}, \ldots, E_{i_N}) p_{i_1} \cdots p_{i_N}$$

We note that

$$M(N = 1) = \langle E \rangle_p$$

Furthermore, denoting by $E_{\text{max}}$ the maximum target energy, we expect that, if the support of $p$ contains the state with maximum energy (the MIS),

$$M(N = \infty) = E_{\text{max}}.$$  

2. **Computation using the density of states**

We now define the following "target" density of states:

$$D(\varepsilon) \equiv \sum_{i=0}^{2^n-1} p_i \delta(\varepsilon - E_i)$$

It gives the probability of observing a given target energy over one repetition. Starting from Eq. (G3), we can then rewrite $M(N)$ as:

$$M(N) = \sum_{i_1, \ldots, i_N} \int d\varepsilon_1 \cdots \int d\varepsilon_N$$

$$\times \delta(\varepsilon_1 - E_{i_1}) \cdots \delta(\varepsilon_N - E_{i_N})$$

$$\times \max (E_{i_1}, \ldots, E_{i_N}) p_{i_1} \cdots p_{i_N}$$

$$= \int d\varepsilon_1 \cdots \int d\varepsilon_N$$

$$\times \sum_{i_N} p_{i_N} \delta(\varepsilon_N - E_{i_N})$$

$$\times \max (E_{i_1}, \ldots, E_{i_N})$$

i.e.

$$M(N) = \int d\varepsilon_1 \cdots \int d\varepsilon_N D(\varepsilon_1) \ldots D(\varepsilon_N) \max (\varepsilon_1, \ldots, \varepsilon_N)$$

$$\equiv \mathbb{E}_{R_N} \left[ \max (\varepsilon_1, \ldots, \varepsilon_N) \right]$$

with $R_N(\varepsilon_1, \ldots, \varepsilon_N) = D(\varepsilon_1) \ldots D(\varepsilon_N)$.

Let us define the random variable

$$\mathcal{E}_N \equiv \max (\varepsilon_1, \ldots, \varepsilon_N)$$

Let us call $F_N$ the cumulative distribution function (CDF) of $\mathcal{E}_N$. Then:

$$M(N) = \mathbb{E}_{R_N}$$

$$= \int_{E_{\text{min}}}^{E_{\text{max}}} \varepsilon F_N(\varepsilon) d\varepsilon$$

$$= [\varepsilon F_N(\varepsilon)]_{E_{\text{min}}}^{E_{\text{max}}} - \int_{E_{\text{min}}}^{E_{\text{max}}} F_N(\varepsilon) d\varepsilon$$ (G7)
Figure 23: Comparison of approximation ratio with its upper bound in the uniform case. Solid lines: $S \cdot (\alpha(N, S) - 1/2)^2$, with $
abla \alpha(N, S) = \mathbb{E} \left[ \max_{p=1\ldots,N} \{ E_i(S; p) \} \right] / S$. Dashed black line: log $N/2$.

Figure 22: Approximate upper bound for the approximation ratio corresponding to a uniform distribution of states, $p_i = \frac{1}{d}$, $\forall i = 0 \ldots d - 1$ (H1) with $d = 2^S$, and $S$ denotes the system size ($S = N_{\text{atoms}}$ and $N = n_{\text{shots}}$ in the main text).

The cost function we consider for a solution (bitstring) $i = (n^i_1, \ldots, n^i_S)$ is its Hamming weight (sum of bits): $E_i(S) = \sum_{k=1}^{S} n^i_k$ (H2)

It corresponds to a simplification of the target energy of the main text that corresponds to taking $U = 0$ (and inverting the sign). The approximation ratio is then $\alpha_i = E_i(S)/S$ as the maximum Hamming weight is $S$.

To study the expectation value of the maximum of $\alpha_i$ over $N$ repetitions, we first notice that $E_i$ is a random variable with binomial distribution law, $E_i \sim \mathcal{B}(p = 1/2, S)$. For large enough $S$, it is well approximated by a Gaussian-distributed random variable, $E_i \approx E^G_i$, with $E^G_i \sim \mathcal{N}(\mu = S/2, \sigma^2 = S/4)$. The expectation value of the maximum of $N$ Gaussian samples obeys the inequality:

$$\mathbb{E} \left[ \max_{p=1\ldots,N} \{ E^G_i(S; p) \} \right] \leq \mu(S) + \sigma(S) \sqrt{2 \log N} \quad (H3)$$

(Indeed, using Jensen’s inequality to obtain the first line, we have, $\forall \kappa > 0$:)

$$e^{\kappa \mathbb{E}[M_N]} \leq \mathbb{E} \left[ e^{\kappa M_N} \right] = \mathbb{E} \left[ \max_{p=1\ldots,N} e^{\kappa E^G_p} \right] \leq \sum_{p=1}^{N} \mathbb{E} \left[ e^{\kappa E^G_p} \right] = Ne^{\kappa^2 S/2}.$$

Taking the logarithm of this expression and picking $\kappa = \sqrt{2 \log(N) / \sigma}$ yields the $\sigma \sqrt{2 \log(N)}$ upper bound. Approximating $E_i \approx E^G_i$ in inequality (H3), and dividing by $S$, we obtain the final approximate upper bound on the approximation ratio:

$$\alpha(N, S) \approx 1/2 + \sqrt{\frac{\log N}{2S}} \quad (H4)$$

In Fig. 23, we check numerically that $\alpha(N, S)$ follows the same $(N, S)$ dependence as its upper bound, i.e that $\alpha(N, S) \approx 1/2 + \beta \sqrt{\log N / 2S}$, motivating the fitting function that is used in the main text.
[36] T. Lahaye and A. Browaeys, Private communication.

[37] V. Lienhard, Physique quantique expérimentale à N corps dans des matrices d’atomes de Rydberg. Des modèles de spins à la matière topologique., Ph.D. thesis, Institut d’Optique Graduate School (2019).

[38] S. C. Nandy, S. Pandit, and S. Roy, Information Processing Letters 127, 58 (2017).

[39] G. K. Das, M. De, S. Kolay, S. C. Nandy, and S. Sur-Kolay, Information Processing Letters 115, 439 (2015).

[40] C.-M. Li, H. Jiang, and F. Manyà, Computers & Operations Research 84, 1 (2017).

[41] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, (2016), arXiv:1607.03044.

[42] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, (2020), arXiv:2006.12326.

[43] P. K. Barkoutsos, N. Moll, P. W. J. Staar, P. Mueller, A. Fuhrer, S. Filipp, M. Troyer, and I. Tavernelli, (2017), arXiv:1706.03637.

[44] H. Yu, F. Wilczek, and B. Wu, (2020), arXiv:2005.13089.

[45] R. G. Downey and M. R. Fellows, Fundamentals of parameterized complexity, Vol. 4 (Springer, 2013).

[46] C. Lemaréchal, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 2241 (2001) pp. 112–156.

[47] J. Dalibard, Y. Castin, and K. Molmer, Physical Review Letters 68, 580 (1992).

[48] J. R. Johansson, P. D. Nation, and F. Nori, Computer Physics Communications 184, 1234 (2013), arXiv:1211.6518.

[49] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, Nature Methods 17, 261 (2020).